



Custom Name Andrei



CONTENTS

1. Amélioration de l'onglet Appels	8
2. test ddd	10
2.1. Plan d'exercice de laboratoire — Groupe LF HOL	63
3. À propos	69
3.1. Problème de MedScout avec le boling et les couleurs	69
3.1.1. Problème de table des matières Cadent	70
3.1.1.1. 5.4.1 Élément DocumentSearchRequest requis	101
3.2. Aruba 660-330 Linxusme3s	126
3.2.1. Tester le saut du curseur Adobe sur les tableaux	127
4. copier-coller de test	141
4.1. 5.4.1 Exigences des éléments de DocumentRetrieveRequest	142
4.2. vas duplicate	147
5. Test des signes diacritiques	148
6. Page de test CrossRiver 3	151
7. Dossier de test	152
7.1. recettes	152
7.2. test d'fr-CA-tête TOC manquant	154
8. Page de test CrossRiver	185
9. Page de test CrossRiver 2	186
9.1. Untitled	193
9.2. Test	194
9.3. Le texte alternatif de l'image Ashbyk disparaît	195
9.3.1. Untitled	204
10. c'est un titre h1	205
11. Baguette parce que car cela ouais	
11.1. Document de test	212
12. Retrait	218
12.1. Une autre page	218
12.1.1. Filtrage avancé	219
12.2. Nouvelle page	227
13. page de test Ocient	228
14. Tally Platform API	276
14.1. member	277
14.1.1. Find Members / Verify Member	277
14.1.2. Enroll Member	287
14.1.3. Get Member Profile	295
14.1.4. Update Member Profile	304
14.1.5. Partial Update Member Profile	307
14.1.6. Register for promotion using one-click encrypted tokens.	310
14.1.7. Update Member's Enrollment Date	312
14.1.8. Evaluate Bonus Eligibility	314
14.2. member-address	316
14.2.1. Get Member Addresses	316
14.2.2. Create Member Address	319
14.2.3. Get Member Address	323
14.2.4. Update Member Address	326
14.2.5. Delete Member Address	329
14.3. member-alias	330
14.3.1. Get Member Aliases	330
14.3.2. Create Member Alias	332
14.3.3. Get Member Alias	335
14.3.4. Update Member Alias	337
14.3.5. Delete Member Alias	339
14.4. member-coupon	340
14.4.1. Get Member Coupons	340
14.4.2. Get Member Coupon By Coupon Code	344
14.4.3. Cancel member coupon	348
14.4.4. Transfer Coupon	352
14.4.5. Post Coupon Use	354



14.4.6. Cancel coupon use(s)	360
14.4.7. Edit Coupon Use	364
14.4.8. Return Coupon Use	370
14.5. member-currency	376
14.5.1. Get Member Currencies	376
14.5.2. Adjust member currency	379
14.5.3. Get member currency buckets	388
14.6. member-email	390
14.6.1. Get Member Emails	390
14.6.2. Create Member Email	392
14.6.3. Get Member Email	394
14.6.4. Update Member Email	396
14.6.5. Delete Member Email	398
14.7. member-event	399
14.7.1. Post a member event	399
14.7.2. Get event types	401
14.8. member-list	403
14.8.1. Get member's lists	403
14.8.2. Add to member list	405
14.8.3. Remove from member list	407
14.8.4. Add to member list	408
14.8.5. Remove from member list	411
14.8.6. Find member lists	413
14.9. member-payment-card-account	415
14.9.1. Get Member Payment Card Accounts	415
14.9.2. Create Member Payment Card Account	418
14.9.3. Get Member Payment Card Account	422
14.9.4. Update Member Payment Card Account	425
14.9.5. Delete Member Payment Card Account	428
14.10. member-phone	429
14.10.1. Get Member Phone Numbers	429
14.10.2. Create Member Phone Number	431
14.10.3. Get Member Phone	434
14.10.4. Update Member Phone	436
14.10.5. Delete Member Phone	438
14.11. member-merge	439
14.11.1. Merge Members	439
14.11.2. Query Member Merge Status	441
14.12. member-preference	443
14.12.1. Get Member Preferences	443
14.12.2. Create Member Preference	445
14.12.3. Get Member Preference	448
14.12.4. Update Member Preference	450
14.12.5. Delete Member Preference	452
14.13. member-promotion	453
14.13.1. Get Member Promotions	453
14.13.2. Register member for promotion	464
14.13.3. Get Member Promotion By Member Promotion Id	466
14.14. member-redemption	477
14.14.1. Get Member Redemptions	477
14.14.2. Create Member Redemption	487
14.14.3. Get Member Redemption By Member Redemption Id	499
14.14.4. Patch Member Redemption	509
14.14.5. Modify Member Redemption	519
14.14.6. Cancel Member Redemption	530
14.14.7. Get Member Redemption By Reference Id	540
14.14.8. Get Member Redemption Aggregates	550
14.15. member-tier	552
14.15.1. Get Member Tiers	552
14.15.2. Update Member Tier	557
14.15.3. Rollover Member	561
14.16. member-transaction	562



14.16.1. Post Earning	562
14.16.2. Post Transaction Group for Earnings Processing	571
14.16.3. Get Member Transactions	579
14.16.4. Get Member Transaction	587
14.16.5. Edit Member Transaction	595
14.16.6. Cancel Member Transaction	597
14.16.7. Cancel Member Transaction (POST)	605
14.16.8. Clone Member Transaction	613
14.16.9. Get Member Transaction By Reference Id	621
14.16.10. Cancel Member Transaction By Reference Id	630
14.16.11. Cancel Member Transaction By Reference Id (POST)	638
14.16.12. Get Member Transaction Groups	646
14.16.13. Get Member Transaction Group By Transaction Group Id	654
14.16.14. Cancel Member Transaction Group By Transaction Group Id	662
14.16.15. Cancel Member Transaction Group By Transaction Group Id (POST)	670
14.16.16. Get Member Transaction Group By Transaction Group Reference Group Id	678
14.16.17. Cancel Member Transaction Group By Transaction Group Reference Id	686
14.16.18. Cancel Member Transaction Group By Transaction Group Reference Id (POST)	694
14.16.19. Get Member Transaction Aggregates	702
14.16.20. Retrieve member transaction data, aggregated and grouped by year.	704
14.16.21. Post Transaction for Earnings Processing	708
14.16.22. Get Available Manual Bonus Rules	711
14.16.23. Post Member Bonus	714
14.17. ping	722
14.17.1. Ping Service	722
14.18. promotion	723
14.18.1. Get Promotions	723
14.18.2. Generate one click promotion token	730
14.18.3. Search ProBonuses	732
14.19. tier	738
14.19.1. Get Tiers Information	738
14.20. redemption	740
14.20.1. Get Redemption Items	740
14.20.2. Find Redemption Rules	746
14.20.3. Create Redemption Rule	748
14.20.4. Replace Redemption Rule	753
14.20.5. Update Redemption Rule	758
14.20.6. Delete Redemption Rule	761
14.21. reward	762
14.21.1. Find Redemption Rewards	763
14.22. smart-group	765
14.22.1. Get Smart Groups	765
14.22.2. Create Smart Group	767
14.22.3. Get Smart Group	770
14.22.4. Get Smart Group Details	772
14.22.5. Add Smart Group Detail Record	774
14.22.6. Get Smart Group Detail Information	776
14.22.7. Delete (remove) Smart Group Detail record	778
14.23. member-transfer-points	779
14.23.1. Transfer Points	780
14.24. member-comment	782
14.24.1. Get Member Comments	782
14.24.2. Get Member Comments / Call Logs	786
14.24.3. Create a member comment / call log	791
14.24.4. Get the comment associated with given commentId	798
14.24.5. Update Member Comments	803
14.24.6. Update Status for Member Comments	809
14.24.7. Retrieve member comments	814
14.24.8. Create member comments for multiple members	819
14.24.9. Lock call log	821
14.24.10. Unlock call log	826
14.25. member-badge	831



14.25.1. Get Member badges	831
14.25.2. Create Member badge	834
14.25.3. Delete Member Badge	837
14.26. coupon-compensation	838
14.26.1. Get Compensation by Coupon Code	838
14.26.2. Get Compensation by Coupon Code or Confirmation Number	841
14.26.3. Add Compensation Nights	845
14.26.4. Edit Compensation Nights	847
14.27. member-referral	849
14.27.1. Get Referrals	849
14.27.2. Create a new referral for a member	851
14.27.3. Find Referral Codes	853
14.27.4. Find referral code details	855
14.27.5. Finds Referrer	857
14.28. background-task	859
14.28.1. Get Background task	859
14.29. async-task	861
14.29.1. Find Async Tasks	862
14.29.2. Get Async Task	864
14.29.3. Retry Async Task	866
14.29.4. Find Async Task Batches	868
14.29.5. Submit multiple async operations	870
14.29.6. Get Async Task Batch	872
14.29.7. Find Async Task Batch Types	874
14.29.8. Get Async Task Batch Type	876
14.30. system-broadcast-message	878
14.30.1. Create New Broadcast Message	878
14.30.2. Find Broadcast Messages	881
14.30.3. Update Existing Broadcast Message	883
14.30.4. Get Broadcast Message	886
14.31. member-linked-account	888
14.31.1. Links individual member account to an another member account	888
14.31.2. Retrieve member accounts linked to another member	891
14.31.3. Unlink Member Linked account	893
14.32. event-rule	895
14.32.1. Create Event Rule	895
14.32.2. Find Event Rules	899
14.32.3. Update event rule	902
14.32.4. Patch event rule	906
14.32.5. Retrieve event rule	910
14.32.6. Delete event rule	912
14.33. audit	913
14.33.1. Find Audit Logs	913
14.34. audit-configuration	915
14.34.1. List audit configurations	915
14.34.2. Create audit configuration	917
14.34.3. Patch audit configuration	920
14.34.4. Delete Audit Configuration	923
14.35. support	924
14.35.1. Submit Contact Us Inquiry	924
14.36. data-release	926
14.36.1. Search Data Releases	926
14.37. segment	930
14.37.1. Find segments	930
14.37.2. Create segment	933
14.37.3. Get segment	935
14.37.4. Update segment	937
14.37.5. Delete segment	940
14.38. communication-type	941
14.38.1. Find communication types	941
14.38.2. Create communication type	943
14.38.3. Get communication type	945



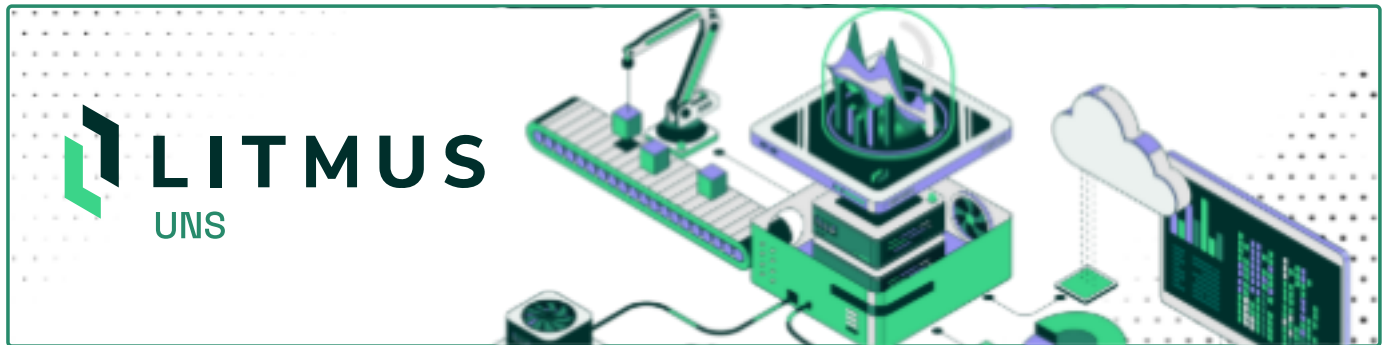
14.38.4. Update communication type	947
14.39. points-pool	950
14.39.1. Find Points Pools	950
14.39.2. Create Points Pool	952
14.39.3. Get Points Pool	954
14.39.4. Delete Points Pool	956
14.39.5. Add or Update Pool Member	957
14.40. customData	959
14.40.1. Get All Custom Data Metadata	959
14.40.2. Get Custom Data Metadata by ID	961
14.40.3. Gets Custom Data Detail Items	963
14.40.4. Create records under individual Ace Values	965
14.40.5. Get Custom Data Detail Item by ID	968
14.40.6. Update records under individual Ace values	970
14.40.7. Get Custom Data Detail Item by External Key	972
14.40.8. Update records under individual Ace values using externalKey	974
14.41. data-codes	977
14.41.1. Get data codes	977
14.41.2. Create Data Code	979
14.41.3. Get data codes by code type	982
14.41.4. Gets data code by type and code	984
14.41.5. Update datacode by codeld	986
14.41.6. Gets all data code types	989
14.42. managed-entity	991
14.42.1. Retrieve managed entity options.	991
14.43. expressions	993
14.43.1. Find expression builder configuration	993
14.44. report	995
14.44.1. Under Investigation Report	995
14.45. hotel-daily-summaries	997
14.45.1. Get hotel daily summaries	997
14.45.2. Update hotel daily summaries	999
14.46. user	1002
14.46.1. Get user context	1002
14.46.2. Create User	1004
14.46.3. Update User	1007
14.47. user-role	1010
14.47.1. Create a User Role, including Entitlements when provided.	1010
14.47.2. Retrieve all user roles. Include entitlements if specified.	1013
14.47.3. Update User Role and associate Entitlement if provided and active	1015
14.48. user-task	1017
14.48.1. Find user tasks	1017
14.48.2. Create user task	1020
14.48.3. Get user task	1023
14.48.4. Update user task	1025
14.48.5. Lock user task	1028
14.48.6. Unlock user task	1030
14.49. badge	1032
14.49.1. Get Badges	1032
14.50. coupon	1034
14.50.1. Create a coupon request.	1034
14.51. adjustment-rules	1040
14.51.1. Get adjustment rules	1040
14.52. outbound-notification	1042
14.52.1. Get outbound notification types	1042
14.52.2. fetch all outbound notifications sent within a specified timeframe	1044
14.52.3. Send an outbound notification	1046
14.53. standard-reports	1048
14.53.1. fetch results for a standard report	1048
14.54. enroll-codes	1050
14.54.1. Get enroll codes	1050
14.55. customer-satisfaction	1052



14.55.1. Customer Satisfaction	1052
14.56. member-referral-code	1055
14.56.1. Create Referral Code	1055
14.57. preference	1057
14.57.1. Get Preferences	1057
14.58. content	1059
14.58.1. Create Content	1059



1. Amélioration de l'onglet Appels



Litmus UNS

Litmus Unify (Unify) est un hub de données haute performance, sécurisé et hautement disponible, conçu pour l'évolutivité. Il est basé sur un moteur MQTT de classe entreprise pour une meilleure fonctionnalité et fiabilité.

Vous pouvez accéder à Unify en utilisant l'un des éléments suivants :

- Interface Web (Interface utilisateur)
- APIs

Sécurité

Unify est conçu avec la sécurité intégrée à chaque étape afin de répondre aux exigences spécifiques de l'IoT industriel.

Il offre :

- **Sécurité du déploiement** : Prend en charge les installations hors ligne et isolées du réseau.
- **Authentification et autorisation** : Authentification, autorisation et politiques RBAC (Role-Based Access Control) prêtes pour l'entreprise. L'interface utilisateur Unify est accessible uniquement avec un identifiant et un mot de passe.
- **Sécurité de l'accès à la plateforme** : Interfaces web HTTPS protégées et modernes.
- **Gestion des certificats** : Comprend un certificat autosigné par défaut et, par conséquent, toutes les communications sont sécurisées par défaut.
- **Chiffrement** : Toute communication entre un appareil Unify et le navigateur de l'utilisateur est protégée par le chiffrement de la couche de transport.
- **Mise à jour logicielle** : Les systèmes de mise à jour OTA implémentent des vérifications détaillées de hachage et d'autres validations.
- **Sécurité basée sur les jetons** : Fournit une sécurité basée sur des jetons pour l'authentification des applications.
- **Sécurité de l'accès aux données** : Broker de messages sécurisé et isolé pour l'abonnement aux données en temps réel.

Fonctionnalités

Unify offre un ensemble de fonctionnalités qui vous permettent d'implémenter vos conceptions UNS et d'atteindre des objectifs stratégiques en matière de données.

1. Exécution et surveillance du Unified Namespace d'entreprise
2. Sert de source unique de vérité pour tous les consommateurs de données



3. Répondre aux exigences de gouvernance des données



2. test ddd



- ✓ Ce tableau fournit un aperçu des nouvelles fonctionnalités et améliorations disponibles dans la nouvelle plateforme Cadent.





Fonctionnalité	Détails	Ancien	Nouveau	Améliorations
Audiences de santé	Capacité à créer une audience prédictive avec des données de santé.	✓	✓	Recherche basée sur LLM, logique AND-OR, identification directe des annonceurs et attribution



Fonctionnalité	Détails	Ancien	Nouveau	Améliorations
				à un groupe d'annonces
Intégration 1P	Téléchargez et activez directement les données first-party.		✓	Intégration directe sans devoir passer par LiveRamp
Informations sur l'audience	Informations clés sur les ensembles de données et les audiences, notamment le genre, l'emplacement, le revenu et l'appareil.		✓	Informations en temps réel sur les données intégrées et la composition de l'audience
Créateur d'audience	Créez une audience personnalisée basée sur des données 1P et 3P.	✓	✓	Portée et dénombrements d'informations en temps réel
Détails de campagne et d'élément de ligne*	Configuration de campagne et d'élément de ligne	✓	✓	UX rationalisée et représentation des données pour une visualisation plus rapide des données. La désignation du canal de l'élément de ligne automatise les paramètres en aval.
Détails du groupe d'annonces	Configuration principale et données de performance pour les groupes d'annonces.	✓	✓	Mappage des KPI aux modèles et mappage d'optimisation à l'optimiseur
API client	Capacité de créer et de gérer une campagne publicitaire via des APIs externes.		✓	API pour Advertiser, Campaign, Line Item, Ad Group et Creatives.
Diffusion d'annonces / Gestion des créations*	Créez, prévisualisez et gérez les créations dans toutes les campagnes.	✓	✓	La configuration de créations basée sur des modèles permet de rationaliser le trafic des créations
Gestion des pixels*	Créez et gérez des pixels 1P pour le suivi et la mesure des conversions de campagne.	✓	✓	Le mappage des pixels des éléments de ligne permet un mappage plus granulaire des conversions KPI
Détails/journal des enchères*	Les rapports granulaires et les journaux des demandes d'enchères optimisent les stratégies d'enchères et facilitent le dépannage de la diffusion.	✓	✓	Possibilité d'afficher les détails des enchères aux niveaux de l'élément de ligne et du groupe d'annonces
Gestion du ciblage*	Créer, appliquer et gérer des objets de ciblage complexes à travers les agences et les annonceurs.	✓	✓	Possibilité pour les utilisateurs en libre-service de gérer des objets au niveau d'une agence ou d'un annonceur
Options de ciblage*	Une vaste bibliothèque de paramètres de ciblage pour l'activation des campagnes, tels que l'Emplacement, l'appareil et le ciblage de contenu	✓	✓	Expérience utilisateur cohérente à travers les options de ciblage



Fonctionnalité	Détails	Antécédent	Nouveau Cadent	Améliorations
Connexions directes aux éditeurs	Intégrations directes avec les éditeurs pour un accès simplifié à l'inventaire.		✓	Gestionnaire d'inventaire Cadent

Q Search or ask...⌘ Shift F+

SUPPORT TEST SPACE⚙️ + ▾

klmn>

Furat de Adrian -- Test Github space for inte...>

Custom Space>

Articles-afbb51c6deb485629c74d9258766c...>

Articles-afbb51c6deb485629c74d9258766c...>

supportandrei/a1>

user-access-management-ditamap.zip>

Archbee-Andrei-Test/TestRepo>

ionutzzu12/demo-docs>

t1>

Space for test>

New test>



Fonctionnalité	Détails	An cie n	Nouv eau Cade nt	Améliorations
Audiences de santé	Possibilité de créer une audience prédictive avec des données de santé.	✓	✓	Recherche basée sur LLM, logique AND-OR, balisage direct des annonceurs et attribution à un groupe d'annonces
Intégration 1P	Téléchargez et activez directement des données first-party.		✓	Intégration directe sans devoir passer par LiveRamp
Informations sur l'audience	Informations clés sur les ensembles de données et les audiences, y compris le sexe, l'emplacement, le revenu et l'appareil.		✓	Informations en temps réel sur les données intégrées et la composition de l'audience
Créateur d'audience	Créez une audience personnalisée basée sur des données 1P et 3P.	✓	✓	Portée et nombre d'informations en temps réel
Détails de la campagne et de l'élément de ligne*	Configuration de la campagne et de l'élément de ligne	✓	✓	UX rationalisée et représentation des données pour une visualisation plus rapide des données. La désignation du canal de l'élément de ligne automatise les paramètres en aval.
Détails du groupe d'annonces	Données de configuration de base et de performance pour les groupes d'annonces.	✓	✓	Mappage des KPI aux modèles et mappage de l'optimisation à l'optimiseur
API client	Possibilité de créer et de gérer une campagne publicitaire via des API externes.		✓	API pour Advertiser, Campaign, Line Item, Ad Group et Creatives.
Diffusion d'annonces / Gestion des créations*	Créer, prévisualiser et gérer les créations dans toutes les campagnes.	✓	✓	La configuration de créations basée sur des modèles permet de rationaliser le trafic des créations
Gestion des pixels*	Créez et gérez des pixels 1P pour le suivi et la mesure des conversions de campagne.	✓	✓	Le mappage des pixels d'élément de ligne permet un mappage plus granulaire des conversions KPI
Détails/journal des enchères*	Les rapports détaillés et les journaux des demandes d'enchères optimisent les stratégies d'enchères et facilitent le dépannage de la diffusion.	✓	✓	Possibilité d'afficher les détails des enchères aux niveaux des éléments de ligne et des groupes d'annonces
Gestion du ciblage*	Créer, appliquer et gérer des objets de ciblage complexes à travers les agences et les annonceurs.	✓	✓	Capacité pour les utilisateurs en libre-service de gérer les objets au niveau d'une agence ou d'un annonceur
Options de ciblage*	Une vaste bibliothèque de paramètres de ciblage pour l'activation des campagnes,	✓	✓	Expérience utilisateur cohérente à travers les options de ciblage



Fonctionnalité	Détails	An cie n	Nouv eau Cade nt	Améliorations
	tels que Emplacement, ciblage des appareils et du contenu			
Connexions directes aux éditeurs	Intégrations directes avec les éditeurs pour un accès simplifié à l'inventaire.		✓	Cadent Inventory Manager





Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
aioc.alert.notification.mail.address	Si ce paramètre est défini, une notification est également envoyée à des utilisateurs spécifiques. S'il n'est pas défini, la notification est uniquement envoyée au gestionnaire concerné.	test_user@mail.com	NON	N/A
aioc.approval.excludeRequest.enabled	Lorsque ce paramètre est défini sur true et qu'un groupe possède plusieurs approbateurs, si l'un de ces approbateurs effectue une action nécessitant une approbation, il ne peut pas approuver sa propre demande. Dans ce cas, il est exclu de la liste des approbateurs et ne recevra aucun e-mail ni notification d'approbation. Cependant, même si le paramètre est true, s'il n'y a qu'un seul approbateur, cet utilisateur peut approuver sa propre demande. Si un approbateur est membre de plusieurs groupes et effectue une action nécessitant une approbation, le système évalue les approbateurs dans le realm concerné. Si l'utilisateur est le seul approbateur dans ce realm, il recevra l'e-mail/la notification d'approbation et pourra approuver sa propre demande. S'il existe d'autres approbateurs en plus de lui, il est exclu de la liste des approbateurs et ne recevra aucun e-mail ni aucune notification.	true,false	NON	false
aioc.available.jdbc.drivers	Ce paramètre est utilisé pour définir les types de bases de données demandés séparés par une virgule ",".	oracle.jdbc.driver.OracleDriver,org.postgresql.Driver,com.microsoft.sqlserver.jdbc.SQLServerDriver,com.mysql.jdbc.Driver,org.apache.cassandra.cql.jdbc.CassandraDriver,com.teradata.jdbc.TeraDriver,org.apache.hive.jdbc.HiveDriver,org.apache.hive.jdbc.DB2Driver,com.cdata.jdbc.couchbase.CouchbaseDriver	NON	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
aioc.backup.backupdir	Chemin du répertoire de sauvegarde dans Kron PAM. Le chemin de fichier "/pam/backup" doit être créé manuellement dans Kron PAM avant de démarrer la sauvegarde. Le fichier de sauvegarde est d'abord créé dans ce chemin, puis il est transféré vers le chemin défini dans le paramètre "aioc.backup.ftp.dirname".		OUI	N/A
aioc.backup.ftp.server.ip	IP du serveur vers lequel le fichier de sauvegarde sera transféré.		YES	N/A
aioc.backup.ftp.username	Nom d'utilisateur pour se connecter au serveur ftp défini avec le paramètre « aioc.backup.ftp.server.ip ».		YES	N/A
aioc.backup.ftp.password	Mot de passe de l'utilisateur défini avec le paramètre « aioc.backup.ftp.username ». Le paramètre doit être défini avec une option de chiffrement « yes ».		YES	N/A
aioc.backup.ftp.dirname	Chemin du répertoire où le fichier de sauvegarde sera envoyé sur le serveur ftp cible défini avec le paramètre « aioc.backup.ftp.server.ip ».		YES	N/A
aioc.backup.diskspace.min.gbyte	Espace disque requis pour que le serveur ftp transfère le fichier de sauvegarde. La valeur doit être définie en fonction de la taille de la base de données à sauvegarder.		YES	N/A
aioc.command	Définit la requête qui sera exécutée pour tous les tests de connexion si le	SELECT 1 FROM DUAL	NO	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
nd.provisioning.c3p0.PreferredTestQuery	ConnectionTester par défaut est utilisé. Définir une preferredTestQuery qui s'exécute rapidement dans la base de données peut accélérer considérablement les tests de connexion.			
aioc.connection.reservation.expiration.alert.before.values	Ce paramètre définit une notification d'expiration à envoyer n jours avant la fin de la réservation de connexion.	P. ex. : 1	NO N	N/A
aioc.connection.reservation.approval.requester.email.visibility	Ce paramètre active l'affichage de l'adresse courriel du demandeur dans les courriels de notification, l'application mobile et la section 'My Approvals' de l'interface utilisateur.	true,false	NO N	false
aioc.connection.reservation.max.allowed.hours	Ce paramètre est utilisé pour définir la durée maximale que les utilisateurs peuvent demander lors de la création d'une réservation de connexion.	P. ex. : 2160	NO N	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
aioc.connection.requireid	Ce paramètre détermine si le champ "reason" est obligatoire lors de la création d'une réservation à partir de l'écran Appareil > Réservation . Sa valeur par défaut est faux , ce qui signifie que le champ est facultatif. Si le paramètre est défini à vrai , les utilisateurs devront remplir ce champ afin de compléter le processus de réservation.	P. ex. : true,false	NON	false
aioc.date.format	Ce paramètre définit le format de la date et de l'heure en fonction d'une valeur de paramètres régionaux. Il utilise des paramètres régionaux tels que "en-US", "tr-TR" ou "ru-RU" pour déterminer automatiquement la façon dont la date et l'heure sont affichées dans le système. Son comportement par défaut suit le format standard des paramètres régionaux sélectionnés. Si ce paramètre est défini, le système formate la date et l'heure selon les paramètres régionaux fournis, sans nécessiter de définition de format personnalisée.	ru-RU,tr_TR	NON	N/A[EA1]
aioc.date.time.format	Ce paramètre permet de définir un format de date et d'heure personnalisé directement à l'aide d'un modèle de format (p. ex., "YYYY-MM-DD HH:mm:ss"). Contrairement au formatage basé sur les paramètres régionaux, il offre un contrôle total sur la manière dont les valeurs de date et d'heure sont affichées. Si ce paramètre est défini, le système utilise le format spécifié au lieu de s'appuyer sur les paramètres régionaux, ce qui permet des options de formatage plus précises et personnalisables.	YYYY-mm-DD HH:MM:SS	NON	dd.MM.yyyy HH:mm:ss



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
aioc.device.available.interface.names	Ce paramètre est utilisé pour définir un nom d'interface pour les appareils ayant la même adresse IP, afin qu'ils puissent être distingués lors de la connexion.	P. ex. : interface_1, interface_2	YES	N/A
aioc.download.android.active	Ce paramètre est utilisé pour activer le lien vers le Play Store.	Valeurs d'exemple : True,false	NO	true
aioc.download.ios.active	Ce paramètre est utilisé pour activer le lien vers l'App Store.	Valeurs d'exemple : True,false	NO	true
aioc.download.linux.active	Ce paramètre est utilisé pour activer une application que le système Linux peut télécharger.	Valeurs d'exemple : True,false	NO	true
aioc.download.mac.active	Ce paramètre est utilisé pour activer le téléchargement de l'application de bureau sur les ordinateurs Mac.	Valeurs d'exemple : True,false	NO N	true
aioc.download.window.s.active	Ce paramètre est utilisé pour activer le téléchargement de l'application de bureau.	Valeurs d'exemple : True,false	NO N	true
aioc.device.	Ce paramètre est utilisé pour définir les propriétés du groupe d'appareils.	Valeurs d'exemple :	NO N	null



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
group.property.keys		tag.Name,tag.Region,addDeviceSshKeyToUserSelection		
aioc.email.domains	Définissez ce paramètre avec les domaines de courriel associés. (Plus d'un domaine peut être ajouté avec une virgule. Ex. : SingleConnect.com, abc.com)	gmail.com, kronpam.com	NON	N/A
aioc.force.use.default.interface	Définissez ce paramètre à true pour forcer les utilisateurs suivants à utiliser la nouvelle interface utilisateur final. · Utilisateurs qui ne sont pas dans le groupe system.admins · Utilisateurs qui ne sont pas dans un groupe avec l'indicateur Admin Group · Utilisateurs qui ne portent pas le nom admin La valeur par défaut est faux .	true, false	NON	true
aioc.insteaddofad.line.manager	Ce paramètre définit où diriger la demande d'approbation afin de fournir le gestionnaire hiérarchique AD manquant et dirigera toutes les demandes dans le cas où le gestionnaire hiérarchique AD n'est pas importé dans Kron PAM ou est manquant	test_user@mail.com	NON	N/A
aioc.languages	Ce paramètre définit la langue préférée comme option dans l'interface graphique. Plus d'une préférence de langue peut être ajoutée avec un séparateur par virgule.	en_US, ru_RU, ko_KR	NON	en_US,de_DE,ko_KR
aioc.nssso.active	Ce paramètre est utilisé pour activer le module nssso dans aioc et accorder les droits SSH Proxy à Kron PAM.	true	NON	true
aioc.portal.d	Ce paramètre active ou restreint les connexions multiples sur l'interface	true, false	NON	false



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
isallowmulti.login	graphique de Kron PAM			
aioc.portal.session.idletime.minute	Ce paramètre définit le temps en minutes avant de déconnecter la session si l'utilisateur est inactif.	45	YES	30
aioc.portal.session.idletime.warning.before.min	Ce paramètre définit l'avertissement affiché avant la fin de la session.	5	NO	N/A
aioc.portal.allowed.ip.for.admin.users	Ce paramètre définit que les utilisateurs du groupe admin ne peuvent accéder qu'à travers les IP définies.	10.20.42.55,10.20.42.50	NO	N/A
aioc.region.visibility	Si ce paramètre est défini sur « true », le champ région est visible dans l'écran New Device Discovery de Device Inventory. S'il est défini sur « false », le champ région n'est pas visible dans l'écran New Device Discovery de Device Inventory. Si ce paramètre n'est pas défini, une licence controller/controlled est requise pour visualiser le champ région.			false
aioc.locking.ad.us	Ce paramètre définit la disponibilité permettant de rendre les utilisateurs AD verrouillables. Les utilisateurs AD seront	true	NO	false



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
er.enabled	verrouillés en cas de tentatives de connexion échouées et d'inactivité, de la même manière que les utilisateurs locaux.			
aioc.session.play.alert.mail.address	Les utilisateurs peuvent saisir une ou plusieurs adresses e-mail séparées par des virgules dans ce paramètre afin de spécifier où les notifications d'alerte doivent être envoyées lorsqu'une personne consulte une « Play Session » dans les « Session Logs ».	user1@krontech.com,user2@krontech.com	NO	N/A
aioc.users.default.password.length	Il définit le niveau de sécurité du mot de passe par défaut pour les utilisateurs dans PAM.	5	NO	N/A[EA1]
aioc.users.password.length	Il définit la longueur minimale du mot de passe requise pour les utilisateurs dans KronPAM.	5	NO	N/A
aioc.users.password.symbol.count	Il définit le nombre minimal de caractères spéciaux requis dans le mot de passe.	5	NO	N/A
aioc.users.password.symbol.chars	Il définit quels caractères spéciaux sont autorisés ou requis dans le mot de passe.	5	NO	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
aioc.users.password.strength.number.count	Il définit le nombre minimal de chiffres requis dans le mot de passe.	5	NO	N/A
aioc.users.password.strength.lowercase.count	Il définit le nombre minimal de lettres minuscules requises dans le mot de passe.	5	NO	N/A
aioc.users.password.strength.uppercase.count	Il définit le nombre minimal de lettres majuscules requises dans le mot de passe.	5	NO	N/A
aioc.timezone	Il peut être défini comme "New York USA" ou "GMT-5" ou "Etc/GMT-5". Pour ces 3 définitions, l'heure affichera "... (GMT-05:00)". Si le paramètre est défini, l'heure dans les e-mails sera convertie dans le fuseau horaire défini ; sinon, le fuseau système du serveur envoyant l'e-mail sera utilisé.			N/A
aioc.server.properties.keys	Ce paramètre définit les propriétés du groupe d'utilisateurs.	allowSftpInSshDevices	YES	allowSftpInSshDevices,httpProxyEnabled,otpEnabled,rdpProxyEnabled,sf



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
				tpProxyEnabled,sqlProxyEnabled,sshProxyEnabled,otpSmsEnabled
aioc.user.password.algo.rhythm	Ce paramètre définit l'algorithme préféré pour encoder les mots de passe des utilisateurs. Les algorithmes possibles sont définis dans les exemples.	Par ex. : MD5(par défaut), SHA256, SHA384, SHA512 et NTLM	YES	SHA512
aioc.user.group.change.notification.enabled	Lorsqu'un groupe d'utilisateurs est modifié ou qu'un utilisateur est ajouté/supprimé, un e-mail de notification est envoyé au gestionnaire du groupe. Définissez ce paramètre à faux pour désactiver l'envoi des e-mails de notification. La valeur par défaut est vrai .	false	No n	true
aioc.user.group.change.notification.email	Lorsqu'un groupe d'utilisateurs est modifié ou qu'un utilisateur est ajouté/supprimé, un e-mail de notification est envoyé aux adresses e-mail spécifiées. Plusieurs adresses e-mail peuvent être ajoutées à ce paramètre.	krontest@gmail.com, test@krontech.com	NO	N/A
aioc.user.password.log.check.count	Cette valeur vérifie si le nouveau mot de passe de l'utilisateur correspond à d'anciens mots de passe. (Ex : si la valeur est 2, les 2 derniers mots de passe sont vérifiés)	2	NO	2
aioc.users.default.password	Ce paramètre est utilisé pour définir le niveau préféré de robustesse du mot de passe utilisateur selon des niveaux prédéfinis : 0-Aucun	0	NO	1



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
ord.strength	1-La longueur du mot de passe doit être d'au moins 5 caractères 2-La longueur du mot de passe doit être d'au moins 5 caractères / Chiffre requis 3-La longueur du mot de passe doit être supérieure à 7 caractères / Majuscules-minuscules, chiffre et caractère spécial requis 4-La longueur du mot de passe doit être supérieure à 15 caractères / Majuscules-minuscules, chiffre et caractère spécial requis			
allow.changing.requirement.time.by.approver	Ce paramètre permet de modifier la durée pendant le temps d'approbation par l'approbateur.	true,false	NO	false
approval.sms.http.encoding	Format d'encodage. Voir 5.3.6 Paramètres SMS du flux de travail d'approbation.	Valeurs alternatives : UTF-8, UTF-16BE		
approval.sms.http.headers	En-tête HTTP SMS. Voir 5.3.6 Paramètres SMS du flux de travail d'approbation.	Ex : Content-Type:text/xml		N/A
approval.sms.http.url	URL http de l'expéditeur SMS. Voir 5.3.6 Paramètres SMS du workflow d'approbation.	Ex : http://api.smsexample.com/v1/send-sms		N/A
approval.stat	Lorsque le délai défini pour approuver la connexion instantanée prend fin, le statut	Ex :10,		N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
atus.change.to.expire	de la demande passe de « Waiting » à « Expired » si elle n'est pas approuvée par le(s) approbateur(s).			
approval.workflow.level.timeout.period.values	Les alternatives de valeur de période d'expiration pour les paramètres du workflow d'approbation, séparées par des virgules. Voir 5.3.5 Workflow d'approbation pour plus de détails. Les valeurs par défaut sont : 30 minutes, 2 heures, 1 jour	Ex : 30m,2h,1d	NON	30 minutes, 2 heures, 1 jour
autoapproval.when requester is approver	Lorsque l'approbateur est également le demandeur, l'approbation est automatiquement accordée.	true, false	NON	false
cookie.warning.message.content	Ce paramètre est utilisé pour informer les utilisateurs que des cookies sont utilisés.	Chez Kron, nous respectons profondément votre vie privée et nous nous engageons à faire en sorte que votre expérience avec notre produit soit tout simplement exceptionnelle. Notre engagement à offrir la meilleure expérience possible nous amène à utiliser des cookies. Ces petits éléments de données jouent un rôle crucial dans l'amélioration de votre interaction avec notre plateforme. En utilisant des cookies, nous pouvons offrir des paramètres et des préférences personnalisés qui correspondent à vos besoins uniques. Merci d'avoir choisi KronPAM. Nous avons hâte de continuer à améliorer votre expérience et de vous fournir un produit qui correspond parfaitement à vos besoins et à vos attentes.	NON	false
command.approval	Modèle pour les messages SMS à envoyer pour l'approbation des commandes via HTTP.			N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
al.sms.http.body	Voir 5.3.6 Paramètres SMS du flux de travail d'approbation.			
command.al.sms.smpp.body	Modèle pour les messages SMS à envoyer pour l'approbation des commandes via SMPP. Voir 5.3.6 Paramètres SMS du flux de travail d'approbation.			N/A
connection.approval.sms.http.body	Modèle pour les messages SMS à envoyer pour l'approbation des connexions via HTTP. Voir 5.3.6 Paramètres SMS du flux de travail d'approbation.			N/A
connection.approval.sms.smpp.body	Modèle pour les messages SMS à envoyer pour l'approbation des connexions via SMPP. Voir 5.3.6 Paramètres SMS du flux de travail d'approbation.			N/A
desktop.client.enabled	Ce paramètre est utilisé pour activer ou désactiver l'authentification multifacteur (MFA) pour la connexion à l'application cliente de bureau pour ses fonctions en ligne. La valeur par défaut est false.	true, false	NO N	false
disable.instantapproval.forhttp	Ce paramètre vérifie la case à cocher du niveau WF (« disable instant approval »). Toutes les approbations HTTP sont désactivées lorsque ce paramètre est défini sur true	true,false	NO N	false
disable.instantapproval	Ce paramètre vérifie la case à cocher au niveau WF (« disable instant approval »). Toutes les approbations RDP sont	true,false	NO	false



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
proval .for.rd p	désactivées lorsque ce paramètre est défini sur true.			
disabl e.inst ant.ap proval .for.sft p	Ce paramètre vérifie la case à cocher au niveau WF (« disable instant approval »). Toutes les approbations SFTP sont désactivées lorsque ce paramètre est défini sur true.	true,false	NO	false
disabl e.inst ant.ap proval .for.ss h	Ce paramètre vérifie la case à cocher au niveau WF (« disable instant approval »). Toutes les approbations SSH sont désactivées lorsque ce paramètre est défini sur true.	true,false	NO	false
export .secur ecrt.r ole.gr oups	Ce paramètre détermine s'il faut inclure les rôles des groupes d'appareils dans le dossier d'exportation. Le paramètre est utilisé pour exporter des listes d'appareils pour SecureCRT. SecureCRT est un client commercial SSH, Telnet et un émulateur de terminal.	true	YES	true
export .secur ecrt.s cript.e xtensi on	Ce paramètre détermine le type de fichier de script des appareils exportés. Le paramètre est utilisé pour exporter des listes d'appareils pour SecureCRT. SecureCRT est un client commercial SSH, Telnet et un émulateur de terminal.	js	YES	js
export .secur ecrt.s horten .name s	Si ce paramètre est enregistré avec la valeur true, le nom du groupe d'appareils parent est supprimé des noms de groupes d'appareils. Le paramètre est utilisé pour exporter des listes d'appareils pour SecureCRT. SecureCRT est un client commercial SSH, Telnet et un émulateur de terminal.	true	YES	true



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
export.securecrt.single.script	Ce paramètre détermine s'il faut inclure un fichier de script dans le dossier d'exportation. Le paramètre est utilisé pour exporter des listes d'appareils pour SecureCRT. SecureCRT est un client SSH et Telnet commercial, ainsi qu'un émulateur de terminal.	true	YES	true
export.securecrt.templates.dir	Ce paramètre définit le dossier de répertoire pour les appareils. Le paramètre est utilisé pour exporter des listes d'appareils pour SecureCRT. SecureCRT est un client SSH et Telnet commercial, ainsi qu'un émulateur de terminal.	\${netright.home}/templates/securecrt	YES	\${netright.home}/templates/securecrt
hsm.enabled	<u>Intégration du module de sécurité matériel (HSM)</u>	true	YES	
hsm.method	<u>Intégration du module de sécurité matériel (HSM)</u>	Client	YES	
hsm.provider.classname	<u>Intégration du module de sécurité matériel (HSM)</u>	com.ncipher.provider.km.nCipherKM	YES	
hsm.keystore.type	<u>Intégration du module de sécurité matériel (HSM)</u>	nCipher.sworld	YES	
hsm.keystore.alias	<u>Intégration du module de sécurité matériel (HSM)</u>	secureworld	YES	
hsm.keystore.load	<u>Intégration du module de sécurité matériel (HSM)</u>	xxx	YES	



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
.password				
hsm.keystore.entry.password	<u>Intégration du module de sécurité matériel (HSM)</u>	xxx	YES	
hsm.secretkey.algorithm	<u>Intégration du module de sécurité matériel (HSM)</u>	AES	YES	
iga.2fa.token.time step	Définit la période de validité du jeton	(en secondes) La valeur par défaut est 30	YES	30
iga.2fa.sms.http.body	Corps de la requête HTTP pour l'envoi de SMS	Valeur du corps HTTP	NO	N/A
iga.2fa.sms.http.headers	En-têtes inclus dans la requête HTTP SMS	Valeur des en-têtes HTTP	NO	N/A
iga.2fa.sms.http.secret.body	Données secrètes ajoutées à la requête HTTP pour la sécurité	Valeur des données secrètes HTTP	NO	N/A
iga.2fa.sms.http.url	URL utilisée pour l'envoi de SMS via HTTP	URL HTTP	NO	N/A
iga.2fa.sms	Contenu du message SMS lors de l'utilisation du protocole SMPP	Valeur du corps SMPP	NO	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
.smpp.body				
iga.2fa.sms.smpp.secret.body	Données secrètes dans le corps du message SMS SMPP	Valeur des données secrètes SMPP	NO	N/A
iga.2fa.use.external.mobile.app	Permet aux applications clientes MFA telles que Google Authenticator ou Microsoft Authenticator de s'exécuter.	true	YES	false
kron.criptoes.key	Ce paramètre définit la clé permettant de masquer les données sensibles.	IQtn5Fh70qhOeKnEDNKLcIZREHoWwhWdfmG0uOyKMtc=	YES	N/A
kill.session.reservation.enabled	Ce paramètre est utilisé pour terminer automatiquement les sessions après la durée spécifiée pour les connexions de réservation telles que RDP, SSH et SFTP Proxy.	true,false	NO	false
legal.notice.enabled	Les clients peuvent configurer un message d'avertissement juridique à afficher au début des sessions RDP ou VNC. Ce paramètre doit être défini sur « true » pour afficher le message. Ce paramètre est utilisé pour configurer un message d'avertissement juridique. Un redémarrage du service du portail web est nécessaire après les configurations.	true, false	NO	false
legal.notice.text	Le texte à afficher comme message d'avertissement juridique. Ce paramètre est utilisé pour configurer un message d'avertissement juridique. Un	<text>	NO	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
	redémarrage du service du portail web est nécessaire après les configurations.			
mail.templates.dir	Ce paramètre définit le répertoire par défaut des modèles de courriel. Kron PAM envoie des courriels aux administrateurs de groupe afin de les informer des nouvelles demandes d'utilisateur, des actions du gestionnaire de mots de passe, des demandes d'autorisation de commande, etc. Kron PAM envoie également des courriels de réinitialisation de mot de passe et des courriels de jeton d'activation MFA. Afin d'effectuer ces actions, les paramètres de messagerie doivent être configurés sur Kron PAM à l'aide de l'écran Mail Config dans le menu System Config Manager.	<code>\${netright.home}/templates/mail</code>	YES	<code>\${netright.home}/templates/mail</code>
max.pushcount.seconds.in.one.minute	Ce paramètre limite les notifications à envoyer dans la Mobile App afin d'éviter une suractivité du système de notification.	10	NO	
mobile.authentication.otp.enabled	Ce paramètre est utilisé pour activer ou désactiver l'authentification multifacteur (MFA) pour la connexion à l'application mobile dans ses fonctions en ligne.	true, false	NO	false
mobile.authentication.register.token.timeout	La période de validité des mots de passe à usage unique envoyés pour l'opération Register Token (en secondes)	Entier (valeur par défaut = 60)	NO	60



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
p.vali dity.s econd s				
mobile.tom cat.url	Ce paramètre définit l'adresse du serveur de l'application mobile Kron PAM.	https://sc251.SingleConnect.com:9443/mobile-api/rest	NO	http://localhost:9080/mobile-api/rest
multitenancy.enabled	Ce paramètre active la fonction de multilocation de Kron PAM.	true	NO	false
multitenancy.tacacs.port.range	La plage de ports à utiliser pour les périphériques TACACS des locataires doit être définie avec ce paramètre.	50000-50100	NO	N/A
netri ght.alias	Ce paramètre détermine quels modules utiliser dans AIOC. Vous pouvez définir le système comme Kron PAM, Single Monitor ou Single Command à l'aide de ce paramètre.	sc	NO	sc
netri ght.auth.ldap	Ce paramètre active ou désactive l'authentification LDAP/AD.	false	YES	false
netri ght.auth.ldap.base DN	Ce paramètre définit le DN de base LDAP. Le DN de base est la section du répertoire dans laquelle l'application commencera à rechercher des utilisateurs et des groupes.	DC=example,DC=com	NO	DC=example,DC=com
netri ght.auth.ldap.principal	Principal de sécurité du contexte défini à partir de l'expression définie comme uid.	uid=?,DC=example,DC=com	NO	uid=?,DC=example,DC=com



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
netrig ht.aut h.ldap .url	Ce paramètre détermine le nom d'hôte/l'adresse IP Active Directory/LDAP, le numéro de port et le protocole LDAP/LDAPS. Si plus d'une URL est utilisée, les paramètres doivent être séparés par « , ». (p. ex., ldap://10.10.10.10:389, ldaps://10.10.10.20:636)	ldap://1.1.1.1:389	NO	ldap://1.1.1.1:389
netrig ht.aut h.ldap .timeout	Durée d'expiration pour une réponse du serveur AD/LDAP	1200	NO	1000 (valeur par défaut en ms)
netrig ht.aut h.ldap .socket.timeout	Période d'expiration pour la connexion socket avec le serveur AD/LDAP	6000	NO	5000 (valeur par défaut en ms)
netrig ht.aut h.tacacs	Ce paramètre détermine l'utilisation de l'autorisation TACACS+.	true	YES	false
netrig ht.aut h.tacacs.server	Ce paramètre définit l'adresse du serveur TACACS+.	127.0.0.1	NO	127.0.0.1
netrig ht.aut h.tacacs.server.key	Ce paramètre définit la clé du serveur TACACS+.	z7i/Z15wXHgEJRwGFAQO3A==	NO	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
netright.autoddl	Ce paramètre est défini sur true lors de la mise à niveau de Kron PAM. La valeur de ce paramètre reste false lors de l'utilisation du système.	false	YES	false
netright.htbaseurl	Ce paramètre est utilisé pour configurer l'URL de base afin de fournir une connexion à partir d'un service proxy.	http://127.0.0.1:80	NO	N/A
netright.htcache.enable	Ce paramètre détermine si l'interface utilisateur dispose d'un cache.	false	YES	false
netright.htcontentroot	Ce paramètre définit le dossier du contenu racine.	\${netright.home}/filerepo	YES	\${netright.home}/filerepo
netright.hthiddenproperty.keys	Ce paramètre stocke les propriétés masquées.	.*.password	NO	.*.password
netright.hthome	Ce paramètre définit le répertoire netright.home.	/pam/gui/netright	YES	/pam/gui/netright
netright.htinstance name	Ce paramètre définit le nom de l'instance Kron PAM. Vous pouvez utiliser différents noms si vous utilisez plus d'une instance. Les informations sur le nom de l'instance dans l'interface WEB sont rendues visibles par défaut.	KronPAM	NO	N/A
netright.htjdbc.database	Ce paramètre définit le type de la base de données Kron PAM.	postgresql	NO	postgresql



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
netright.jdbc.password	Ce paramètre définit le mot de passe de la base de données Kron PAM.	*****	NO	N/A
netright.jdbc.url	Ce paramètre définit l'adresse de la base de données Kron PAM.	jdbc:postgresql://localhost:5444/aioc	NO	N/A
netright.jdbc.username	Ce paramètre définit le nom de la base de données Kron PAM.	aioc	NO	aioc
netright.license.file.path	Ce paramètre définit le chemin du fichier de licence.	\${netright.home}/licence.properties	NO	\${netright.home}/licence.properties_1
netright.name	Ce paramètre définit l'en-tête dans l'interface graphique Kron PAM.	KronPAM	YES	KronPAM
netright.version	Ce paramètre définit la version affichée dans l'interface graphique.	2.14.3	NO	N/A
notify.user.before.TTL.password.expiration	Ce paramètre est utilisé pour permettre aux utilisateurs de recevoir des notifications d'expiration un certain nombre de jours avant la date d'expiration (TTL du mot de passe) des mots de passe qu'ils ont définis via le groupe d'utilisateurs.	3	NO	N/A
nsso.nssso.ssl.port	Ce paramètre définit le port SSL du lien entre le SSH Proxy et Kron PAM.	4443	NO	4443
nsso.remote.desktop	Ce paramètre définit le dossier dans Kron PAM où sont stockés les fichiers transférés pendant une session RDP.	/tmp	NO	/pam/RDPShare



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
p.base.dir				
nsso.remote.desktop.daemon.host	Ce paramètre définit l'adresse hôte du RDP Proxy.	127.0.0.1	NO	127.0.0.1
nsso.remote.desktop.daemon.port	Ce paramètre définit le port du RDP Proxy.	4822	NO	4822
nsso.remote.desktop.drive.share.enabled	Ce paramètre est utilisé pour transférer des fichiers entre les points de terminaison RDP. Lorsque cette propriété est définie, un dossier spécial sur Kron PAM (/tmp/<username>) est partagé avec tous les points de terminaison RDP en tant que lecteur partagé nommé « G on SC RDP ».	true, false	NO	N/A
nsso.remote.desktop.idle.threshold	Limite de temps pour commencer le calcul du temps d'inactivité (millisecondes).	Exemple :40000	NO	30000ms(30secon des)
nsso.remote.desktop.keylogger.enable d	Ce paramètre accorde les droits permettant de voir les journaux de session RDP en tant que Key Logger. Les entrées de la souris et du clavier pendant les sessions RDP peuvent être consultées sur cette page.	true	NO	true



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
nsso.remote.desktop.p.key.logger.hidden.key.	Le Key logger des sessions RDP journalise tous les mouvements de touches en texte clair. Cette fonctionnalité doit être arrêtée afin de masquer certaines données. Lorsque les utilisateurs appuient deux fois sur la touche définie dans une session, le Key logger masque les mouvements de touches selon la limite définie dans ce paramètre.	20	YES	15
nsso.remote.desktop.p.key.logger.key.hiding.shortcut	Ce paramètre est utilisé pour définir la touche qui désactivera la journalisation des touches pour les touches cachées limitées définies. La touche par défaut est « ESC ».	ESC	YES	« ESC »
nsso.remote.desktop.p.ocrenabled	Ce paramètre accorde les droits permettant de voir les journaux de session RDP en tant que journaux OCR. Vous pouvez voir les activités effectuées par l'utilisateur pendant une session RDP.	true	NO	true
nsso.remote.desktop.p.ocrlang	Ce paramètre est utilisé pour obtenir les journaux OCR dans la langue requise. Les codes des langues prises en charge se trouvent dans le Guide de l'administrateur. Plusieurs langues doivent être séparées par "+".	eng+kor+tur+spa	NO	N/A
nsso.remote.desktop.p.ocr.threads	Ce paramètre définit le nombre maximal de threads alloués aux processus OCR dans les environnements multilocataires.			
	Ex. : 2	NO	2	



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
nsso.remote.desktop.session.duration.limit.warning.before.min	Ce paramètre est utilisé pour déterminer la durée pendant laquelle un avertissement est affiché avant l'expiration de la session.	Exemple : 4	NO	N/A
otp.rest.url	Ce paramètre est utilisé pour activer MFA. L'URL REST doit être définie comme l'adresse IP publique et le port de Kron PAM.	Ex. : http://127.0.0.1	NO	
pam.auth.username.case-sensitive	Lorsque ce paramètre est défini sur true, le système vérifie le nom d'utilisateur en tenant compte de la casse et bloque les connexions qui ne correspondent pas exactement au nom d'utilisateur défini.	true,false	NO	false
rdp.idle.session.timeout	Les sessions utilisateur peuvent être terminées en fonction de leur durée d'inactivité. Ce paramètre est utilisé pour définir une limite de délai d'expiration (minute).	Exemple :5	NO	60
rdp.min.required.character.limit	Ce paramètre est utilisé pour spécifier la limite minimale de caractères requise pour la connexion.	Min :5	NO	N/A
rdp.timeout.server.response	Ce paramètre est utilisé pour définir le délai d'expiration du temps de réponse du serveur. Le paramètre de délai d'expiration de connexion peut être configuré selon les besoins (seconde).	Ex. : 10	NO	15



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
nse.time				
sapm.job.password.change.thread.count	Ce paramètre est utilisé pour modifier le nombre de threads exécutant les tâches SAPM Auto Import.	5	YES	5
sapm.create.parent.group.right	Ce paramètre permet aux utilisateurs finaux de créer ou de restreindre leurs propres groupes parents Password Vault.	true,false	NO	false
sapm.show.password.expiration.time.value.s	Ce paramètre définit les durées de réservation des mots de passe des comptes SAPM. Lorsqu'un utilisateur effectue une réservation de mot de passe pour un compte SAPM, ces options de durée sont présentées pour réserver une durée.	5m,30m,2h,24h	NO	5m,30m,2h,24h
sc.aaa.freeraadius.password	Ce paramètre définit le mot de passe pour se connecter au serveur RADIUS.	*****	NO	N/A
sc.aaa.freeraadius.url	Ce paramètre définit l'adresse URL du serveur RADIUS.	jdbc:postgresql://127.0.0.1:5444/aioc	NO	jdbc:postgresql://127.0.0.1:5444/aioc
sc.aaa.freeraadius.u	Ce paramètre définit le nom d'utilisateur pour se connecter au serveur RADIUS.	aioc	NO	aioc



Nom du paramètre	Description	Exemple de valeur de paramètre	Redémarrage requis	Valeur par défaut
sername				
sc.aaa.radius.conf.path	Ce paramètre est utilisé pour définir le chemin du fichier de configuration RADIUS afin d'insérer les paramètres Active Directory/LDAP.	/etc/raddb/mods-available/ldap	NO	/etc/raddb/mods-available/ldap
sc.aaa.radius.restart.command	Ce paramètre définit la commande permettant de redémarrer le serveur RADIUS. Le serveur doit être redémarré avec cette commande pour appliquer les modifications.	systemctl restart pam-radius	NO	systemctl restart pam-radius
sc.aaa.tacacs.conf.path	Ce paramètre est utilisé pour définir le chemin du fichier de configuration TACACS+ afin d'insérer les paramètres Active Directory/LDAP.	/pam/tacacs/etc/kron_tacacs.conf	NO	/pam/tacacs/etc/kron_tacacs.conf
sc.aaa.tacacs.restart.command	Ce paramètre définit la commande pour redémarrer le serveur TACACS+. Le serveur doit être redémarré avec cette commande pour appliquer les modifications.	systemctl restart pam-tacacs	NO	systemctl restart pam-tacacs
sc.free.radius.server	Ce paramètre définit l'adresse du serveur Free RADIUS. Si ce paramètre n'est pas égal à l'adresse distante demandée, le programme renverra une erreur d'autorisation.	127.0.0.1	YES	127.0.0.1
sc.policy.xml.dir	Ce paramètre définit l'emplacement du fichier policy.xml.	/u01/nssoapp/conf/xml	YES	/u01/nssoapp/conf/xml
sc.portal.otp.enabled	Ce paramètre est utilisé pour activer ou désactiver l'authentification multifacteur (MFA) pour la connexion à l'interface graphique Kron PAM.	true, false	NO	false



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
sc.rdp.connection.otp.enabled	Ce paramètre est utilisé pour activer ou désactiver l'authentification multifacteur (MFA) pour les connexions RDP. (true=activé, false=désactivé)	false	NO	false
sc.rdp.otp.cache.enabled	Si ce paramètre est enregistré comme true, l'utilisateur ne sera pas invité à saisir un OTP pendant la durée du cache après avoir saisi l'OTP.	true	NO	true
sc.rdp.otp.cache.seconds	Ce paramètre définit le temps de cache en secondes.	300	NO	300
sc.rdp.page.list.element.length	Ce paramètre définit la longueur des noms de comptes et d'applications distantes. Parfois, les noms peuvent être longs et ne pas tenir à l'écran.	Valeur : Longueur 1 : 29 caractères 2 : 36 caractères 3 : 56 caractères 4 : 89 caractères	NO	1
sc.user.group.managed.member.of.group	Ce paramètre est utilisé pour déterminer si les gestionnaires feront partie du groupe d'utilisateurs ou non.	true, false	NO	true
selenium.chrome.binary	Ce paramètre spécifie le chemin de fichier exact vers l'exécutable Google Chrome sur votre système. Il garantit que Selenium lance la bonne instance du navigateur, ce qui est particulièrement	/usr/bin/google-chrome	NO	



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
	important dans les environnements personnalisés ou serveur.			
selenium.side.runner.binary	Ce paramètre pointe vers l'outil de ligne de commande utilisé pour exécuter les fichiers de projet Selenium IDE (.side). Il permet au framework d'automatisation de localiser et d'exécuter vos suites de tests préenregistrées directement depuis le terminal.	selenium-side-runner	NO	
selenium.chrome.args	Ce paramètre définit les arguments et indicateurs de ligne de commande spécifiques transmis à Google Chrome lors du lancement. Il vous permet de personnaliser complètement le comportement du navigateur pour votre environnement de test, notamment en l'exécutant en mode headless ou en contournant les restrictions de sandbox.	--headless, --disable-infobars, --no-sandbox, --enable-javascript, --disable-dev-shm-usage, --disable-gpu, --remote-debugging-port=9222, --incognito , --user-data-dir=/tmp/selenium-profil	NON	
selenium.side.runner.screenshots.directory	Spécifie le chemin du répertoire local où les captures d'écran sont automatiquement enregistrées pendant l'exécution des tests, fournissant un emplacement centralisé pour les journaux visuels.	/pam/gui/templates	NON	
selenium.side.runner.debug.mode	Active un état de journalisation amélioré qui affiche les journaux détaillés de la console du navigateur et les étapes internes d'exécution dans le terminal, facilitant un dépannage technique plus approfondi.	true, false	NON	
showWiretoSessionWarning	Ce paramètre peut être utilisé pour définir si la notification envoyée à l'utilisateur lorsque le processus wire to session démarre sera affichée ou non.	false, true	NON	



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
smpp. adresseRan	La plage d'adresses de destination à desservir par ce compte ESME. Ce paramètre est facultatif et les paramètres SMSC seront appliqués s'il n'est pas défini.	Exemple : 1* (pour les numéros commençant par 1	OUI	N/A
smpp. addrNpi	Indicateur de plan numérique (NPI) à utiliser pour les paramètres de plage d'adresses. Ce paramètre est facultatif et les paramètres SMSC seront appliqués s'il n'est pas défini.	Valeurs possibles : 0 : Inconnu 1 : ISDN (E163/E164) 3 : Données (X.121) 4 : Téléex (F.69) 6 : Mobile terrestre (E212) 8 : National 9 : Privé 10 : ERMES 14 : Internet (IP) 18 : ID client WAP	OUI	N/A
smpp. addrTon	Type de numéro (TON) à utiliser pour le paramètre de plage d'adresses. Ce paramètre est facultatif et les paramètres SMSC seront appliqués s'il n'est pas défini.	Valeurs possibles : 0 : Inconnu 1 : International 2 : National 3 : Spécifique au réseau 4 : Numéro d'abonné 5 : Alphanumérique 6 : Abrégé	OUI	N/A
smpp. bindMode	Mode de liaison pour le compte ESME. Ce paramètre est obligatoire pour l'envoi/là réception de SMS via SMPP.	Valeurs possibles : t : émetteur r : récepteur tr : émetteur-récepteur (émetteur et récepteur)	OUI	N/A
smpp. destinationNpi	Paramètre d'indicateur de plan numérique (NPI) à utiliser pour l'adresse de destination. Ce paramètre est facultatif et les paramètres SMSC seront appliqués s'il n'est pas défini.	Valeurs possibles : 0 : Inconnu 1 : ISDN (E163/E164) 3 : Données (X.121) 4 : Téléex (F.69) 6 : Mobile terrestre (E212) 8 : National	OUI	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
		9 : Privé 10 : ERMES 14 : Internet (IP) 18 : ID client WAP		
smpp.destination	Type de paramètre Number (TON) à utiliser pour l'adresse de destination. Ce paramètre est facultatif, et les paramètres SMSC seront appliqués s'il n'est pas défini.	Valeurs alternatives : 0 : Inconnu 1 : International 2 : National 3 : Spécifique au réseau 4 : Numéro d'abonné 5 : Alphanumérique 6 : Abrégé	OUI	N/A
smpp.enquireLinkPeriodMs	Période en millisecondes pour effectuer des requêtes EnquireLink vers le SMSC. Les requêtes EnquireLink sont utilisées pour vérifier l'état de santé de la connexion entre l'ESME et le SMSC cible. Toute valeur inférieure ou égale à « 0 » sera remplacée par la valeur par défaut de 5000 ms (5 secondes). La connexion SMPP sera automatiquement rétablie en cas d'échec de connexion SMPP durant les requêtes Enquire Link.	Ex : 10000 (En millisecondes)	OUI	N/A
smpp.ip	Adresse IP du SMSC. Ce paramètre est obligatoire pour l'envoi/la réception de SMS via SMPP.	Ex : 10.20.40.95	OUI	N/A
smpp.password	Le mot de passe utilisé pour authentifier un compte ESME défini dans le SMSC. Ce paramètre est obligatoire pour l'envoi/la réception de SMS via SMPP.	Ex : netright	OUI	N/A
smpp.port	Port de liaison pour SMPP, écouté sur le SMSC. Ce paramètre est obligatoire pour l'envoi/la réception de SMS via SMPP.	Ex : 16000	OUI	N/A
smpp.receive	Durée du délai d'expiration pour tenter de recevoir un message du SMSC. Ce	Valeurs alternatives : -1 (Attente infinie jusqu'à la réception d'un PDU.	OUI	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
eTime out	paramètre est facultatif.	1,2,3.... (nombre de secondes)		
smpp.serviceType	Service d'application SMS associé au message. Ce paramètre est facultatif et envoyé comme valeur par défaut s'il n'est pas défini.	Valeurs alternatives : (NULL): Par défaut CMT: Messagerie cellulaire CPT: Radiomessagerie cellulaire VMN: Notification de messagerie vocale VMA: Alerte de messagerie vocale WAP: Protocole d'application sans fil USSD: Données de services supplémentaires non structurées	OU	N/A
smpp.sourceAddress	L'adresse source à utiliser lors de l'envoi des messages. Ce paramètre est obligatoire pour l'envoi/la réception de SMS via SMPP.	Ex : PAM, +12348372939	OU	N/A
smpp.sourceNpi	Indicateur de plan de numérotation (NPI) à utiliser dans les paramètres d'adresse source SME. Ce paramètre est obligatoire pour l'envoi/la réception de SMS via SMPP. Il doit être défini comme Unknown (0) si une adresse source alphanumérique doit être utilisée pour envoyer des messages (Ex : SingleCon)	Valeurs alternatives : 0: Unknown 1: ISDN (E163/E164) 3: Données (X.121) 4: Téléx (F.69) 6: Mobile terrestre (E212) 8: National 9: Privé 10: ERMES 14: Internet (IP) 18: ID client WAP	OU	N/A
smpp.sourceTon	Type de Number (TON) à utiliser dans les paramètres d'adresse source SME. Ce paramètre est obligatoire pour l'envoi/la réception de SMS via SMPP. Il doit être défini comme Alphanumeric si une adresse source alphanumérique doit être utilisée pour envoyer des messages (Ex : SingleCon)	Valeurs possibles : 0 : Unknown 1 : International 2 : National 3 : Network Specific 4 : Subscriber Number 5 : Alphanumeric 6 : Abbreviated	OU	N/A
smpp.sync	Mode de réception. S'il est défini sur sync, l'application attend une réponse	Valeurs possibles : sync – Synchrone	OU	sync



Nom du paramètre	Description	Exemple de valeur de paramètre	Requise	Valeur par défaut
Mode	après l'envoi d'une requête PDU. S'il est défini sur async, l'application n'attend pas les réponses; celles-ci sont plutôt transmises à une implémentation de ServerPDUListener par le Receiver. Le listener reçoit également chaque requête PDU reçue du SMSC. Il s'agit d'un paramètre facultatif et la valeur par défaut est sync.	async - Asynchrone		
smpp.systemId	L'ID système utilisé pour identifier un ESME défini dans SMSC. Il est utilisé pour l'authentification de l'expéditeur SMPP. Ce paramètre est obligatoire pour l'envoi/la réception de SMS via SMPP.	Ex : netright	OUI	netright
smpp.systemType	Le type de système utilisé pour catégoriser le type de liaison ESME au SMSC. Ce paramètre est facultatif pour l'envoi/la réception de SMS via SMPP et, s'il n'est pas défini, le type de système est envoyé comme null.	Valeurs possibles : VMS : Système de messagerie vocale OTA : Système d'activation Over-the-air (NULL) : Par défaut	OUI	N/A
smss.channel	Canal à utiliser pour envoyer des SMS.	Valeurs possibles : http : pour utiliser un proxy SMS basé sur http smpp : pour utiliser SMPP vers SMPP	OUI	http
ssh.min.connection.characterLimit	Ce paramètre est utilisé pour spécifier la limite minimale de caractères requise pour la connexion.	Min : 5	NO	
sso.gossites	Ce paramètre définit l'emplacement du serveur.		NO	
sso.ip	Ce paramètre définit l'adresse IP du proxy SSH.	127.0.0.1	NO	Adresse IP du proxy SSH
sso.port	Ce paramètre définit le port de l'adresse IP du proxy SSH.	2222	NO	2222



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
syslog.messager.cFormat	Les formats RFC_5424 et RFC_3164 sont pris en charge dans la configuration SIEM. Ce paramètre détermine le format RFC et doit être défini avec l'une de ces valeurs.	RFC_5424, RFC_3164	NO	RFC_5424
syslog.messenger.content.format	Ce paramètre est utilisé pour déterminer le format du contenu.	KEY_VALUE, CEF, LEGACY_CEF	NO	KEY_VALUE
syslog.server.hostName	Kron PAM peut envoyer des journaux aux systèmes SIEM. Ce paramètre est utilisé pour définir l'adresse IP de l'hôte SIEM.	10.10.20.20	YES	-
syslog.server.port	Ce paramètre est utilisé pour définir le port de l'hôte SIEM.	514	YES	514
tfa.otp.issue	Le nom du serveur MFA. Cette chaîne est affichée en haut de la valeur Offline Token dans l'application mobile lorsqu'un code QR émis par le serveur est numérisé dans une application mobile.	String	NO	
user.forceManagerUserGroup	Si ce paramètre est enregistré comme true, vous devez définir un gestionnaire pour un groupe d'utilisateurs. Sinon, des groupes d'utilisateurs peuvent être créés sans gestionnaire.	true	NO	true
user.lock.afterFailedLoginsAttempts	Les utilisateurs sont verrouillés après un certain nombre de tentatives de connexion échouées. Ce paramètre définit le nombre de tentatives de connexion échouées avant le verrouillage du compte utilisateur.	5,10,20	YES	20



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
aioc.locking.ad.user.enabled	Ce paramètre définit la disponibilité permettant de rendre les utilisateurs AD verrouillables. Les utilisateurs AD seront verrouillés en cas de tentatives de connexion échouées et d'inactivité, comme les utilisateurs locaux.	true,false	NO	false
user.lock.afterInactiveMillis	Kron PAM verrouille les utilisateurs inactifs. Ce paramètre définit le temps d'inactivité maximal avant de verrouiller un utilisateur. Le mot de passe de l'utilisateur doit être réinitialisé pour déverrouiller le compte utilisateur.	2629743000	NO	2629743000
user.mail.from	Ce paramètre définit l'adresse courriel de l'expéditeur pour MFA.	change_it@change_it.com	YES	N/A
user.registration.enabled	Ce paramètre permet de masquer le bouton de nouvel utilisateur sur l'écran de connexion. La valeur par défaut est false.	true,false	NO	false
user.suspended.afterFailedLoginAttempts	Les utilisateurs sont suspendus après un certain nombre de tentatives de connexion échouées. Ce paramètre définit le nombre de tentatives de connexion échouées avant de suspendre l'utilisateur.	10	YES	10
user.suspended.forMillis	Après un certain nombre de tentatives de connexion échouées, les utilisateurs sont suspendus pendant la durée déterminée dans ce paramètre. Cette valeur est en millisecondes.	60	NO	60_1
windows.aut	Ces paramètres sont utilisés pour configurer les paramètres	= /pam/gui/conf/sc.keytab	NO	



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
h.keytab.pat.h	d'authentification Windows. (Les paramètres mentionnés ci-dessus doivent être définis pour l'interface utilisateur respective).			
windows.auth.spn		=HTTP/ Kron PAM ServerName	NO	
aioc.auth.rdc	Ce paramètre active la fonctionnalité d'authentification Windows pour le client de bureau.	true	YES	false
aioc.auth.windows	Ce paramètre active l'authentification Windows pour l'interface Web GUI de Kron PAM.	true	YES	false
rdp.idle.session.timeout.warning.before.min	Ce paramètre est utilisé pour configurer l'affichage de la fenêtre contextuelle d'avertissement.	5	YES	5
rdp.hide.top.menu.by.default	Lorsque ce paramètre est défini sur true, la barre dans les connexions RDP sera masquée par défaut.	true	NO	false
aioc.portal.client.ip.header	Cette propriété sera disponible. La valeur par défaut est null, donc aucun en-tête ne sera vérifié pour détecter l'IP du client (l'IP source TCP sera utilisée par défaut). Vous pouvez la définir avec la valeur "X-Forwarded-For".	X-Forwarded-For	OUI	null
session.record.encoded	Ce paramètre permet de stocker de manière sécurisée les enregistrements vidéo RDP/VNC et SSH dans un format	true	NON	false



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
ryption.enabled	chiffré dans une base de données. Les enregistrements restent chiffrés jusqu'à ce que l'utilisateur les relise, garantissant ainsi la protection et la confidentialité des données.			
sc.rdp.reason.mail.recipient	Si le paramètre reasonRequiredForConnection est défini et si ce paramètre est disponible dans systemconfig, la raison écrite sera envoyée par e-mail aux destinataires. Plus d'une adresse e-mail peut être utilisée, avec une virgule comme séparateur.	kron@krontech.com,pam@kron.com		
NON	N/A			
ssh.client.kex.algorithms	Ce paramètre spécifie les algorithmes d'échange de clés utilisés par le client SSH lorsque les comptes Vault établissent des connexions SSH. Les algorithmes d'échange de clés déterminent comment le client et le serveur négocient une clé secrète partagée pour la session. .	ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group-exchange-sha256,diffie-hellman-group18-sha512,diffie-hellman-group17-sha512,diffie-hellman-group16-sha512,diffie-hellman-group15-sha512,diffie-hellman-group14-sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1		N/A
ssh.client.hostkey.algorithm	Ce paramètre spécifie les algorithmes de clés hôtes utilisés par le client SSH pour authentifier le serveur lors de l'établissement d'une connexion SSH au compte Vault.	ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp521-cert-v01@openssh.com,ssh-ed25519-cert-v01@openssh.com,rsa-sha2-512-cert-v01@openssh.com,rsa-sha2-256-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-ed25519,sk-ecdsa-sha2-nistp256@openssh.com,sk-ssh-ed25519@openssh.com,rsa-sha2-512,rsa-sha2-256,ssh-rsa,ssh-dss		N/A
ssh.client.encryption.algorithms	Ce paramètre spécifie les algorithmes de chiffrement utilisés par le client SSH pour chiffrer les données transférées entre le client et le serveur.	<u>chacha20-poly1305@openssh.com</u> , <u>aes128-ctr</u> , <u>aes192-ctr</u> , <u>aes256-ctr</u> , <u>aes128-gcm@openssh.com</u> , <u>aes256-gcm@openssh.com</u> , <u>aes128-cbc</u> , <u>aes192-cbc</u> , <u>aes256-cbc</u> , <u>arcfour256</u> , <u>arcfour128</u> , <u>3des-cbc</u> , <u>blowfish-cbc</u>		N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
ssh.client.algorithm	Ce paramètre spécifie les algorithmes de code d'authentification de message (MAC) utilisés par le client SSH afin de garantir l'intégrité et l'authenticité des données.	<u>hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1,hmac-md5,hmac-sha1-96,hmac-md5-96</u>		N/A
aioc.rest.api.rate.limiter.actif	Paramètre indiquant si la valeur limite est active ou passive	<u>Vrai</u>	NO	true
aioc.rest.api.rate.limiter.threshold	Le nombre de requêtes.	<u>Nombre de requêtes</u>		300
aioc.rest.api.rate.limiter.unit	Le futur des requêtes sera défini comme quotidien, horaire, à la minute ou à la seconde	<u>Jour/Heure/Minute/Seconde</u>		minute
rap.cloud.server	Ce paramètre définit l'adresse du portail Remote Access Portal. Le paramètre peut être défini comme une URL avec IP (par exemple, https://34.234.69.53/connect) ou comme une URL avec un nom de domaine (par exemple, https://cloudpam.com/connect)	http://cloudpam.com/connect	NO	http://localhost:7777/connect
rap.rdp.session.duration.limit.warning	Ce paramètre définit combien de minutes avant l'expiration de la session RDP l'avertissement de délai d'expiration sera envoyé sur le portail Remote Access Portal.	1	NO	1



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
.before.min				
rap.ssh.session.duration.limit.warning.before.min	Ce paramètre définit combien de minutes avant l'expiration de la session SSH l'avertissement de délai d'expiration sera envoyé sur le portail Remote Access Portal.	1	NO	1
rap.http.session.duration.limit.warning.before.min	Ce paramètre définit combien de minutes avant l'expiration de la session HTTP/HTTPS l'avertissement de délai d'expiration sera envoyé sur le portail Remote Access Portal.	1	NO	1
rap.token.expiration.period	Ce paramètre indique la durée de vie d'un jeton et est utilisé pour empêcher la création de liens d'invitation RPAM à long terme.	1	NO N	1
rap.sms.http.url	L'URL est utilisée pour envoyer des SMS via HTTP pour les jetons Remote Privileged Access Management.	https://api.iletimerkezi.com/v1/send-sms	NO N	N/A
rap.sms.http.body	Le contenu du message SMS lors de l'utilisation du protocole HTTP pour les jetons Remote Privileged Access Management.	<pre><request><authentication> <username>XXXphonenumbersXXX</username> <password>XXXpasswordXXX</password> </authentication><order><sender>KRON</sender> <sendDateTime></sendDateTime><message><text></pre>	NO N	N/A



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
		<u><![CDATA[</u> <u>Cher %userEid%,</u> <u> Veuillez utiliser le code d'accès ci-</u> <u>dessous pendant la phase de connexion de votre</u> <u>connexion Secure Remote Access.</u> <u> Code d'accès : %passcode%</u> <u> Connexion Secure Remote Access</u> <u>(accès sur navigateur Web) : %connURL%]></u> <u></text><receipients><number>%phoneNumber%</u> <u></number></receipients></message></order></u> <u></request></u>		
rap.s ms.s mpp.b ody	Le contenu du message SMS lors de l'utilisation du protocole SMPP pour les jetons Remote Privileged Access Management.	<u>---Valeur du corps SMPP---</u>	NO N	N/A
rap.s ms.htt p.hea ders	Les en-têtes inclus dans le SMS pour les jetons de Remote Privileged Access Management.	<u>Content-Type:text/xml</u>	NO	Content-Type:text/xml
rap.s ms.htt p.enc oding	Un encodage de caractères utilisé dans le SMS pour les jetons de Remote Privileged Access Management.	<u>UTF-8</u>	NO	UTF-8
rap.s ms.htt p.met hod	La méthode HTTP utilisée dans le SMS pour les jetons de Remote Privileged Access Management.	<u>POST, GET</u>	NO	POST
rap.s ms.htt p.deli miter	Le caractère délimiteur utilisé dans le SMS pour les jetons de Remote Privileged Access Management.	<u>&</u>	NO	&
rap.s ms.ch annel	Le canal SMS utilisé pour les jetons de Remote Privileged Access Management.	<u>http, smpp</u>	NO	http



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
rap.client.enabled	Le paramètre MFA pour accéder à Remote Access Portal. Si le paramètre est défini sur true, l'utilisateur doit saisir un code OTP à 6 chiffres sur la page de connexion de Remote Access Portal.	<u>vrai/faux</u>	NO	false
rap.password.characters.count	Ce paramètre indique combien de caractères sont utilisés dans la définition du code d'accès. La valeur de ce paramètre doit être numérique, et la valeur par défaut est 8. Si l'administrateur système définit ce paramètre à 4 ou moins, le code d'accès est créé avec 4 caractères.	<u>12</u>	NO	<u>8[DT1]</u>
rap.password.only.numeric.text	La valeur de ce paramètre doit être un booléen, et la valeur par défaut est false. Si la valeur de ce paramètre est définie à true, le code d'accès contient uniquement des valeurs numériques ; cependant, si la valeur de ce paramètre est définie à false, le code d'accès contient des valeurs alphanumériques.	<u>vrai / faux</u>	NO	false
show.wireto sessionWarning	Ce paramètre, dont la valeur par défaut est true, est ajouté au System Config. Manager afin d'afficher l'avertissement "Votre session est surveillée par l'administrateur" à l'utilisateur qui a démarré la session SSH/Telnet ou RDP/VNC lorsque l'administrateur se connecte à une session active via wire to session. Lorsque la valeur du paramètre est définie à false, aucun avertissement n'apparaît sur l'écran de l'utilisateur qui démarre la session. Avec cette fonctionnalité, même si l'utilisateur ne voit pas l'avertissement, le journal « Votre session a été surveillée	<u>vrai/faux</u>	NO	true



Nom du paramètre	Description	Exemple de valeur de paramètre	Requis	Valeur par défaut
	par l'administrateur » s'affiche dans le Journal de session.			
resources.of.containers	Avec ce paramètre, il est défini si le Container utilisé pour Http Proxy sera exécuté sur la machine locale ou sur une autre machine.	Remote/Local /All	NO	Local/Remote/All
rdp.time.striction.warning.before.min	Spécifie le nombre de minutes avant l'heure de fin planifiée de la session auquel un message d'avertissement est affiché à l'utilisateur dans une session RDP. Ce paramètre fait partie de la politique de restriction de temps et permet aux administrateurs de configurer à quel moment les utilisateurs sont avertis avant la fin de leur session.	7	NO	1
aioc.hide.on.behalf.of.tenant.switch	Ce paramètre est utilisé dans les tenants hôtes et non hôtes pour masquer la fonctionnalité de changement de tenant, si le paramètre est défini sur true. La valeur par défaut de ce paramètre est false , ce qui signifie que tant que ce paramètre n'est pas défini sur true, la fonctionnalité d'accès pour le compte d'un tiers est utilisée comme dans le scénario habituel.	true / false	NO	false
sapm.duallock.option.hide	Détermine si la case à cocher Activer Duallock sur l'écran de définition du Coffre-fort apparaîtra ou non. Si elle est non définie comme valeur par défaut ou définie sur Faux , cette valeur apparaîtra à l'écran; si elle est définie sur Vrai , cette case à cocher n'apparaîtra pas à l'écran.	true/false	No n	N/A
connector.tunnel.p	Ce paramètre est utilisé pour définir la plage de ports du tunnel du connecteur du tenant.	10000-11000	NO	10000-11000



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
ort.range				
connector.tunnel.subnet.base	Ce paramètre est utilisé pour définir la valeur de base du sous-réseau du connecteur du tenant.	192.168.0.0	NO	192.168.0.0
connector.heartbeat.check.interval	Ce paramètre est utilisé pour définir l'intervalle de vérification du heartbeat. Si les messages heartbeat ne peuvent pas être envoyés au serveur Kron PAM pendant le nombre de minutes spécifié par cette valeur, le statut du connecteur devient Échec . (Par exemple, si cette valeur est définie sur 5, si aucun message heartbeat n'est reçu pendant 5 minutes consécutives, le statut du connecteur devient Échec .)	5	NO	5
connector.port.range	Ce paramètre est utilisé pour définir la plage de ports virtuels pour les appareils accessibles par le connecteur du tenant. Ces ports ne sont pas utilisés pour accéder à ces appareils depuis l'extérieur du connecteur, mais uniquement pour le mappage des appareils.	40000-50000	NO	40000-50000_1
netright.job.master.instance.name				
netright.job.master.instance.name				



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
me.\$region_name				
mfa.provider	Ce paramètre définit le fournisseur MFA externe utilisé par le système. Si la valeur de ce paramètre est définie sur radius, le système lance le processus MFA via un serveur RADIUS.	radius	NO	
mfa.external.provider.radius.addresses	Ce paramètre est utilisé pour définir les informations d'hôte et de port du ou des serveurs RADIUS. Plusieurs définitions peuvent être spécifiées sous forme de liste séparée par des virgules pour assurer la redondance ; si l'un est inaccessible, le suivant sera essayé.	10.20.30.40	NO	
mfa.external.provider.radius.secret	Ce paramètre, qui doit être défini en sélectionnant la case à cocher Encryption, contient la clé secrète partagée utilisée pour la communication sécurisée entre Kron PAM et le serveur RADIUS.	*****	NO	
mfa.external.provider.radius.timeout	Ce paramètre définit le temps maximal, en millisecondes, pendant lequel le système attend une réponse du serveur RADIUS avant l'expiration du délai.	3000	NO	3000
mfa.external.provider.retry-count	Ce paramètre détermine le nombre de tentatives de nouvelle tentative que le système effectuera si aucune réponse n'est reçue du serveur RADIUS dans le délai spécifié.	3	NO	3
mfa.external.	Ce paramètre spécifie le type de protocole d'authentification utilisé dans le	pap	NO	pap



Nom du paramètre	Description	Exemple de valeur de paramètre	Requise	Valeur par défaut
provider.radius.protocol	paquet de requête RADIUS.			
mfa.external.provider.radius.user-field	Ce paramètre détermine quel attribut utilisateur est envoyé dans le champ de nom d'utilisateur de la requête RADIUS. Les valeurs disponibles incluent username, phone, email ou UserPrincipalName.	username	NO	username
mfa.external.provider.radius.nas-id	Ce paramètre est utilisé pour définir l'attribut RADIUS NAS-Identifiant dans le paquet de requête, qui est fourni lorsqu'il est spécifiquement requis par le serveur RADIUS.			
mfa.external.provider.radius.nas-ip-addresses	Ce paramètre définit l'attribut RADIUS NAS-IP-Address envoyé dans le paquet de requête afin d'identifier l'adresse IP du client auprès du serveur RADIUS.		NO	
mfa.external.provider.radius.factor	Ce paramètre, lorsque sa valeur est définie sur push, permet au système d'utiliser l'authentification multifacteur spécifiquement via des notifications push.	push		
allow.config.reservation.time.b	Ce paramètre est utilisé pour permettre à l'approbateur de modifier l'heure de réservation dans les applications mobiles avant d'approuver les demandes.	true / false	NO	false



Nom du paramètre	Description	Exemple de valeur de paramètre	Re dé ma rra ge re qui s	Valeur par défaut
y.approver				
tenant.expiration.warning.before.day	Ce paramètre est utilisé pour afficher un message d'avertissement dans l'interface Web GUI, XX jours avant l'expiration du locataire.	15	NO	15
aioc.login.different.methods.enabled	Ce paramètre est utilisé pour ajouter un nouveau bouton appelé « Login with Different Methods » sur le serveur Kron PAM et le client de bureau. Le nouveau bouton permet aux utilisateurs de se connecter au système autrement que par le processus de connexion conventionnel (par exemple, SAML).	true / false	NO	false
aioc.hide.on.behalf.of.tenant.switch	Ce paramètre est utilisé à la fois dans les environnements de locataire et d'hôte pour masquer la fonctionnalité de changement de locataire si le paramètre est défini sur true. La valeur par défaut de ce paramètre est faux , ce qui signifie que tant que ce paramètre n'est pas défini sur vrai , la fonctionnalité d'accès au nom d'un autre utilisateur est utilisée comme dans le scénario habituel.	true / false	YES	false



2.1. Plan d'exercice de laboratoire — Groupe LF HOL

Durée du labo : 1 jour (7h en salle, ~5h pratiques)

Temps total des exercices estimé : ~245 min (≈ 4h 5min)

Temps tampon (Q&R, transitions, problèmes techniques) : ~55 min

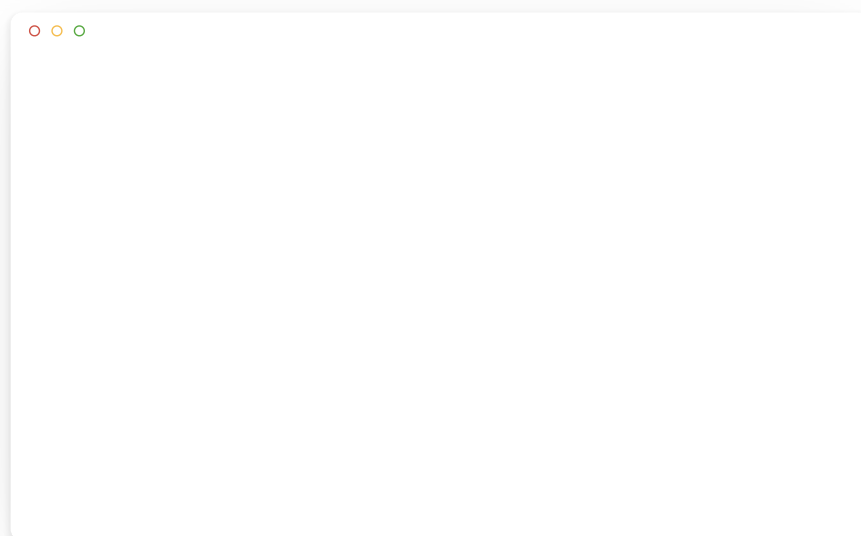
Nombre total de modèles requis : 16

Aruba 660-330 Linuxme3s



Your Guide to The CTI Module

- The CTI Module at-a-glance
- Identify threats with the Intelligence Browser
- Package research into a report with Threat Flow
- Operationalize IOCs using the IOC Feed
- Safely Investigate Suspicious Files/URLs with the Sandbox



[LEARN MORE](#)

Structure du module

Module 1 — Fondations des parcours

Durée : ~70 min

Concepts enseignés : canevas de parcours, événements d'entrée, nœuds d'attente (statiques + dynamiques), séparation conditionnelle (NTB/ETB), canal push + gestion explicite du consentement, qualification de l'audience avec Customer AI, expérimentation de parcours

Exercices

- E1.1 — Créer le parcours + configurer l'événement d'entrée + premier nœud d'attente (30 min)



- E1.2 — Séparation NTB/ETB + push « Bentornato » + option de secours pour le consentement + attente de convergence (25 min)
- E1.3 — Condition d'audience + expérimentation de parcours + attente dynamique (15 min)

Modèles préconfigurés pour ce module

- T1.1 — Courriel « Benvenuto » NTB (complet — modèle de courriel répétitif, les participants construiront un courriel complet dans le module 2)
- T1.2 — Courriel de secours ETB (complet — même logique)
- T1.3 — Critères de sortie configurés sur le parcours (intégrés dans E1.1 comme sous-étape guidée)

Remarque sur les nœuds d'attente : les nœuds d'attente se trouvent dans le canevas du parcours et ne peuvent pas être préchargés comme modèle — les participants les insèrent dans le cadre de l'exercice pertinent. Configuration mécanique, mais inévitable.

Module 2 — Analyse approfondie du courriel « Recap simulazione »

Durée : ~90 min

Concepts enseignés : modèle maître central + verrouillage du contenu, fragments de contenu (HTML, expression, personnalisable), contenu conditionnel, génération d'actifs Firefly avec jetons de marque, élément de décision + création de règles, intégration du bloc de décision, expérimentation de contenu (objet A/B), optimisation du moment d'envoi, score d'alignement AJO Brands

Exercices

- E2.1 — Modèle maître + contenu conditionnel + image héro Firefly (35 min)
- E2.2 — Créer 1 élément de décision + 1 règle, puis intégrer la politique de décision complète + prévisualiser les résultats par profil (35 min)
- E2.3 — Expérimentation de contenu + optimisation du moment d'envoi + révision AJO Brands (20 min)

Modèles préconfigurés pour ce module

- T2.1 — Catalogue d'offres : **7 sur 8** éléments de décision + **6 sur 7** règles de décision + méthode de classement + stratégie de sélection + politique de décision (préconstruites et approuvées). Les participants créent le 8e élément de décision + la 7e règle dans E2.2 — cela enseigne le flux de création sans les 3 à 4 h de travail massif sur le catalogue.
- T2.2 — Structure du fragment personnalisable « Regional CTA » (préconstruite; les participants le connectent au jeu de données de recherche dans E2.1 — cela enseigne le modèle sans le code structurel répétitif)
- T2.3 — 4 fragments de contenu (HTML de marque d'en-tête, HTML de marque de pied de page, expression de clause de non-responsabilité, expression Saluto) — préconstruits et verrouillés; les participants observent le verrouillage du contenu, ils ne le configurent pas
- T2.4 — **4 sur 5** actifs d'image héro (générés par Firefly, préchargés dans la bibliothèque d'actifs). Les participants génèrent le 5e dans E2.1 — cela enseigne le flux Firefly + jetons de marque sur un seul actif.

Module 3 — Gouvernance



Durée : ~25 min

Concepts enseignés : limitation de fréquence (règles métier et ensembles de règles), gestion des conflits (priorité des parcours — discussion du facilitateur seulement), alertes opérationnelles AJO, guide d'arrêt manuel (point de discussion)

Exercices

- E3.1 — Limitation de fréquence : configurez la règle `Limitation des e-mails` dans l'ensemble de règles `Règles marketing`, puis activez l'ensemble de règles (8 min)
- (E3.2 supprimé — la gestion des conflits est maintenant un point de discussion de 5 min dirigé par le facilitateur entre E3.1 et E3.3. Voir les notes du facilitateur.)
- E3.3 — Alertes : configurer l'abonnement aux alertes opérationnelles AJO; CJA mentionné comme surface d'alerte alternative (12 min)

Modèles préconfigurés pour ce module

- T3.1 — Parcours « Promo carta credito » (priorité moyenne, audience de prospects chauds hypothécaires) — préconstruit; utilisé par le facilitateur pour *montrer* le scénario de conflit pendant la discussion E3.2 (aucune configuration par les participants).

Remarque : les tests de suppression au niveau du profil pour E3.1 ne sont pas réalisables dans l'environnement de laboratoire (temps + surcharge de configuration de l'état). Le comportement de suppression est expliqué comme point de discussion. La suppression des conflits est couverte uniquement en discussion dirigée par le facilitateur — voir les notes du facilitateur. Un exercice CJA dédié est reporté au Module 4.

Module 4 — Test et analyses

Durée : ~55 min

Concepts enseignés : mode de test de parcours, vérification du branchement par profil, lecture des rapports d'expérimentation, rapports de parcours AJO, alertes planifiées CJA

Exercices

- E4.1 — Mode de test de parcours avec 5 profils de test (25 min)
- E4.2 — Lire le rapport d'expérimentation (chemin A vs B) + rapports de parcours AJO (15 min)
- E4.3 — Alerte planifiée CJA sur la baisse des conversions (15 min)

Modèles préconfigurés pour ce module

- T4.1 — 5 profils de test préchargés (ETB-young-Stockholm, ETB-senior-Göteborg, NTB-prime-Skåne, NTB-Östergötland, ETB-premium-Halland)
- T4.2 — Espace de travail CJA préconnecté au jeu de données AJO, amorcé avec suffisamment de données historiques pour un aperçu significatif des alertes de baisse des conversions dans E4.3

Résumé du temps



Module	Exercices	Temps estimé
Module 1 — Fondements des parcours	E1.1, E1.2, E1.3	70 min
Module 2 — Récapitulatif Simulazione email	E2.1, E2.2, E2.3	90 min
Module 3 — Gouvernance	E3.1, E3.3 (+ discussion dirigée par le facilitateur E3.2)	25 min
Module 4 — Test et analyses	E4.1, E4.2, E4.3	55 min
Total	10 exercices	240 min

Marge pour les transitions, Q&R, problèmes techniques : ~50 min → tient dans 5 h de pratique. ✓

Justification des principales décisions de sélection

- **Courriel « Benvenuto » + solution de secours par courriel ETB → modèles.** Les participants configureront un courriel complet dans le Module 2. Construire d'abord deux courriels supplémentaires ajoute plus de 30 min de répétition avant d'atteindre l'exercice principal. Valeur pédagogique : marge nulle.
- **Catalogue d'offres (8 éléments + règles + classement) → modèle.** La construction correcte du catalogue prend au minimum 3-4 h. L'exercice important consiste à *intégrer* la politique de décision dans un courriel et à *lire* le résultat par profil — c'est E2.2. Le catalogue est un prérequis, pas la leçon.
- **Fragments de contenu (Header, Footer, Disclaimer, Saluto) → modèles, verrouillés.** Les participants *observent* le verrouillage du contenu dans E2.1 — ils voient qu'ils ne peuvent pas modifier ces zones. C'est la leçon. Les construire ajoute du temps sans apport conceptuel.
- **Expérimentation de chemin (E1.3) → exercice court.** La configuration est simple (définir l'attente, attribuer le chemin A/B, configurer la condition de mise à l'échelle automatique). La valeur pédagogique consiste à comprendre *pourquoi* la structure est conçue ainsi, et non les étapes de l'interface utilisateur. 10 min est approprié.
- **Webhook d'action personnalisée (Étape 13a) → omis des exercices.** Trop spécifique à l'infrastructure; nécessite un point de terminaison CRM actif pour être pertinent; n'ajoute aucune valeur conceptuelle AJO au-delà de ce que les participants apprennent déjà avec la condition de parcours (Étape 12). Mentionné comme point de discussion dans le Module 4.
- **E-mail « Avanza online » (Étape 13b) → omis en tant qu'exercice autonome.** Intégré dans E4.1 (mode test) comme résultat observé. La génération de texte AJO Brands est couverte dans E2.3 — inutile de la répéter.
- **Branche de géolocalisation → retirée du laboratoire.** Nécessite une configuration Places SDK qui ajoute une complexité importante à l'environnement pour un faible bénéfice conceptuel dans le temps disponible.
- **Exercice CJA (E4.3) → placé dans le Module 4, et non le Module 3.** Les alertes CJA relèvent des insights, et non de la gouvernance opérationnelle. Les alertes AJO (E3.3) sont l'outil opérationnel; CJA sert à la détection analytique/planifiée des tendances de jeux de données. Deux surfaces distinctes, deux contextes distincts.
- **Tests de suppression au niveau du profil dans E3.1 et E3.2 → retirés.** Non réalisables dans les délais de l'environnement de laboratoire. Le comportement de suppression est abordé comme sujet de discussion pendant les exercices.
- **E3.2 (gestion des conflits) → rétrogradé d'exercice pratique à discussion dirigée par l'animateur.** Définir un menu déroulant de priorité prend 30 secondes; le contenu conceptuel est la *conversation* sur la manière dont les décisions de priorité sont prises entre des campagnes concurrentes. Cette conversation mérite 5 minutes du temps de l'animateur, et non un créneau pratique de 10 minutes. Cela libère du budget et retire un exercice de suppression empiilée que les participants ne pouvaient de toute façon pas vérifier.



Modèles — liste de vérification de configuration du bac à sable avant le laboratoire

Dans l'ordre des dépendances (Alex / administrateur du bac à sable suit cette séquence) :

1. Schéma XDM : tous les champs de profil + événement (incl. `conversionPropensityScore` , `lfCompanyId` , `branchAssigned`)
2. Jeu de données de recherche AEP « LF Company Registry » (5 sociétés LF : Stockholm, Göteborg, Skåne, Östergötland, Halland)
3. 5 profils de test chargés avec tous les attributs requis (scores de propension préremplis de manière statique)
4. Audience « Mortgage hot leads » configurée (comportemental + score >= 60)
5. Customer AI — vérifier les scores de propension sur les profils de test; si le modèle n'est pas entraîné, des valeurs statiques sont en place
6. Profil AJO Brands : LF Group (ton, palette, logo, ressources)
7. **4 sur 5** ressources d'image principale générées (Firefly + marque LF Group) → chargées dans la bibliothèque de ressources (les participants génèrent la 5e dans E2.1)
8. Fragment de contenu HTML : « Header brand » (verrouillé par le modèle maître)
9. Fragment de contenu HTML : « Footer brand » (verrouillé par le modèle maître)
10. Texte du fragment d'expression : « Disclaimer legale » (verrouillé par le modèle maître)
11. Texte du fragment d'expression : « Saluto » (verrouillé par le modèle maître)
12. Modèle maître « Mortgage Comms » — structure avec verrouillage du contenu et emplacements vides pour E2.1
13. Fragment personnalisable « Regional CTA » — structure uniquement (les participants câblent la recherche dans E2.1)
14. Catalogue d'offres : **7 sur 8** éléments de décision (approuvés) + **6 sur 7** règles de décision + méthode de classement + stratégie de sélection + politique de décision (les participants créent l'élément + la règle manquants dans E2.2)
15. ⚠ Vérifier l'activation de la politique de décision pour le canal Email (Limited Availability) avec le représentant Adobe avant le laboratoire
16. E-mail « Benvenuto » NTB — terminé (T1.1)
17. E-mail de secours ETB — terminé (T1.2)
18. Parcours « Promo carta credito » — terminé, priorité moyenne (pour l'exercice de conflit du Module 3)
19. Règle métier de limitation de fréquence : catégorie `hypothèque` , max 3 / 7 jours — **pas encore active** (les participants l'activent dans E3.1)
20. Abonnement aux alertes AJO sur le parcours hypothécaire — **pas encore configuré** (les participants le configurent dans E3.3)
21. Espace de travail CJA connecté au jeu de données AJO, alimenté avec des données historiques de conversion suffisantes pour un aperçu significatif des alertes de baisse de conversion (E4.3) et pour le contexte des rapports de parcours AJO (E4.2)

Liste des exercices pour Myrna



ID	Titre	Module	Temps estimé	Fonctionnalité(s) clés d'AJO
E1.1	Créer le parcours + l'événement d'entrée + la première attente	1	30 min	Canvas de parcours, événement d'entrée, payload XDM, nœud d'attente (statique), critères de sortie, règle de réentrée
E1.2	Division NTB/ETB + push + consentement + attente de convergence	1	25 min	Condition de profil, canal push, condition de consentement + fallback, tokens multi-marques, attente de convergence de branche
E1.3	Condition d'audience + expérimentation de parcours + attente dynamique	1	15 min	Qualification d'audience, activité d'optimisation, mise à l'échelle automatique du gagnant, attente dynamique
E2.1	Modèle maître + contenu conditionnel + image Firefly	2	35 min	Modèle maître, verrouillage du contenu, fragments, fragment personnalisable, contenu conditionnel, notation lookup, Firefly + tokens de marque
E2.2	Créer 1 élément de décision + 1 règle, intégrer une politique + aperçu	2	35 min	Création d'éléments de décision, règle de décision, intégration de politique de décision, aperçu d'offre par profil, formule de classement
E2.3	Expérimentation de contenu + STO + marques	2	20 min	Sujet A/B, expérimentation de contenu, optimisation de l'heure d'envoi, score d'alignement des marques AJO
E3.1	Règle métier de limitation de fréquence (configuration uniquement)	3	8 min	Règles métier, limitation par catégorie
E3.2	Gestion des conflits — définir et vérifier les priorités — supprimé; discussion avec le facilitateur uniquement	3	Discussion de 5 min	Concept de priorité de parcours (sans pratique)
E3.3	Alertes opérationnelles AJO (CJA mentionné uniquement)	3	12 min	Abonnement aux alertes AJO, guide d'arrêt manuel
E4.1	Mode test du parcours avec 5 profils	4	25 min	Mode test, vérification des résultats par profil, embranchement
E4.2	Rapport d'expérimentation + rapports de parcours	4	15 min	Rapport d'expérimentation AJO, rapport en direct du parcours
E4.3	Alerte planifiée CJA sur une baisse de conversion	4	15 min	Espace de travail CJA, alertes planifiées, métriques de conversion, aperçu des alertes



3. À propos

3.1. Problème de MedScout avec le boling et les couleurs

Exemple de texte où j'appliquerai le gras/la couleur

Exemple de texte où j'appliquerai le gras/les couleurs

Exemple de texte où j'appliquerai le gras/les couleurs

Exemple de texte où j'appliquerai le gras/les couleurs

Exemple de texte où j'appliquerai le gras/les couleurs



3.1.1. Problème de table des matières Cadent

Les notes de version fournissent des mises à jour sur les nouvelles fonctionnalités et les changements apportés à la plateforme Cadent dans le cadre d'une version donnée. Nous mettons les nouvelles notes de version à disposition dès qu'une nouvelle version est publiée.

✓ 22 avril 2026

Améliorations de la page Audiences

La page Audience a été mise à jour afin d'améliorer le flux de travail et de vous aider à être plus efficace. De nouvelles options de tri et de filtrage sont désormais disponibles. De plus, vous pouvez télécharger un fichier Excel contenant une liste de tout le contenu de la page Audiences. Configurez la page pour qu'elle reflète votre flux de travail individuel et elle conservera cette configuration lorsque vous reviendrez plus tard sur la page. [En savoir plus.](#)

Campaign	Advertiser	Status	Budget	Start Date	End Date
Auto_A_Campaign_Cadent_S_Mar20	18 Brokers	Live	\$212,218.45	03/10/2026	08/10/2026
Auto_A_Campaign_P_Address_Mar20	18 Brokers	Paused	\$1,988.00	03/10/2026	05/10/2026
Auto_A_Campaign_P_Address_Mar20	AutoZone	Live	\$815,000.00	04/08/2026	05/10/2026
Auto_Marketing_Campaign_Cadent_S_Mar20	1 Vision - Choclette	Completed	\$99,218.00	02/10/2026	05/10/2026
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Paused	\$18,118.00	03/10/2026	04/10/2026
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Paused	\$2,810.00	02/10/2026	02/10/2026
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Completed	\$15,008.00	07/10/2025	03/10/2026
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Completed	\$81,111.48	03/10/2026	03/10/2026
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Paused			
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Paused			
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Paused			
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Paused			
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Paused			
Auto_Marketing_Campaign_Cadent_S_Mar20	18 Brokers	Paused	\$8,988.00	12/10/2025	12/10/2026



✓ 21 avril 2026

Fonctionnalité Ad Manager

KPI de coût par action pour les éléments de ligne

Ad Manager prend désormais en charge les modèles de coût par action (CPA), qui sont des modèles spécifiques à une campagne créés automatiquement lorsque vous sélectionnez **CPA** comme KPI pour un élément de ligne, puis mappez les activités de la campagne.

Les modèles CPA sont un outil clé pour générer de solides résultats KPI en fournissant des modèles basés sur la performance entraînés à partir de données de conversion, comme une vente ou une installation d'application. Cela diffère des autres modèles, comme les modèles de clics, qui optimisent les clics indépendamment du fait qu'ils mènent ou non à une conversion.

The screenshot shows the 'ADD KPI' interface. At the top, it says 'ADD KPI' followed by a blue plus icon and 'KPI'. Below this, there is a section with a circular icon containing the number '1'. To its right, there is a label '* KPI' and a dropdown menu currently showing 'Select...'. A hand cursor is pointing at the 'Cost Per Action' option in the dropdown list. To the right of the dropdown is a label '* Target' and a text input field containing the number '10'. The dropdown list includes the following options: 'Cost Per Rich Media Expansion', 'Conversion Rate', 'Cost Per Action' (highlighted), 'Cost Per Action Retargeting', and 'Cost Per Action Look-'. A scrollbar is visible on the right side of the dropdown menu.

Cost Per Action est un KPI d'élément de ligne



✓ 20 avril 2026

Fonctionnalités d'Ad Manager

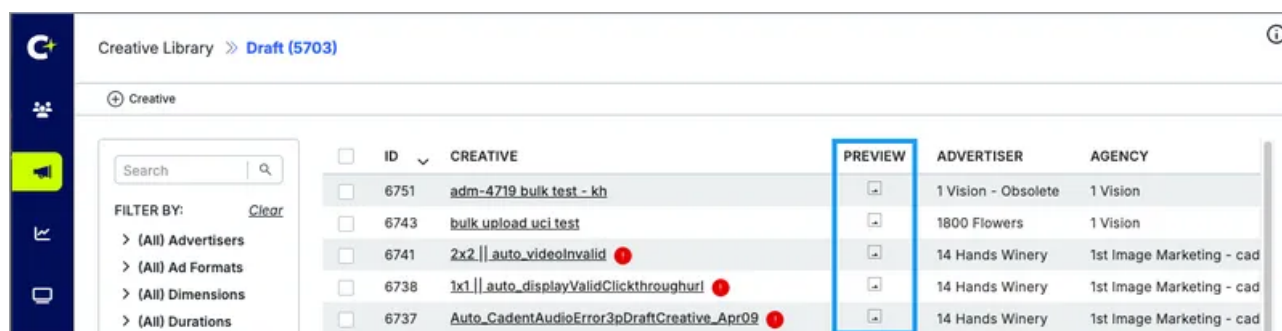
Prévisualisez et testez vos créations publicitaires brouillon

Vous pouvez maintenant prévisualiser et tester vos créations publicitaires brouillon. Cette étape de contrôle de la qualité garantit que vos créations publicitaires brouillon s'affichent et fonctionnent exactement comme prévu avant que vous **approuviez vos créations publicitaires brouillon**.

En savoir plus sur [la prévisualisation et le test des créations publicitaires](#).

Comment prévisualiser et tester une création publicitaire

1. Accédez à la Creative Library, puis cliquez sur l'icône **Aperçu** à côté de la création publicitaire. Cela ouvre le panneau de prévisualisation et de test.



Icône Preview dans la Creative Library

2. Ici, vous pouvez effectuer les actions suivantes :
 - Cliquez sur la prévisualisation de la création publicitaire pour tester l'URL de clic.
 - Afficher la réponse d'enchère.





Creative Lib

1 Selected | G

Search

FILTER BY:

> (All) Ad

> (All) Ad

> (All) Dir

> (All) Dur

FOLDERS:

Availabl

Draft (5

VIDEO CREATIVE: ADM-4719 BULK TEST - KH

EDIT CREATIVE INFORMATION

X



OPEN TEST PAGE

Bid Response

Environment

Rubicon

SITE APP

Environment

SITE APP

CANCEL

SAVE

Présentation du journal Change History pour les éléments de ligne

Vous pouvez maintenant consulter un journal complet de toutes les modifications apportées à vos éléments de ligne. Accédez simplement au nouvel onglet **Historique des modifications** sur la page Line Item Details. Vous y trouverez un journal détaillé de chaque modification, y compris :

- Date à laquelle la modification a été effectuée (heure de l'Est)
- Attribut modifié
- Ancienne valeur
- Nouvelle valeur
- Personne ayant effectué la modification

Mindys campaign >>
Line Item Details: Mindy's campaign_CTV (1000016730)

Details Change History

Change History 2

Q Search...

Changed On (Eastern Time) Updated Attributes Changed By

04/15/2026 03:25:05 PM KPIs Mindy Carpenter

04/15/2026 03:25:05 PM Basics Frequency Cap Budget & Pacing 1 More Mindy Carpenter

Attribute	Old Value	New Value
Budget	--	\$50,000
Channel	--	CTV
05.01.2026 - 06.01.2026 / spend cap \$25,000 / imp		
04/15/2026 03:25:05 PM Basics Frequency Cap Budget & Pacing 1 More Mindy Carpenter		
Flight 2	--	06.01.2026 - 07.01.2026 / spend cap \$25,000 / imp
Flight Pacing	--	cap --
Frequency Capping Enabled	--	Automatic, ahead 115%
Metrics Primary Partner	--	False
Name	--	Cadent
Pacing Towards	--	Mindy's campaign_CTV
Status	--	Spend
		Disabled

Onglet Change History

Cette fonctionnalité vous offre une plus grande transparence et facilite le dépannage en fournissant une piste d'audit claire de tous les ajustements.

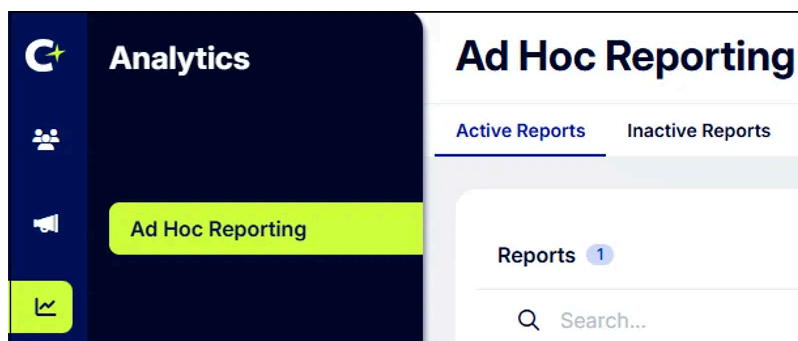
En savoir plus sur le **Journal Change History**.



Fonctionnalités Analytics - Ad Hoc Reporting

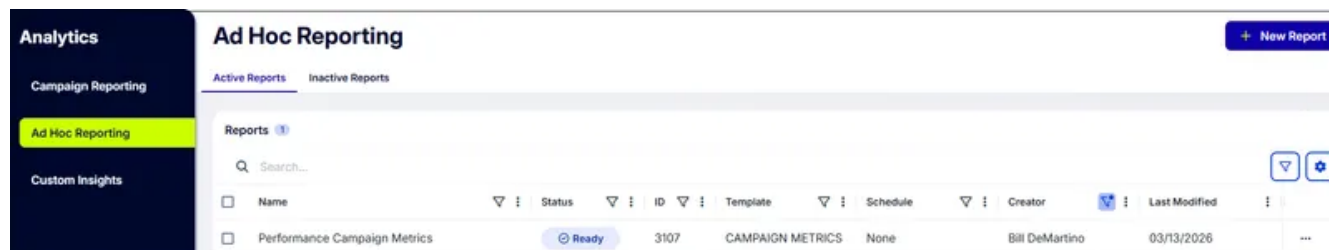
Ad Hoc Reporting a été déplacé

Vous pouvez maintenant trouver Ad Hoc Reporting en cliquant sur l'icône Analytics  dans la barre de navigation de gauche. Auparavant, il se trouvait sous Ad Manager.



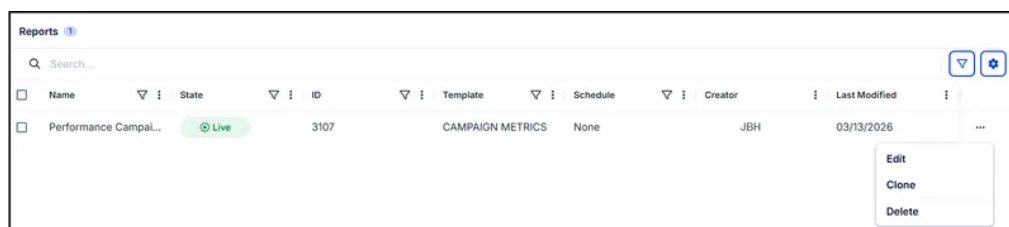
Améliorations de la page Ad Hoc Reporting

La page Ad Hoc Reporting a été mise à jour pour améliorer son flux de travail et son efficacité. Vous avez la possibilité d'ajouter ou de supprimer des colonnes. Des filtres plus puissants permettent d'afficher un ensemble spécifique de rapports et de les trier selon vos besoins. [En savoir plus.](#)



Cloner des rapports pour des modifications plus rapides

Sur la page Ad Hoc Reporting, vous pouvez maintenant cloner un rapport. Le clonage vous permet de gagner du temps lors de la création d'un nouveau rapport en utilisant un rapport existant comme base pour un nouveau rapport. Après avoir créé le clone, vous pouvez ensuite modifier les détails dans le clone afin de répondre aux besoins du nouveau rapport. [En savoir plus.](#)



✓ 6 avril 2026

Nouvelle action groupée : mettre à jour les paramètres d'optimisation pour plusieurs groupes d'annonces

Vous pouvez désormais appliquer des modifications des paramètres d'optimisation à plusieurs groupes d'annonces simultanément, ce qui simplifie votre processus de gestion de campagnes et assure une cohérence entre vos groupes d'annonces. Auparavant, la mise à jour de ces paramètres nécessitait la modification de chaque groupe d'annonces un par un.

Vous trouverez cette option de mise à jour groupée sur les pages suivantes :

- **Détails des campagnes** : pour la liste des éléments de ligne qui y apparaissent
- **Liste des éléments de ligne** : sur la page Campagnes
- **Page de détails des éléments de ligne** : pour la liste des groupes d'annonces qui y apparaissent

Pour mettre à jour l'optimisation des groupes d'annonces de manière groupée :

1. Sélectionnez un ou plusieurs groupes d'annonces. Notez que les groupes d'annonces doivent être associés au même annonceur pour effectuer cette mise à jour groupée.
2. Cliquez sur **Mettre à jour l'optimisation** pour ouvrir le panneau **Optimisation**.
3. Cliquez sur **Enregistrer** pour appliquer vos modifications à tous les éléments sélectionnés.

The screenshot displays the 'Audience Assignment Test Line Item' page. The top navigation bar includes a '1B' icon, a 'Campaign' breadcrumb, and a 'Line Item' breadcrumb. The main header shows the line item name 'Audience Assignment Test Line Item' and a 'Paused' status. A toolbar contains various actions like 'View Bidding Details', 'View Rate Card', '2 Selected', 'Allocate', 'Update Status', 'Clone', 'Delete', 'Update Frequency Cap', 'Update Targeting', 'Update Optimization' (highlighted with a blue box), and 'Update Audience'. Below the toolbar, there are tabs for 'Details' and 'Change History'. The 'Summary' section is expanded, showing three main panels: 'Overview' with metrics like Impressions, CPM (USD), Win Rate (%), KPI 1 - KPI (USD), and VCR (%); 'Pacing (USD)' with a 'Projected' gauge; and 'Insights' with a line graph. Below the summary, there is a table of 'Ad Groups (3)'. The table has columns for ID, AD GROUPS, CHANNEL, ALLOCATION, YESTERDAY (USD), DAILY GOAL EST. (USD), Impressions, and CPI. The first two rows are selected. A sidebar on the left contains filters for 'DATE RANGE' (Preset, Custom, Lifetime) and 'FILTER BY' (All Status).

ID	AD GROUPS	CHANNEL	ALLOCATION	YESTERDAY (USD)	DAILY GOAL EST. (USD)	Impressions	CPI
1000021001	Audience Assignment Test Ad Group 1	CTV	0%	—	0	—	—
1000021002	Audience Assignment Test Ad Group 1...	CTV	0%	—	0	—	—
1000021003	Audience Assignment Test Ad Group 1...	CTV	0%	—	0	—	—
TOTAL	—	—	0%	—	—	—	—

Mise à jour groupée de l'optimisation

Référez facilement vos tarifs avec le visualiseur de grille tarifaire



Nous sommes heureux d'annoncer le lancement du nouveau visualiseur de grille tarifaire dans Ad Manager. Cette nouvelle vue fournit un accès rapide et transparent à votre grille tarifaire, réduisant le temps passé à rechercher vos tarifs.

Pour accéder à la grille tarifaire, cliquez sur **Afficher la grille tarifaire** dans l'un des écrans suivants :

- Détails de la campagne
- Détails des éléments de ligne
- Détails des groupes d'annonces



 Visualiseur de grille tarifaire

Visualiseur de grille tarifaire



✓ 1 avril 2026

Notes de version d'Ad Manager

Problème résolu : écart d'impressions du partenaire principal

La méthodologie utilisée pour calculer la différence entre les impressions suivies par Cadent et les impressions du partenaire principal (par exemple, celles suivies par un serveur publicitaire comme CM360) affichait des chiffres inexacts. Nous avons résolu ce problème, de sorte que la colonne

Écart de PP Imp reflète désormais des chiffres exacts.

Pour afficher la colonne PP Imp Discrepancy :

1. Accédez à la page **Détails de la campagne** ou **Détails des éléments de ligne**.
2. Sélectionnez **Écart PP** dans le sélecteur de vue.

Cadent PROD Self-Service Verification >>

Campaign Details:

Line Item

SPEND (USD)

BUDGET: 40.00
SPEND: 20.41

3 DAYS REMAINING

51%

OVERVIEW

Lifetime Yesterday Today

IMPRESSIONS	CPM (USD)	CLICKS	CTR (%)	VCR (%)
13,064	1.56	43	0.32%	19.54%

LINE ITEMS (2)

PP Discrepancy

Id	Line Item	Impressions	PP Impressions	PP Imp Discrepancy	PP Clicks	PP Conversions
1000000536	Cadent_GCM_Validation_Video	8,280	2526	-227.79	4	-
1000000535	Cadent_GCM_Validation_Dis...	10,667	3031	-251.93	3	-
TOTAL		18,947	5,557	-239.86 %	7	0

DATE RANGE: Preset Custom Lifetime

FILTER BY: Clear

> (All) Status

> (All) Advertisers

SETTINGS

BASICS BUDGET & DATES FREQUENCY CAP

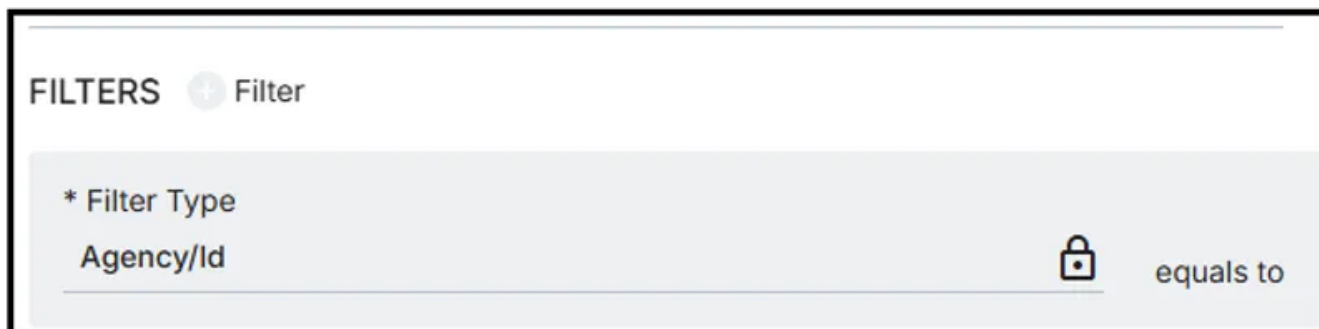


✓ 23 mars 2026

Notes de version d'Ad Manager

Les filtres de rapport pour Ad Hoc Reporting ont été simplifiés

Lors de la création d'un nouveau rapport, vous pouvez désormais filtrer les rapports uniquement par agence. Aucun autre filtre n'est requis. Cela permet de produire des rapports plus complets que ce qui était auparavant possible. [En savoir plus.](#)



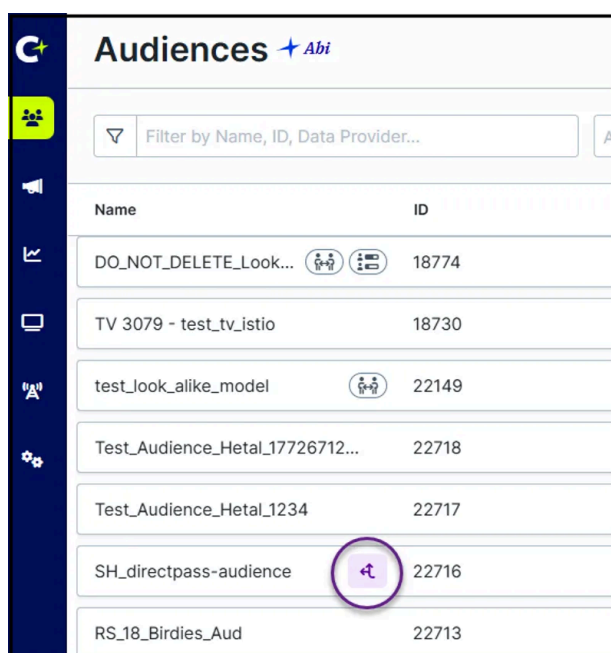
Notes de version d'Audience Manager

Ajout d'icônes DirectPass dans Audience Manager

Une nouvelle icône DirectPass  est maintenant affichée à côté des ensembles de données DirectPass dans Audience Manager, ce qui les rend plus faciles à repérer. DirectPass utilise un flux de travail spécialisé pour l'intégration et le déploiement des ensembles de données qui permet l'ingestion et le déploiement directs d'identifiants d'appareils bruts (comme les Device IDs et les adresses IP) sans devoir d'abord les résoudre dans le graphique des foyers de Cadent. Les ensembles de données DirectPass n'incluent pas les informations au niveau du foyer.

Les icônes DirectPass s'affichent aux emplacements suivants dans Audience Manager :

- Page Audiences
- Page Deployments
- New Audience, onglet My Datasets
- New Audience, onglet Marketplace Datasets
- New Audience, onglet Shared with Me



Audiences <i>Abi</i>	
Filter by Name, ID, Data Provider...	
Name	ID
DO_NOT_DELETE_Look...	18774
TV 3079 - test_tv_istio	18730
test_look_alike_model	22149
Test_Audience_Hetal_17726712...	22718
Test_Audience_Hetal_1234	22717
SH_directpass-audience	22716
RS_18_Birdies_Aud	22713

Icône DirectPass sur la page Audiences

Les balises d'audience s'affichent sur la page Audiences

Vous pouvez désormais voir les balises d'audience avec les audiences sur la page Audiences. Les balises permettent de créer des catégories d'audiences qui partagent des caractéristiques communes. Bien que les balises ne s'affichent pas par défaut, vous pouvez les ajouter à la page à l'aide de l'icône  sur la page Audiences.

L'indicateur % Reach sur la page New Audience a été mis à jour

Dans la section **Portée totale** de la page New Audience, la colonne **% Portée** a été renommée **Variation en pourcentage** afin d'améliorer la clarté. Cette colonne affiche le pourcentage de variation de la portée totale lorsque vous configurez une nouvelle audience. Elle est mise à jour en temps réel lorsque vous apportez des modifications à l'audience que vous créez. En savoir plus



✓ 18 mars 2026

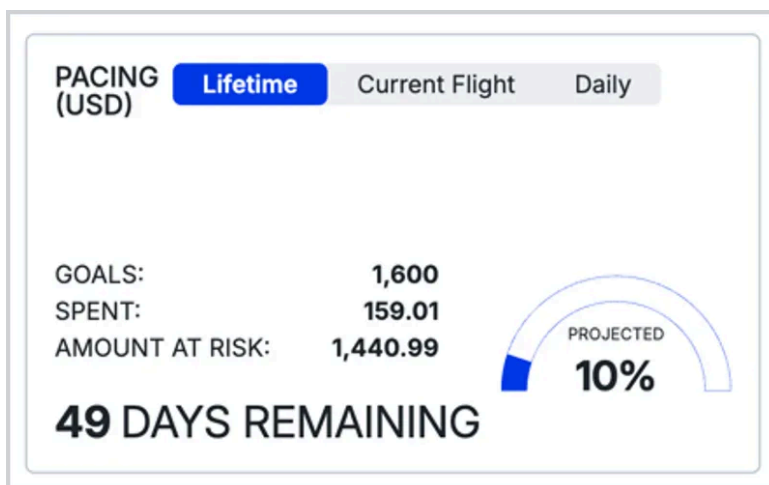
Notes de version du gestionnaire de publicités Manager

Nouvelle vue « Lifetime » pour le rythme des éléments de ligne

Nous avons ajouté une nouvelle période **À vie** à la tuile de rythme des éléments de ligne. Cela vous permet de voir les métriques de rythme sur l'ensemble des vols d'un élément de ligne, vous offrant une vue plus globale des performances et vous aidant à mieux comprendre et analyser les performances de vos campagnes à long terme.

Voici un aperçu des métriques que vous verrez dans cette nouvelle vue :

Métrique	Description
Objectif	Il s'agit de l'objectif global de l'élément de ligne, selon la façon dont votre élément de ligne suit le rythme : - Si le rythme est basé sur les impressions, l'objectif reflète les impressions du vol. - Si le rythme est basé sur les dépenses, l'objectif reflète le plafond de dépenses du vol.
Dépensé	Le montant total dépensé jusqu'à hier.
Montant à risque	Cela montre la portion de l'objectif qui devrait être manquée, vous donnant une vue rapide des performances du rythme. Le montant à risque est calculé à l'aide de cette formule : $(100\% - \% \text{ projeté}) * \text{Objectif} = \text{Montant à risque}$
Jours restants	Le nombre de jours restants dans le vol actuel, sans inclure aujourd'hui.



Méthode simplifiée pour désactiver les campagnes, les éléments de ligne et les groupes d'annonces

Nous avons simplifié le processus de désactivation des éléments dans votre compte, le rendant plus rapide et plus intuitif pour gérer vos campagnes.

Désactivation des campagnes et des éléments de ligne

Vous pouvez maintenant définir directement le statut d'une campagne entière ou d'un élément de ligne à **Désactivé** sans modifier sa configuration budgétaire sous-jacente. Auparavant, vous deviez accéder à chaque groupe d'annonces individuellement pour





désactiver l'allocation. Ce changement rend la mise en pause de vos campagnes beaucoup plus rapide et se fait en une seule étape.

Désactivation des groupes d'annonces

Vous pouvez définir un groupe d'annonces individuel à **Désactivé**, mais uniquement si l'allocation budgétaire active de l'élément de ligne parent totalise déjà 100%.

Pour vérifier l'allocation budgétaire d'un élément de ligne, suivez simplement ces étapes :

1. Accédez à l'écran **Détails des postes**.
2. Sélectionnez le **Groupe d'annonces** et cliquez sur **Attribuer**. Cela ouvre le panneau Edit Ad Group Allocation.



3. Passez en revue les détails de l'allocation budgétaire et, si nécessaire, apportez des modifications pour porter l'allocation totale à 100%.

EDIT AD GROUP ALLOCATION

Search

Q

Clear

1 Selected

Allocate Evenly

☐	ID	AD GROUP	CURRENT ALLOCATION (%)	NEW ALLOCATION (%)
☐	100000010	MW Test Campaign_Video_AdGroup 1	25	25
☐	100000635	MW Test Campaign_Video_AdGroup 1_Clone-1	25	25
☐	100000636	MW Test Campaign_Video_AdGroup 1_Clone-2	25	25
☐	100000637	MW Test Campaign_Video_AdGroup 1_Clone-3	25	25
TOTAL				100%

Gagnez du temps avec les téléchargements de pixels en lot

Nous sommes ravis d'annoncer une nouvelle fonctionnalité qui rendra votre flux de travail beaucoup plus efficace. Vous pouvez désormais créer des pixels en lot, y compris des balises universelles, des activités et des activités d'action, directement dans Ad Manager, ce qui vous évite le travail manuel de les créer un par un.

Comment téléverser des pixels en lot

Suivez ces étapes pour commencer :

1. Accédez à l'écran **Pixels** à partir du menu principal.
2. Cliquez sur le bouton **+Pixels** et sélectionnez l'option **En vrac**. Cela ouvrira le panneau Add Pixels in Bulk.
3. Sélectionnez l'**Annonceur** auquel vous souhaitez associer les pixels.
4. Téléchargez le **fichier d'exemple** fourni. Ce modèle garantit que vos données sont correctement formatées pour un téléversement réussi. Ajoutez les informations de vos pixels dans ce fichier et enregistrez-le sur votre ordinateur. Si vous essayez de créer de nouvelles activités pour une Universal Tag existante, vous devez télécharger un nouveau modèle à partir du panneau afin de vous assurer que vos Universal Tags existantes sont préremplies dans la liste déroulante de l'onglet Activités du modèle.
5. Glissez-déposez votre fichier complété dans la zone de téléversement, ou cliquez pour le parcourir sur votre ordinateur.
6. Cliquez sur **Ajouter**.

Une fois le téléversement terminé, vous serez automatiquement redirigé vers l'écran principal Pixels où vous verrez un message de réussite confirmant que vos nouveaux pixels ont été ajoutés.

Optimisez votre ciblage avec les actions en lot

Vous pouvez désormais mettre à jour le ciblage **Catégories personnalisées**, **Catégories d'approvisionnement** et **Réseau et chaîne** pour plusieurs groupes d'annonces en même temps.

Détails importants à retenir

- **Catégories personnalisées** : Vous pouvez sélectionner des catégories personnalisées prédéfinies lors d'une mise à jour en lot. Les nouvelles catégories doivent être créées au préalable.
- **Réseau et canal** : Cette option de ciblage est spécifiquement destinée au CTV.
- **Catégories d'approvisionnement** : Il s'agit de catégories d'inventaire définies par Jounce.



Comment cela fonctionne :

1. Accédez à l'écran Line Items Details et sélectionnez le ou les groupes d'annonces que vous souhaitez mettre à jour, puis cliquez sur **Mettre à jour le ciblage**. Cela ouvre le panneau Add Targeting.

Line Item Details: Test_500_targeting (1000015978)

View Bidding Details View Rate Card 2 Selected Allocate Update Status Clone Delete Update Frequency Cap **Update Targeting**

SUMMARY

OVERVIEW Lifetime Yesterday Today

IMPRESSIONS CPM (USD) WIN RATE (%)

KPI 1 - KPI (USD) VCR (%)

0 - -

PACING (USD) Lifetime Current Flight Daily

GOALS: SPENT: AMOUNT AT RISK: 0

DAYS REMAINING PROJECTED

INSIGHTS KPI 1 KPI 2 KPI 3 BIDDING

LIFETIME AVG. THROUGH YESTERDAY: -

AD GROUPS (2)

Search

DATE RANGE: Preset Custom Lifetime

ID	AD GROUPS	CHANNEL	ALLOCATION	YESTERDAY (USD)	DAILY GOAL EST. (USD)	Impres:
1000020619	Test_500_targeting_AdGroup.1	CTV	0%	-	0	
1000020621	Test_500_targeting_AdGroup.2	CTV	0%	-	0	
TOTAL	-	-	0%	-	-	

2. Dans le panneau **Ajouter un ciblage**, effectuez vos nouvelles sélections de ciblage et fermez le panneau.
3. Tout est prêt ! Les nouvelles options de ciblage seront reflétées dans l'écran **Détails du groupe d'annonces** pour chaque groupe d'annonces que vous avez mis à jour.



✓ 19 février 2026

Notes de version d'Ad Manager

Mise à jour des éléments de campagne : paramètre Days Ahead

Vous avez maintenant la possibilité d'indiquer si un élément de campagne doit viser à être livré intégralement un certain nombre de jours avant la date de fin.

The screenshot shows the 'FLIGHT PACING' section with 'Automatic' selected. Below it, the 'FLIGHTS' section shows a table with columns: FLIGHT, START & END DATE, * IMPRESSION GOAL, * IMPRESSION CAP, and SPEND CAP (OPTIONAL). The 'DAYS AHEAD' dropdown is highlighted with a blue box, showing '1'.

FLIGHT	START & END DATE	* IMPRESSION GOAL	* IMPRESSION CAP	SPEND CAP (OPTIONAL)
1	03/01/2026 - 03/31/2026			
TOTAL				

Mise à jour du ciblage : ciblage par catégories

Vous pouvez désormais cibler des catégories de contenu IAB ou Peer39, y compris la création et la sélection de segments Peer39 personnalisés et avancés, afin de vous assurer que vos annonces apparaissent sur du contenu pertinent.

Vous pouvez cibler des catégories IAB spécifiques ou des segments Peer39, ou exclure des catégories afin d'empêcher vos annonces d'apparaître sur du contenu indésirable.

The 'ADD CATEGORIES' dialog box is shown with the 'Market Rate CPM' tab selected. It displays a table of segments with columns: ID, NAME, TYPE, and RATE. The table lists 6 segments, all of which are Peer39 segments. The 'Selected (6)' status is indicated at the top right.

ID	NAME	TYPE	RATE (
296077542	Custom Segment	Peer39	0.3
680163062	Custom Segment	Peer39	0.3
989898	Custom Segment	Peer39	0.3
111222333	Custom Segment	Peer39	0.3
414141	Custom Segment	Peer39	0.3
434343	Advanced Custom Segment	Peer39	0.35
7654321	Custom Segment	Peer39	0.3
6754321	Advanced Custom Segment	Peer39	0.35

Mise à jour du ciblage : ciblage par genre

Le ciblage par genre vous permet d'aligner vos campagnes sur des types spécifiques de contenu vidéo CTV. Cette fonctionnalité vous permet de cibler ou d'exclure du contenu en fonction du genre — par exemple, Comédie, Sports, Drame, Documentaires et True Crime. Vous pouvez soit inclure des genres spécifiques à cibler, soit exclure les genres que vous souhaitez éviter.











ADD GENRE

Search

Q

GENRE

+

+

Action/ Adventure

+

+

Adult

+

+

Animation

+

+

Comedy

+

+

Crime

+

+

Documentary

+

+

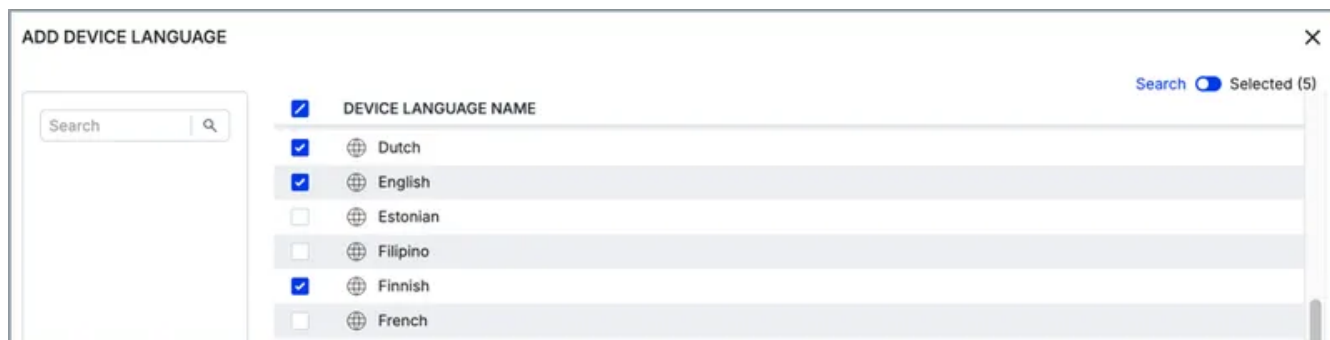
Drama

Search

Selected (6)

Mise à jour du ciblage : ciblage par langue de l'appareil

Le ciblage par langue de l'appareil vous permet de diffuser des annonces spécifiquement aux utilisateurs dont les appareils sont configurés dans une langue particulière. C'est un moyen simple de vous assurer que votre message est pertinent et compris par votre public cible, particulièrement dans les régions multilingues ou les campagnes mondiales.



ADD DEVICE LANGUAGE

Search Selected (5)

☒	DEVICE LANGUAGE NAME
☒	Dutch
☒	English
☐	Estonian
☐	Filipino
☒	Finnish
☐	French

Mise à jour du ciblage : ciblage par langue du contenu

Cette méthode vous permet de cibler les spectateurs en fonction de la langue principale du contenu CTV lui-même. Par exemple, vous pouvez choisir de faire apparaître vos annonces uniquement pendant des programmes en espagnol, peu importe les paramètres de l'appareil du spectateur.

Vous pouvez choisir soit de cibler une langue, soit de l'exclure.



ADD CONTENT LANGUAGE +\$2.45 USD

Search Selected (6)

	CONTENT LANGUAGE
	Chinese
	Croatian
	Czech
	Danish
	Dutch
	English
	Estonian
	Filipino

Mise à jour du ciblage : ciblage Geo POI

Vous pouvez cibler des POI via le ciblage géographique, lesquels sont présentés à l'utilisateur sous forme d'agrégat plutôt que comme une sélection de points uniques. Par exemple, en un clic, vous pouvez cibler 4 000 Starbucks plutôt que d'avoir à les sélectionner un par un.

EDIT GEOGRAPHY

REGIONS
States, Cities, DMAs, Zips

LOCATIONS
POIs and coordinates

SAVED LISTS
Pre-made Geo/POI lists

SEARCH

Import

Download Sample File

Type at least 3 characters

SELECTIONS

INDIVIDUALS TARGETED MUST BE LOCATED WITHIN:

LOCATIONS

1 Selected

POI Aggregates

Starbucks (17356 Locations)

BUT CANNOT BE LOCATED WITHIN:

No Negative Targeting has been set.

View
POI

CANCEL

SAVE

Mise à jour du ciblage : reciblage de plateforme

Le reciblage de plateforme vous permet de cibler les utilisateurs qui ont déjà été exposés à vos campagnes.

ADD PLATFORM RETARGETING

Campaigns

Search

Filter By:

(All) Status

(All) Campaigns

(All) Line Items

ID

CAMPAIGN / LINE ITEM / AD GROUP

> 1000000109	Collin_Test_Campaign
> 1000000491	Test_Campaign_Audience_001
> 1000000498	DNT_SP_PendingExpiredCampaign
> 1000000499	DNT_SP_CompletedExpired
> 1000000500	Auto_Adgroup_Regression_Campaign
> 1000000887	Bis Campaign4 under Advertiser 1-Vision
> 1000001857	Vh Test ExternalIds campaign
> 1000002809	Tien_Test_Campaign_08_19_25
> 1000002931	SP_Campaign_Fluidity_Aug22
> 1000003229	Campaign_Test_123
> 1000003234	Fluidity_Campaign

* User Frequency (Times)

2 to 5

* Target

IP Address

- Vous pouvez choisir soit un ciblage positif (inclure), soit un ciblage négatif (supprimer) pour l'audience sélectionnée.
- Fréquence utilisateur : Cela vous permet de définir une plage minimale et maximale du nombre de fois qu'une personne a vu votre publicité. Par exemple, si vous définissez un minimum de 2 et un maximum de 5, le système ciblera uniquement les utilisateurs qui ont vu la publicité au moins 2 fois, mais pas plus de 5 fois.



- Cible : Il s'agit de l'identifiant utilisé pour compter la fréquence. Vous pouvez choisir de suivre les utilisateurs par leur adresse IP, un Cookie de navigateur ou Tous (qui utilise les deux méthodes).

Mise à jour du ciblage : Actions en lot

Vous pouvez maintenant sélectionner plusieurs groupes de publicités dans la vue des détails des éléments de ligne afin d'appliquer des actions de ciblage en lot (voir la capture d'écran ci-dessous). Les sélections que vous effectuez remplaceront les sélections précédentes.

Les options de ciblage que vous pouvez mettre à jour en lot sont :

- Liste des applications/sites
- Packages
- Géographie
- Offres
- Mots-clés

Test Camp For Display Status 2.23 >>>
Line Item Details: NewCampaignFeb26_Display_Clone-1 (1000015312)

View Bidding Details View Rate Card 2 Selected Allocate Update Status Clone Delete Update Frequency Cap **Update Targeting** Update Optimization

SUMMARY

OVERVIEW Lifetime Yesterday Today

IMPRESSIONS CPM (USD) WIN RATE (%)

KPI 1 - KPI (USD) CTR (%)

0 -

PACING (USD) Current Flight Daily

GOALS: 1
SPENT: 1.00
AMOUNT AT RISK: 1.00

43 DAYS REMAINING

INSIGHTS KPI

AD GROUPS (2)

Search

DATE RANGE: Preset Custom Lifetime

☒	ID	AD GROUPS	CHANNEL	ALLOCATION	YESTERDAY (USD)	DAILY GOAL
☒	1000019999	NewCampaignFeb26_Display_AdGroup...	Display	0%	—	—
☒	1000020000	NewCampaignFeb26_Display_AdGroup...	Display	0%	—	—
TOTAL		—	—	0%	—	—

Notes de version des rapports

Moins de filtres sont requis lors de la création d'un rapport ad hoc

Nous avons réduit le nombre de filtres que vous devez appliquer lors de la création d'un rapport Ad Hoc. Désormais, vous devez uniquement sélectionner une agence. Cela permet de créer des rapports plus larges que ceux qui étaient possibles auparavant. En savoir plus sur [Création de rapports ad hoc](#).

BASICS

* Name

FILTERS Filter

* Filter Type

Select Type

▼

Nouvelle dimension de rapport ajoutée aux modèles Performance Insights

Afin de vous permettre de voir plus de détails sur vos créations dans les rapports, une nouvelle dimension de rapport, **Type créatif**, a été ajoutée. Un type de création définit la façon dont un utilisateur interagit avec une publicité numérique. Par exemple, **vidéo**, **affichage**, et **bannière** sont des types de créations qui s'appliquent aux publicités en ligne. La dimension du type de création est rendue disponible dans les modèles de rapports suivants :

- Mesures de campagne
- Mesures CTV
- Mesures vidéo
- Mesures de ciblage géographique
- Portée et fréquence
- Mesures HCP

En savoir plus sur [Modèles Performance Insights](#).

DIMENSIONS (0 selected)

Search

Search Selected

☐ Select All 29 Dimensions

☐ Agency / Partner

☐ Agency ID / Partner ID

☐ Advertiser

☐ Advertiser ID

☐ Campaign

METRICS (0 selected)

Search

Search Selected

☐ Select All 20 Metrics

☐ Clicks

☐ Completes

☐ Impressions

☐ Total Spend

☐ Agency Margin

Bidding Snapshot renommé Bidding Details

Lorsque vous accédez au rapport Bidding Snapshot, vous remarquerez qu'il a été renommé **Détails de l'offre**. Le nouveau nom s'affiche dans la barre de navigation de gauche et en haut de la page dans Ad Manager. Il s'agit du même rapport qu'auparavant, mais avec deux nouveaux ajouts décrits ci-dessous.

Ad Manager

- Campaign Management
 - Campaigns
 - Creatives**
- Configuration
 - Deals
 - App/Site Lists
 - Pixels
 - Geo Lists
 - Keywords
- Reporting
 - Ad Hoc Reporting
 - Bidding Details**

Bidding Details

BIDDING SNAPSHOTS

* Advertiser
AutoZone

* Campaign
RTB_Bidding_Campaign1

^ HOURLY VIEW

v DAILY VIEW

Refresh

HOUR (ET)	ELIGIBLE REQUESTS	REQUESTS REJECTED	BIDS	BID RATE (%)	W
8 AM	0	0	0	—	—
9 AM	4	0	4	100	0

Le rapport Bidding Details ajoute une vue horaire

En plus des vues Daily et Troubleshooting, vous avez maintenant l'option de sélectionner une **Vue horaire** des éléments de ligne et des détails des groupes de publicités. La vue horaire affiche l'activité de l'heure en cours et peut être actualisée pour afficher les plus récentes mises à jour.

Bidding Details

BIDDING SNAPSHOTS

* Advertiser
Select Advertiser

* Campaign
Select Campaign

* Line Item
Select Line Item

Ad Group
Select Ad Group

VIEW

v HOURLY VIEW

MINUTE	ELIGIBLE REQUESTS	REQUESTS REJECTED	BIDS	BID RATE (%)	WIN RATE (%)	IMP GOAL	IMPS SERVED	CLICKS/COMPLETES	CTR/VCR (%)	EC
No Results Found.										

De nouveaux champs s'affichent dans le rapport Bidding Details

Les nouveaux champs suivants ont été ajoutés aux résultats du rapport Bidding Details pour les résultats de rapports horaires et quotidiens :

- **Objectif de dépenses** - La cible monétaire qu'un annonceur ou acheteur média vise à atteindre dans la période définie d'un élément de ligne ou d'un groupe d'annonces.
- **Dépenses totales** - Le montant d'argent que votre campagne ou élément de ligne a dépensé à partir du budget pour obtenir des impressions.



✓ 4 février 2026

Notes de version d'Audience Manager

Ingestion de données DirectPass

DirectPass utilise un type spécialisé d'intégration de jeux de données et un flux de travail de déploiement qui permet l'ingestion directe et le déploiement d'identifiants d'appareils bruts, comme les IDs d'appareil et les adresses IP, sans d'abord les résoudre dans le graphe de foyers de Cadent. Les jeux de données DirectPass n'incluent pas les informations au niveau des foyers.

DirectPass permet des flux de travail spécialisés nécessitant l'utilisation d'identifiants bruts et non modifiés. Vous pouvez utiliser DirectPass dans les conditions suivantes :

- Ciblage pour des régions à l'extérieur des États-Unis, comme des segments canadiens à utiliser dans AdManager.
- Transmission directe d'identifiants bruts à des partenaires de destination qui possèdent leurs propres systèmes de résolution d'identité.

Remarque : Vous devez avoir DirectPass activé pour votre organisation afin d'effectuer ce type d'intégration de données. Si cela vous intéresse, contactez votre gestionnaire de compte.

Filtre d'ID ajouté à la page Deployments

Le filtre de la page Deployments comprend maintenant l'option de rechercher par ID de déploiement. [Détails de la page Deployments](#)

Deployments		
Deployments 2541		
Q Filter by Name, ID, Destination		
Name		ID
29_10		53447



✓ 23 janvier 2026

Note de version de Health Audience

Attribuer une audience HCP à un groupe d'annonces

Qu'est-ce qu'une audience HCP?

Une audience de professionnels de la santé (HCP) est un groupe cible composé de professionnels de la santé tels que des médecins, des infirmières et infirmiers, des pharmaciens et des dentistes. Lorsque vous créez et enregistrez une audience HCP dans le **Générateur de Health Audience**, elle devient automatiquement disponible pour attribution dans l'interface utilisateur, tant que l'audience et le groupe d'annonces sont tous deux associés au même annonceur.

Comment attribuer une audience HCP à un groupe d'annonces

Vous pouvez maintenant attribuer une audience HCP à votre groupe d'annonces pendant la configuration du groupe d'annonces. Consultez [Ajouter un groupe d'annonces](#) pour obtenir les instructions.

ADD AUDIENCES (1)		CUSTOM AUDIENCES (0)		HCP AUDIENCES (1)	TOTAL CPM: 24.00 USD	
Search		ID	AUDIENCE		HCPS	MATCHED %
		☒	20687	NoDeployHCPTest	260,805	88%

Écran d'attribution d'audience dans Ad Manager



3.1.1.1. 5.4.1 Élément DocumentSearchRequest requis

Processus d'intégration étape par étape

Ces étapes décrivent le processus d'intégration d'un client à Surescripts Data Solutions, de la configuration du compte jusqu'aux téléchargements de données.

ăndarăt

Remarque: Il s'agit d'une documentation Early Adopter. Les exigences et le contenu peuvent changer.



Étape 1 : Vérifier l'accès à Data Package Builder dans Workbench

- Pour obtenir des instructions sur la configuration de l'accès, consultez les instructions [Configuration de l'accès à Workbench et Data Package Builder](#).

Étape 2 : Générer des clés

Étape 2a : Générer des paires de clés RSA pour l'authentification de Data Package Builder

Pour vous authentifier auprès de l'API Data Package Builder, vous devez générer une paire de clés RSA de 2048 bits et téléverser la clé publique via l'interface Client Keys.

Option 1 : Utilisation de Open SSL (recommandé - fonctionne sous Linux, macOS et Windows)

1. Générez une clé privée RSA de 2048 bits.
- 2.



 Document image

PowerShell

```
openssl genrsa -out private_key.pem 2048
```

2. Extrayez la clé publique.

PowerShell

```
openssl req -new -x509 -key privatekey.pem -out publickey509.pem -subj '/CN=data-package-builder'
```

3. Vérifiez vos paires de clés.

PowerShell

```
# View the private key
cat privatekey.pem
# View the public key
cat publickey509.pem
```

Option 2 : Utilisation de PowerShell (Windows)

1. Générez une paire de clés RSA.

PowerShell

```
$rsa = [System.Security.Cryptography.RSA]::Create(2048)
```

2. Exportez la clé publique au format X.509 SubjectPublicKeyInfo.

PowerShell

```
$publicKey = $rsa.ExportSubjectPublicKeyInfo()
$publicPem = "-----BEGIN PUBLIC KEY-----`n" +
    [Convert]::ToBase64String($publicKey, 'InsertLineBreaks') +
    "`n-----END PUBLIC KEY-----"
$publicPem | Out-File -FilePath "public_key.pem" -Encoding ASCII
```



3. Exportez la clé privée.

PowerShell

```
$privateKey = $rsa.ExportRSAPrivateKey()
$privatePem = "-----BEGIN RSA PRIVATE KEY-----`n" +
    [Convert]::ToBase64String($privateKey, 'InsertLineBreaks') +
    "`n-----END RSA PRIVATE KEY-----"
$privatePem | Out-File -FilePath "private_key.pem" -Encoding ASCII

Write-Host "Keys generated successfully!"
Write-Host "Public key is in X.509 SubjectPublicKeyInfo format"
```



Ce que vous obtiendrez :

Latest Version			
Label	Sync Translation ⓘ		
Archbee English	☐		
Archbee German	☒		

Après la génération de la paire de clés, vous disposerez de deux fichiers :

- 1. private_key.pem - Votre clé privée
 - **CRITIQUE:** Conservez-la en sécurité
 - Stockez-la dans un emplacement sécurisé (HSM, coffre-fort de clés, stockage chiffré)
 - Elle sera utilisée pour signer un JWT compact afin de demander un jeton d'accès au serveur d'autorisation
- 2. public_key.pem - Votre clé publique
 - C'est ce que vous téléversez dans Data Package Builder
 - Peut être partagée en toute sécurité (elle est dite « publique » pour une raison)
 - Exemple de format :

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0Z3VS5JJcds3...
-----END PUBLIC KEY-----
```

Étape 2b : Téléverser votre clé publique



1. Connectez-vous à **Générateur de paquet de données**.
2. Accédez à **Clés du client**.
3. Collez l'intégralité du contenu de **public_key.pem** (y compris les lignes BEGIN/END).
4. Cliquez sur **Créer un client**.

◦ **Remarque** : Cela créera le Client_ID que le client utilisera plus tard dans la section [Construire l'assertion JWT compacte](#).

⚠ Important : Une fois que vous avez créé votre **ID client**, informez votre point de contact Surescripts. Surescripts doit configurer votre package de données avant que vous puissiez passer aux étapes suivantes.

Vous ne pourrez pas continuer tant que cette configuration ne sera pas terminée.



**Bonnes pratiques de sécurité :****✓ À FAIRE :**

- Générez les clés dans un environnement sécurisé
- Stockez les clés privées dans un coffre de clés ou un HSM pour une utilisation en production
- Utilisez des permissions de fichier strictes pour les fichiers de clés privées (chmod 600 private_key.pem sur Unix)
- Effectuez une rotation des clés périodiquement

✗ À NE PAS FAIRE :

- Partager ou envoyer des clés privées vers un système de contrôle de version
- Envoyer des clés privées par courriel
- Stocker des clés privées en texte brut dans le code de l'application
- Réutiliser la même paire de clés pour plusieurs clients/applications

Étape 2c : Erreurs courantes

Le tableau ci-dessous comprend des informations sur les erreurs courantes pouvant être rencontrées lors de la création de paires de clés.

Erreur	Informations de dépannage
Erreur « Invalid certificate/public key format »	• Assurez-vous de téléverser la clé PUBLIC et non la clé privée • Vérifiez que les en-têtes PEM sont inclus • Vérifiez qu'il n'y a pas d'espaces ou de sauts de ligne supplémentaires
La clé semble être téléversée, mais l'authentification échoue	• Vérifiez que la clé privée correspondante est utilisée pour signer les requêtes • Assurez-vous que la clé est de taille 2048 ou 4096 (1024 n'est pas autorisé et les tailles de clé plus grandes ne tiendront pas dans notre limite d'entrée JWT de 8 KB)

Étape 3 : Query Builder

Le **Générateur de requêtes** dans Workbench vous aide à créer une **URL de requête OData** qui définit exactement les données de médicaments prescrits que vous souhaitez récupérer depuis l'API Data Solutions. Vous pouvez créer une requête visuellement ou la modifier directement, selon votre préférence.



Remarque : Le Query Builder est facultatif; les utilisateurs expérimentés peuvent soumettre des **requêtes OData manuelles** directement.

OData n'est pas le seul type de requête pris en charge. Pour plus d'informations sur les autres utilisations du Query Builder, consultez **[Vue d'ensemble du Query Builder](#)**.

Pour plus d'informations sur OData, consultez **<https://learn.microsoft.com/en-us/odata/overview>**



Important : Avant de commencer :

- Vous devez pouvoir **vous connecter à Workbench**.
- Votre utilisateur doit avoir accès à **Générateur de paquet de données** avec le rôle **Modifier la requête**.
- Vous devez utiliser le produit **Médicament prescrit** dans Data Solutions.
- Pour obtenir des instructions à ce sujet, consultez **[Configuration de l'accès à Workbench et Data Package Builder](#)**.



Étape 3a : Exécuter Query Builder (OData)

1. Connectez-vous à **Établi** et ouvrez **Générateur de paquet de données**.
2. Sélectionnez **Générateur de requêtes**.
3. Choisissez le produit **Médicament prescrit**.
4. Sélectionnez les **colonnes** que vous souhaitez retourner (elles correspondent à \$select dans OData).
5. Ajoutez des **filtres** pour limiter les résultats.
 - Un filtre **PARTITION_DAY** est requis.
6. Sélectionnez **Générer à partir des sélections** pour créer automatiquement la requête OData
(ou modifiez manuellement la requête si vous connaissez bien OData).
7. Copiez l'**URL de requête OData** générée.
8. Utilisez l'URL dans votre **requête GET de l'API Data Solutions** pour récupérer les données de Prescribed Medication.

Remarque : Pour obtenir des informations sur l'exécution de Query Builder pour d'autres cas d'utilisation, consultez [Étapes clés](#)
(s'applique à tous les cas d'utilisation)

Étape 4 : Déterminer le filtre PARTITION_DAY

Pour déterminer quel filtre PARTITION_DAY demander, consultez la page [Comment utiliser les endpoints](#).

Étape 5 : Envoyer des requêtes REST à l'API Data Solutions

Authentification

Apprenez à vous authentifier auprès de l'API Data Solutions à l'aide du flux OAuth 2.0 Client Credentials Grant avec JWT Bearer Assertion.

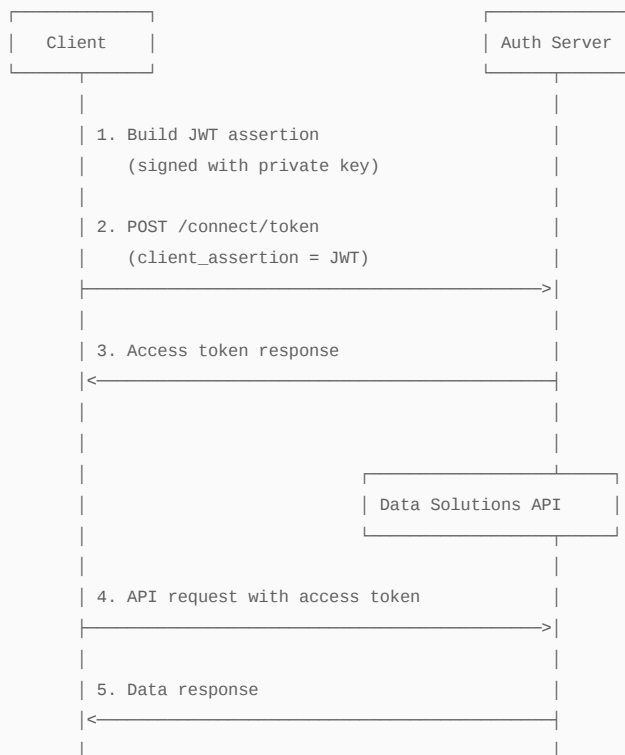
Ce que vous allez faire

L'API Data Solutions utilise un processus d'authentification en deux étapes :

1. **Créer une assertion JWT** - Créez et signez une revendication JWT à l'aide de votre ClientID et de votre clé privée
2. **Demander un jeton d'accès** - Échangez l'assertion JWT contre un jeton d'accès OAuth au endpoint `/connect/token` du serveur d'authentification
3. **Utiliser le jeton d'accès** - Incluez le jeton d'accès dans les requêtes API



Data Solutions Authentication Process



Avant de commencer

Assurez-vous d'avoir :

1. **ID client** - Un identifiant unique pour votre application.
2. **Clé privée** - Une clé privée RSA au format **PKCS#8** PEM.
3. **Enregistrement de la clé publique** - Votre clé publique doit être enregistrée auprès du serveur d'authentification pour votre ID client.

Format de la clé privée

Votre clé privée doit être au **format PKCS#8 PEM** :

JS JS

```
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAgEAAoIBAQC...
... base64 encoded key data ...
-----END PRIVATE KEY-----
```

Étape 5a : Créer l'assertion JWT compacte

Créez un JWT avec la structure suivante, signé avec votre clé privée à l'aide de l'algorithme RS384 (RSASSA-PKCS1-v1_5 avec SHA-384). Ce JWT prouve votre identité au serveur d'authentification.



En-tête JWT

JSON

```
{
  "alg": "RS384",
  "typ": "JWT"
}
```

Payload JWT

JSON

```
{
  "iss": "your-client-id",           // Issuer: your client ID
  "sub": "your-client-id",          // Subject: your client ID
  "aud": "ENVIRONMENT_AUTH_URL/connect/token", // Audience: token endpoint URL
  "jti": "550e8400-e29b-41d4-a716-446655440000", // JWT ID: unique identifier (UUID)
  "iat": 1706572800,                // Issued at: current Unix timestamp
  "nbf": 1706572800,                // Not before: current Unix timestamp
  "exp": 1706573100                 // Expires: current timestamp + 5 minutes
}
```

Remarques importantes sur les champs :

- `iss` et `sub` doivent tous deux être votre ID client
- `aud` doit être l'URL exacte du point de terminaison du jeton (y compris `/connect/token`)
- `jti` doit être unique pour chaque requête (utilisez un UUID)
- `exp` doit être à 5 minutes de `iat` (maximum recommandé)

Exemple : Création d'un JWT en Python

 Python

```
import jwt
import uuid
import time
from datetime import datetime, timedelta

# Your credentials
client_id = "your-client-id"
private_key = """-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAgEAAoIBAQC...
-----END PRIVATE KEY-----"""
token_endpoint = "ENVIRONMENT_AUTH_URL/connect/token"

# Build the JWT payload
now = int(time.time())
payload = {
    "iss": client_id,
    "sub": client_id,
    "aud": token_endpoint,
    "jti": str(uuid.uuid4()),
    "iat": now,
    "nbf": now,
    "exp": now + 300 # 5 minutes from now
}

# Sign the JWT with RS384
client_assertion = jwt.encode(
    payload,
    private_key,
    algorithm="RS384"
)

print(f"JWT Assertion: {client_assertion}")
```

Exemple : Création d'un JWT en C#



C#

```
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Security.Cryptography;
using Microsoft.IdentityModel.Tokens;

// Your credentials
var clientId = "your-client-id";
var privateKeyPem = @"-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAgEAAoIBAQC...
-----END PRIVATE KEY-----";
var tokenEndpoint = "ENVIRONMENT_AUTH_URL/connect/token";

// Load the private key
using var rsa = RSA.Create();
rsa.ImportFromPem(privateKeyPem);
var signingCredentials = new SigningCredentials(
    new RsaSecurityKey(rsa),
    SecurityAlgorithms.RsaSha384
);

// Build the JWT
var now = DateTime.UtcNow;
var handler = new JwtSecurityTokenHandler();
var token = handler.CreateJwtSecurityToken(
    issuer: clientId,
    audience: tokenEndpoint,
    subject: new ClaimsIdentity(new[] { new Claim("sub", clientId) }),
    notBefore: now,
    expires: now.AddMinutes(5),
    issuedAt: now,
    signingCredentials: signingCredentials
);

// Add the jti claim
token.Payload["jti"] = Guid.NewGuid().ToString();

var clientAssertion = handler.WriteToken(token);
Console.WriteLine($"JWT Assertion: {clientAssertion}");
```

Exemple : Création d'un JWT en JavaScript/Node.js



JS JS

```
const jwt = require('jsonwebtoken');
const { v4: uuidv4 } = require('uuid');
const fs = require('fs');

// Your credentials
const clientId = 'your-client-id';
const privateKey = fs.readFileSync('private_key.pem', 'utf8');
const tokenEndpoint = 'ENVIRONMENT_AUTH_URL/connect/token';

// Build the JWT payload
const now = Math.floor(Date.now() / 1000);
const payload = {
  iss: clientId,
  sub: clientId,
  aud: tokenEndpoint,
  jti: uuidv4(),
  iat: now,
  nbf: now,
  exp: now + 300 // 5 minutes
};

// Sign the JWT with RS384
const clientAssertion = jwt.sign(payload, privateKey, {
  algorithm: 'RS384'
});

console.log(`JWT Assertion: ${clientAssertion}`);
```

Étape 5b : Demander un jeton d'accès

Envoyez une requête POST au point de terminaison `/connect/token` du serveur d'authentification avec l'assertion JWT.

Format de la requête

Text

```
POST /connect/token HTTP/1.1
Host: ENVIRONMENT_AUTH_URL
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
&client_id=your-client-id
&client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer
&client_assertion=<your-signed-jwt>
&scope=DsaApi
```

Paramètres :

- `grant_type` - Doit être `identifiants_client`
- `identifiant_client` - Votre identifiant client
- `client_assertion_type` - Doit être `urn:ietf:params:oauth:client-assertion-type:jwt-bearer`



- `client_assertion` - Le JWT signé de l'étape 1
- `portée` - Doit être `DsaApi` pour accéder à l'API Data Solutions

Exemple : Demander un jeton en Python

Python

```
import requests

# Build the token request
token_url = "ENVIRONMENT_AUTH_URL/connect/token"
data = {
    "grant_type": "client_credentials",
    "client_id": client_id,
    "client_assertion_type": "urn:ietf:params:oauth:client-assertion-type:jwt-bearer",
    "client_assertion": client_assertion,
    "scope": "DsaApi"
}

# Request the access token
response = requests.post(token_url, data=data)
response.raise_for_status()

token_response = response.json()
access_token = token_response["access_token"]
expires_in = token_response["expires_in"]

print(f"Access Token: {access_token}")
print(f"Expires in: {expires_in} seconds")
```

Exemple : Demander un jeton en C#



C#

```
using System.Net.Http;

var httpClient = new HttpClient();
var tokenUrl = "ENVIRONMENT_AUTH_URL/connect/token";

var requestData = new Dictionary<string, string>
{
    { "grant_type", "client_credentials" },
    { "client_id", clientId },
    { "client_assertion_type", "urn:ietf:params:oauth:client-assertion-type:jwt-bearer" },
    { "client_assertion", clientAssertion },
    { "scope", "DsaApi" }
};

var response = await httpClient.PostAsync(tokenUrl, new FormUrlEncodedContent(requestData));
response.EnsureSuccessStatusCode();

var tokenResponse = await response.Content.ReadFromJsonAsync<TokenResponse>();
var accessToken = tokenResponse.AccessToken;
var expiresIn = tokenResponse.ExpiresIn;

Console.WriteLine($"Access Token: {accessToken}");
Console.WriteLine($"Expires in: {expiresIn} seconds");

// Token response model
public class TokenResponse
{
    [JsonPropertyName("access_token")]
    public string AccessToken { get; set; }

    [JsonPropertyName("expires_in")]
    public int ExpiresIn { get; set; }

    [JsonPropertyName("token_type")]
    public string TokenType { get; set; }
}
```

Exemple : Demander un jeton en JavaScript/Node.js

JS JS

```
const axios = require('axios');
const qs = require('querystring');

const tokenUrl = 'ENVIRONMENT_AUTH_URL/connect/token';

const requestData = {
  grant_type: 'client_credentials',
  client_id: clientId,
  client_assertion_type: 'urn:ietf:params:oauth:client-assertion-type:jwt-bearer',
  client_assertion: clientAssertion,
  scope: 'DsaApi'
};

axios.post(tokenUrl, qs.stringify(requestData), {
  headers: { 'Content-Type': 'application/x-www-form-urlencoded' }
})
.then(response => {
  const accessToken = response.data.access_token;
  const expiresIn = response.data.expires_in;

  console.log(`Access Token: ${accessToken}`);
  console.log(`Expires in: ${expiresIn} seconds`);
})
.catch(error => {
  console.error('Token request failed:', error.response?.data || error.message);
});
```

Réponse réussie

JSON

```
{  
  "access_token": "eyJhbGciOiJSUzI6ImtpZCIjEYmZqNjc0OTAiLCJ0eXAiOiJKV1QiQfQ...",  
  "expires_in": 3600,  
  "token_type": "Bearer",  
  "scope": "DsaApi"  
}
```

Réponse d'erreur

JSON

```
{
  "error": "invalid_client",
  "error_description": "Invalid client or client credentials"
}
```

Codes d'erreur courants :

- `client_invalide` - ID client introuvable ou incompatibilité de clé publique



- `invalid_grant` - Échec de la validation de l'assertion JWT
- `champ_d'application_invalide` - La portée demandée n'est pas disponible

Étape 5c : Utiliser le jeton d'accès dans les requêtes de l'API Data Solutions

Incluez le jeton d'accès dans l'en-tête `Autorisation` de vos requêtes API.

Format de la requête

Text

```
GET /api/v0/odata/PrescribedMedicationData?$filter=PARTITION_DAY eq '2025-01-30' HTTP/1.1
Host: ENVIRONMENT_DATA_SOLUTIONS_URL
Authorization: Bearer <your-access-token>
```

Exemple : Requête API en Python

Python

```
import requests

api_url = "ENVIRONMENT_DATA_SOLUTIONS_URL/api/v0/odata/PrescribedMedicationData"
headers = {
    "Authorization": f"Bearer {access_token}"
}
params = {
    "$filter": "PARTITION_DAY eq '2025-01-30'",
    "$top": 10
}

response = requests.get(api_url, headers=headers, params=params)
response.raise_for_status()

data = response.json()
print(f"Retrieved {len(data['value'])} records")
```

Exemple : Requête API en C#



C#

```
var apiUrl = "ENVIRONMENT_DATA_SOLUTIONS_URL/api/v0/odata/PrescribedMedicationData";
var httpClient = new HttpClient();
httpClient.DefaultRequestHeaders.Authorization =
    new AuthenticationHeaderValue("Bearer", accessToken);

var queryParams = new Dictionary<string, string>
{
    { "$filter", "PARTITION_DAY eq '2025-01-30'" },
    { "$top", "10" }
};

var queryString = string.Join("&", queryParams.Select(kvp =>
    $"{Uri.EscapeDataString(kvp.Key)}={Uri.EscapeDataString(kvp.Value)}"));
var requestUrl = $"{apiUrl}?{queryString}";

var response = await httpClient.GetAsync(requestUrl);
response.EnsureSuccessStatusCode();

var data = await response.Content.ReadFromJsonAsync<ODataResponse>();
Console.WriteLine($"Retrieved {data.Value.Count} records");
```

Exemple : requête API en JavaScript/Node.js

JS JS

```
const axios = require('axios');

const apiUrl = 'ENVIRONMENT_DATA_SOLUTIONS_URL/api/v0/odata/PrescribedMedicationData';
const headers = {
    'Authorization': `Bearer ${accessToken}`
};
const params = {
    '$filter': "PARTITION_DAY eq '2025-01-30'",
    '$top': 10
};

axios.get(apiUrl, { headers, params })
    .then(response => {
        console.log(`Retrieved ${response.data.value.length} records`);
    })
    .catch(error => {
        console.error('API request failed:', error.response?.data || error.message);
    });
```

Étape 5d : gérer l'expiration du jeton

Les jetons d'accès expirent après la durée spécifiée dans le champ `expire_dans` (généralement 3600 secondes / 1 heure).

Bonnes pratiques :

1. Stockez le jeton et son heure d'expiration
2. Vérifiez si le jeton est expiré avant chaque requête



3. Demandez un nouveau jeton lorsque le jeton actuel expire ou est sur le point d'expirer
4. Ajoutez une marge de sécurité (par exemple, actualisez 5 minutes avant l'expiration réelle)

Exemple : gestion des jetons en Python

 Python

```
from datetime import datetime, timedelta

class TokenManager:
    def __init__(self, client_id, private_key, token_endpoint):
        self.client_id = client_id
        self.private_key = private_key
        self.token_endpoint = token_endpoint
        self.access_token = None
        self.expires_at = None

    def get_access_token(self):
        # Return cached token if still valid
        if self.access_token and datetime.utcnow() < self.expires_at:
            return self.access_token

        # Request new token
        client_assertion = self._build_jwt()
        response = requests.post(self.token_endpoint, data={
            "grant_type": "client_credentials",
            "client_id": self.client_id,
            "client_assertion_type": "urn:ietf:params:oauth:client-assertion-type:jwt-bearer",
            "client_assertion": client_assertion,
            "scope": "DsaApi"
        })
        response.raise_for_status()

        token_response = response.json()
        self.access_token = token_response["access_token"]

        # Set expiration with 5-minute buffer
        expires_in = token_response["expires_in"]
        self.expires_at = datetime.utcnow() + timedelta(seconds=expires_in - 300)

        return self.access_token

    def _build_jwt(self):
        # JWT building logic from Step 1
        now = int(time.time())
        payload = {
            "iss": self.client_id,
            "sub": self.client_id,
            "aud": self.token_endpoint,
            "jti": str(uuid.uuid4()),
            "iat": now,
            "nbf": now,
            "exp": now + 300
        }
        return jwt.encode(payload, self.private_key, algorithm="RS384")

# Usage
token_manager = TokenManager(client_id, private_key, token_endpoint)
access_token = token_manager.get_access_token()
```

Optionnel : utiliser DPOP pour une sécurité renforcée



L'API prend en charge DPoP (Demonstration of Proof-of-Possession) pour une sécurité renforcée. DPoP lie les jetons d'accès à une paire de clés spécifique, empêchant le vol de jetons.

Requête de jeton DPoP

Incluez une preuve JWT DPoP dans l'en-tête `DPoP` :

Text

```
POST /connect/token HTTP/1.1
Host: auth-server.surescripts.net
Content-Type: application/x-www-form-urlencoded
DPoP: <dpop-proof-jwt>

grant_type=client_credentials
&client_id=your-client-id
&client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer
&client_assertion=<your-signed-jwt>
&scope=DsaApi
&dpop_jkt=<jwk-thumbprint>
```

Structure de la preuve DPoP

En-tête :

JSON

```
{
  "typ": "dpop+jwt",
  "alg": "RS256",
  "jwk": {
    "kty": "RSA",
    "n": "<base64url-modulus>",
    "e": "<base64url-exponent>"
  }
}
```

Charge utile :

JSON

```
{
  "htu": "ENVIRONMENT_AUTH_URL/connect/token",
  "htm": "POST",
  "iat": 1706572800,
  "jti": "unique-id-here"
}
```

Requête API DPoP

Lors de l'utilisation d'un jeton lié à DPoP, incluez la preuve DPoP dans les requêtes API :



Text

```
GET /api/v0/odata/PrescribedMedicationData HTTP/1.1
Host: ENVIRONMENT_DATA_SOLUTIONS_URL
Authorization: DPoP <access-token>
DPoP: <dpop-proof-jwt>
```

La preuve DPoP pour les requêtes de ressources doit inclure la revendication `ath` (hachage du jeton d'accès) :

JSON

```
{
  "htu": "ENVIRONMENT_DATA_SOLUTIONS_URL/api/v0/odata/PrescribedMedicationData",
  "htm": "GET",
  "iat": 1706572800,
  "jti": "unique-id-here",
  "ath": "<base64url-sha256-of-access-token>"
}
```

Pour une implémentation DPoP détaillée, consultez le [README de la collection Postman](#).

Dépannage

Erreur "invalid_client"

Cause : ID client introuvable ou non-correspondance de clé publique

Solutions :

1. Vérifiez que votre ID client est correct
2. Assurez-vous que votre clé publique est enregistrée auprès du serveur d'authentification
3. Confirmez que la clé privée correspond à la clé publique enregistrée

Erreur "invalid_grant"

Cause : Échec de la validation de l'assertion JWT

Solutions :

1. Vérifiez l'expiration JWT (revendication `exp`) - assurez-vous qu'elle se trouve dans le futur
2. Vérifiez que la revendication `aud` correspond exactement à l'URL du point de terminaison du jeton
3. Assurez-vous que `iss` et `sub` contiennent tous deux votre ID client
4. Confirmez que le JWT est signé avec l'algorithme RS384
5. Vérifiez que le `jti` du JWT est unique

401 Non autorisé lors des requêtes API

Cause : Jeton d'accès invalide ou expiré

**Solutions :**

1. Vérifiez que le jeton n'a pas expiré
2. Vérifiez que vous utilisez le type de jeton `Bearer`
3. Assurez-vous que le jeton a été demandé avec `scope=DsaApi`
4. Confirmez le format de l'en-tête `Autorisation : Bearer <token>`

403 Interdit lors des requêtes API

Cause : Jeton valide, mais permissions insuffisantes

Solutions :

1. Vérifiez que votre client a acheté le produit/le forfait
2. Vérifiez si les colonnes que vous sélectionnez sont comprises dans votre forfait
3. Assurez-vous d'inclure un filtre `PARTITION_DAY` valide
4. Examinez les restrictions de filtre du forfait

Délai d'expiration de la demande de jeton

Cause : Problèmes de connectivité réseau ou de serveur

Solutions :

1. Vérifiez la connectivité réseau vers le serveur d'authentification
2. Vérifiez que l'URL du serveur d'authentification est correcte
3. Vérifiez que les règles du pare-feu autorisent les connexions HTTPS sortantes

URL des environnements

Environnement	URL du serveur d'authentification	URL de Data Solutions API
Production	<code>https://auth-server.surescripts.net</code>	<code>https://data-solutions-api.surescripts.net</code>

Meilleures pratiques de sécurité**1. Protégez les clés privées**

- Ne validez jamais les clés privées dans le contrôle de source
- Stockez les clés de manière sécurisée (p. ex., variables d'environnement, coffres de clés)
- Utilisez des clés différentes pour différents environnements

2. Stockage des jetons

- Stockez les jetons de manière sécurisée en mémoire ou dans un stockage chiffré
- N'enregistrez jamais et n'exposez jamais les jetons dans les messages d'erreur
- Supprimez les jetons lorsqu'ils expirent

3. Réclamations JWT

- Utilisez toujours des valeurs `jti` uniques (UUID)



- Gardez des durées d'expiration courtes (5 minutes recommandées)
- Validez l'expiration du jeton avant utilisation

4. HTTPS uniquement

- Utilisez toujours HTTPS pour les demandes de jetons et les appels API
- Vérifiez les certificats SSL/TLS

5. Gestion des erreurs

- Implémentez une logique de nouvelle tentative pour les échecs temporaires
- Enregistrez les échecs d'authentification à des fins de surveillance
- N'exposez pas de détails sensibles dans les erreurs destinées aux clients

Étape 6 : Télécharger les fichiers de données (formats optionnels)

L'API Data Solutions prend en charge des formats optionnels de téléchargement de fichiers qui peuvent être activés pour votre package de données. Les formats disponibles dépendent de la configuration de votre package.

Formats pris en charge

- **CSV** – Format lisible par l'humain pour les téléchargements simples et la révision manuelle
- **Parquet** – Format binaire en colonnes optimisé pour l'analytique et le traitement à grande échelle

Remarque : La disponibilité des formats de fichier est contrôlée au niveau du package par Surescripts. Si un format n'est pas activé pour votre package, il n'est pas disponible au téléchargement.

Avant de commencer

Confirmez que :

- Votre Client ID est configuré avec un package prenant en charge le format de fichier souhaité
- Vous avez terminé l'authentification dans **Étape 5**
- Vous avez une valeur **PARTITION_DAY** valide

La sélection du format de fichier n'**affecte pas** la construction des requêtes ni l'authentification.

Télécharger des données CSV (uniquement pour les packages CSV activés)

Envoyez une requête au point de terminaison suivant pour télécharger des données CSV à l'aide d'une URL signée. L'URL signée est active pendant 5 minutes et n'est disponible que si les téléchargements CSV sont activés pour vous.

Données sur les médicaments prescrits :

 CSV Download Endpoint

`/api/v0/customer/product/PrescribedMedicationData/csv/signedurls/{PARTITION_DAY}`

Télécharger des données Parquet (uniquement pour les packages Parquet activés)



Envoyez une requête au point de terminaison suivant pour télécharger des données Parquet à l'aide d'une URL signée. L'URL signée est active pendant 5 minutes et n'est disponible que si les téléchargements Parquet sont activés pour vous.

Données sur les médicaments prescrits :

 Parquet Download Endpoint

`/api/v0/customer/product/PrescribedMedicationData/parquet/signedurls/{PARTITION_DAY}`



3.2. Aruba 660-330 Linxusme3s

JS JS

Code Section Title

C-Like

Responses

Aruba Aruba Aruba 660-300

Aruba



3.2.1. Tester le saut du curseur Adobe sur les tableaux

Segment 1 : Fondation de la gestion intelligente des ressources

Durée : 25 minutes

Document de test

Ce segment explore comment AEM Assets combine l'organisation des dossiers, la génération de métadonnées alimentée par l'IA et les métadonnées de workflow Workfront afin de créer un référentiel de ressources gouverné et consultable. Vous découvrirez les capacités de recherche par IA, notamment les requêtes en langage naturel, la recherche sémantique et la recherche contextuelle, et apprendrez à préparer les ressources pour la distribution.

Exercice 1.1 : Organisation des ressources et hiérarchie des dossiers

Durée : 4 minutes

Comprendre la structure des dossiers dans AEM Assets

AEM Assets utilise une hiérarchie de dossiers structurée qui reflète les besoins de votre organisation, que ce soit par marque, campagne, région ou type de ressource. Cette structure fournit la gouvernance et les autorisations tandis que l'IA rend tout consultable, peu importe l'emplacement des ressources. La hiérarchie des dossiers permet le contrôle d'accès basé sur les rôles et crée une base pour la navigation manuelle et la découverte alimentée par l'IA.

Valeur pour votre organisation :

- La hiérarchie des dossiers permet le contrôle d'accès basé sur les rôles entre les équipes
- Une structure cohérente entre les campagnes simplifie la navigation pour tous les utilisateurs
- Prend en charge les exigences de conformité avec une propriété claire des ressources

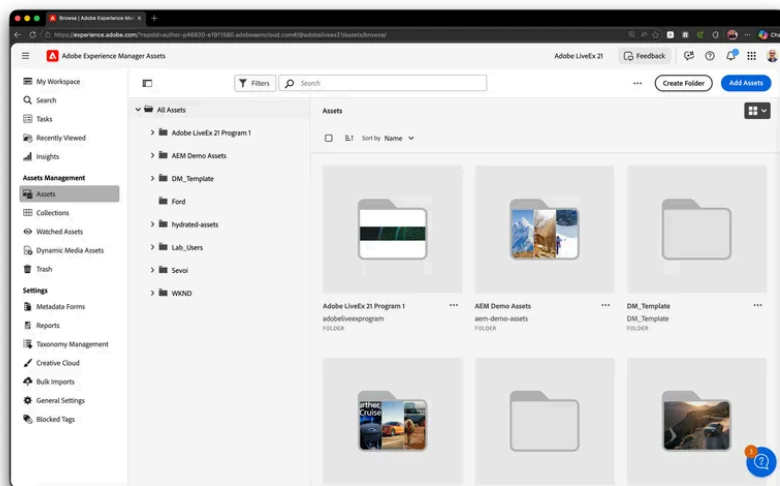
Données de référence : Structure des dossiers

Niveau	Chemin d'exemple	Objectif
Marque	/content/dam/sevoi	Référentiel de ressources au niveau de la marque
Campagne	/content/dam/sevoi/campaigns/escape-beyond-ordinary	Ressources spécifiques à la campagne
Région	/content/dam/sevoi/campaigns/escape-beyond-ordinary/apac	Variations régionales
Type de ressource	/content/dam/sevoi/campaigns/escape-beyond-ordinary/apac/photography	Catégorisation du contenu

Instructions étape par étape

Étape 1 : Ouvrir AEM Assets

- Connectez-vous à votre environnement AEM as a Cloud Service
- Accédez à la vue Assets à l'aide de la navigation de gauche
- La vue Assets fournit une interface simplifiée axée sur le marketing



Cliquez pour prévisualiser

Étape 2 : Accéder au dossier de marque Sevoi Resorts

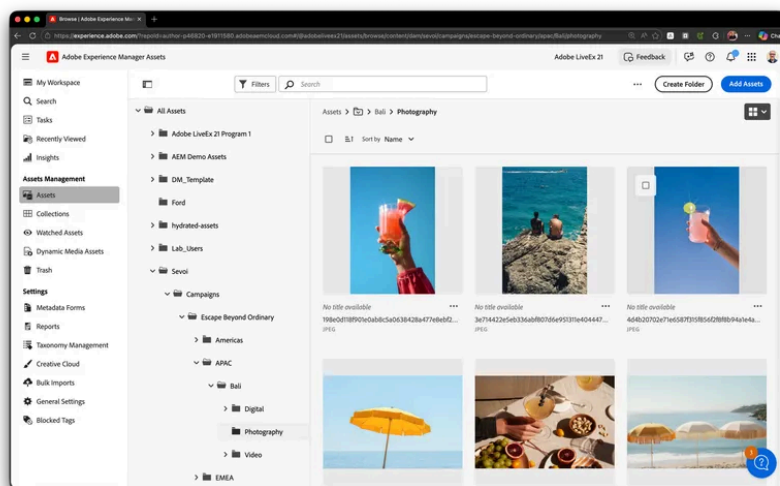
- Dans la vue Assets, localisez la structure de dossiers content/dam
- Cliquez sur le dossier de marque Sevoi
- Il s'agit du référentiel au niveau de la marque pour toutes les ressources Sevoi Resorts

Étape 3 : Développer le dossier campaigns

- Cliquez dans le dossier campaigns
- Localisez le dossier de campagne « escape-beyond-ordinary »
- Cliquez pour ouvrir le référentiel de ressources de cette campagne

Étape 4 : Examiner la structure des dossiers

- Observez les dossiers régionaux : APAC, EMEA, Americas, LATAM
- Notez comment chaque région possède des sous-dossiers cohérents
- Cette structure soutient la gouvernance tout en permettant l'autonomie régionale



Cliquez pour prévisualiser



Étape 5 : Accédez au dossier de la région APAC

- Cliquez sur le dossier APAC
- Celui-ci contient des ressources destinées à la région Asie-Pacifique
- Les équipes régionales disposent d'un accès approprié à leurs dossiers désignés

Étape 6 : Observez les sous-dossiers des types de ressources

- Notez les sous-dossiers photography et banners
- Cette catégorisation aide les équipes à trouver rapidement des types de ressources spécifiques

Point de contrôle : Pouvez-vous accéder au dossier photography de la campagne Bali dans la région APAC ?

Avantage en matière de gouvernance : La hiérarchie des dossiers crée des limites naturelles de permissions. Les équipes marketing régionales peuvent accéder aux ressources de leur marché tandis que les équipes de gouvernance de marque maintiennent une supervision de l'ensemble de la structure. Cela réduit le risque d'utilisation de contenu non conforme à la marque tout en permettant l'autonomie des équipes.

Exercice 1.2 : Métadonnées générées par l'IA et Smart Crops

Durée : 6 minutes

Comprendre l'intelligence des ressources alimentée par l'IA

Lorsque des ressources arrivent dans AEM Assets as a Cloud Service, l'IA d'Adobe se met immédiatement au travail. Elle génère automatiquement trois champs de métadonnées distincts, notamment un **Titre généré**, une **Description générée**, et des **Mots-clés générés** qui apparaissent dans un onglet dédié **Onglet généré par l'IA** sur la page Properties de la ressource. Ceux-ci sont distincts des métadonnées saisies manuellement ; si aucun titre manuel n'est défini, AEM utilise par défaut le titre généré par l'IA dans les vues en cartes et en listes.

Au-delà des métadonnées et du balisage, **Rognage intelligent** alimenté par Adobe Sensei via Dynamic Media identifie le point focal de chaque image et pré-génère automatiquement des rendus pour les rapports d'aspect courants. Cela signifie que des variations prêtes pour les canaux sont disponibles sans générer de fichiers stockés distincts pour chaque format.

Valeur pour votre organisation :

- Les métadonnées générées par l'IA offrent une couverture taxonomique plus large pour la découverte et le filtrage
- Les Smart Crops éliminent le travail de production répétitif pour les formats de canaux courants
- Les ressources sont immédiatement utilisables sur tous les canaux sans intervention d'un designer

Données de référence : Champs de métadonnées générés par l'IA



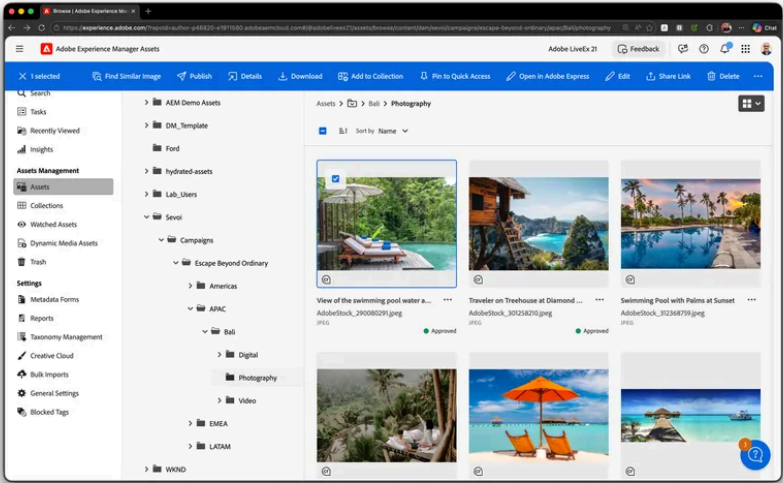
Champ	Valeur d'exemple	Où cela apparaît
Generated Title	Piscine tropicale au coucher du soleil	AI Generated tab dans les Propriétés de la ressource
Generated Description	Palmiers en silhouette devant un coucher de soleil se reflétant dans une piscine à débordement d'un complexe hôtelier de luxe	AI Generated tab dans les Propriétés de la ressource
Generated Keywords	Resort, Coucher de soleil, Palmier, Piscine, Tropical, Luxe, Voyage, Vacances	AI Generated tab dans les Propriétés de la ressource
Smart Crop	2:1 (bannière), 1:1 (carré social), 4:5 (portrait social), 5:8 (story)	Rendus Dynamic Media Smart Crop

Remarque sur les Smart Crops et les rendus dynamiques : Les rendus Smart Crop sont distincts des rendus statiques comme les miniatures. Les rendus statiques sont des fichiers pré-générés stockés dans le DAM. Les rendus Smart Crop sont fournis dynamiquement et les coordonnées de recadrage sont stockées et appliquées au moment de la diffusion, ce qui signifie qu'un seul fichier source sert tous les formats de canal sans générer de copies stockées distinctes.

Instructions étape par étape

Étape 1 : Sélectionner une ressource photographique de complexe hôtelier à Bali

- Accédez au dossier de photographie dans la campagne Bali
- Cliquez sur l'une des images de complexe hôtelier en haute résolution
- Sélectionnez une image avec des éléments visuels forts (coucher de soleil, piscine, architecture)



Cliquer pour prévisualiser

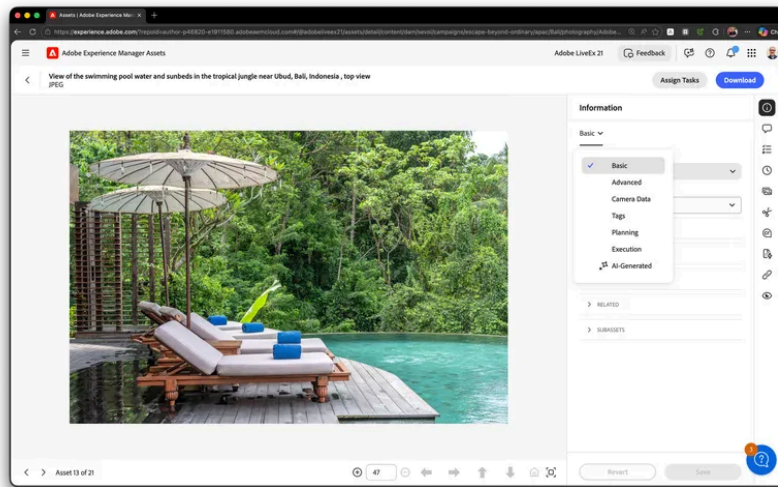
Étape 2 : Ouvrir le panneau des détails de la ressource

- Avec la ressource sélectionnée, le panneau des détails devrait apparaître dans la barre de menu supérieure
- Ce panneau affiche toutes les métadonnées associées à la ressource

Étape 3 : Localiser l'onglet de métadonnées AI Generated

- Dans les Propriétés de la ressource, recherchez l'onglet AI Generated

- Cet onglet est distinct de l'onglet Basic où se trouvent les métadonnées manuelles
- Les champs ici ont été remplis automatiquement lorsque la ressource a été téléchargée



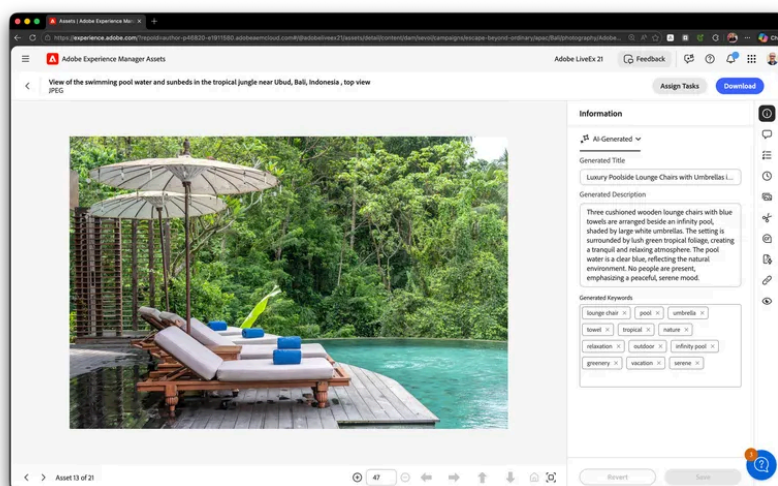
Cliquer pour prévisualiser

Étape 4 : Examiner le Generated Title et la Generated Description

- Lisez le titre généré par l'IA, qui devrait décrire clairement le sujet principal de l'image
- Examinez la description pour vérifier l'exactitude et le niveau de détail de la scène

Étape 5 : Examiner les Generated Keywords

- Trouvez le champ Generated Keywords dans l'onglet AI Generated
- Examinez la liste des termes générés par l'IA représentant les principaux thèmes de la ressource
- Ces mots-clés sont indexés pour la recherche en parallèle des Smart Tags
- Vous pouvez ajouter ou modifier manuellement les mots-clés si certains termes pertinents sont manquants



Cliquer pour prévisualiser

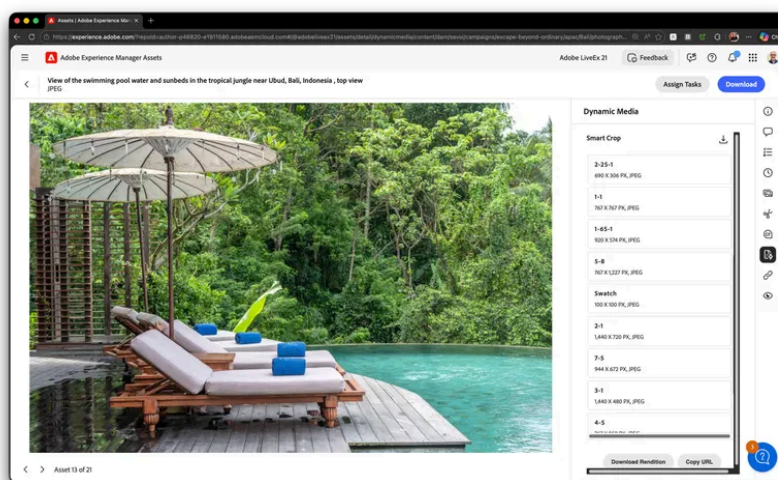
Étape 6 : Cliquer sur la section Smart Crops ou Renditions



- Localisez le panneau Renditions via l'icône Dynamic Media dans le panneau de droite
- Recherchez les rendus Smart Crop dans la liste

Étape 7 : Prévisualiser chaque Smart Crop pré-généré

- Parcourez les rendus Smart Crop disponibles :
 - 2:1 pour les bannières web
 - 1:1 pour les carrés de médias sociaux
 - 4:5 pour les publications sociales en portrait (fil Instagram)
 - 5:8 pour le format Stories



Cliquer pour prévisualiser

Étape 8 : Observer la cohérence du point focal

- Comparez comment le sujet principal reste visible et centré dans tous les recadrages
- Chaque recadrage conserve l'impact visuel de l'original sans intervention manuelle
- Si nécessaire, cliquez sur Edit Smart Crop pour ajuster manuellement le point focal pour n'importe quel ratio

Point de contrôle : Pouvez-vous voir les variations Smart Crop pour votre image sélectionnée ? Tous les recadrages conservent-ils le point focal ? Pouvez-vous localiser le Generated Title, la Description et les Keywords dans l'onglet AI Generated ?

Exemple de cas d'utilisation : Une marque mondiale du secteur hôtelier téléverse 500 images provenant d'une séance photo immobilière. L'IA a généré des métadonnées consultables pour chaque image et créé 2 000 rendus dynamiques pré-recadrés (4 recadrages x 500 images). Ce qui aurait pris une semaine à une équipe de production se produit automatiquement, libérant les ressources créatives pour un travail stratégique.

Exercice 1.3 : Intégration des métadonnées Workfront

Durée : 4 minutes

Comprendre les métadonnées provenant des workflows



L'IA peut nous indiquer ce qui se trouve dans une image, mais elle ne peut pas nous dire pourquoi elle a été créée. Ce contexte provient de Workfront. Lorsque les ressources sont synchronisées à partir du flux de travail créatif via l'intégration AEM-Workfront native, elles apportent des métadonnées de campagne, des informations de marque, un ciblage régional et des dates de lancement. Cette combinaison de métadonnées générées par l'IA et provenant du flux de travail crée un portrait complet qui alimente la gouvernance, les rapports et l'activation en aval.

Valeur pour votre organisation :

- Le contexte de campagne permet les rapports et la mesure du ROI
- Les dates de lancement prennent en charge la gestion du cycle de vie du contenu et l'expiration
- Une source unique de vérité provenant de Workfront élimine les silos de données

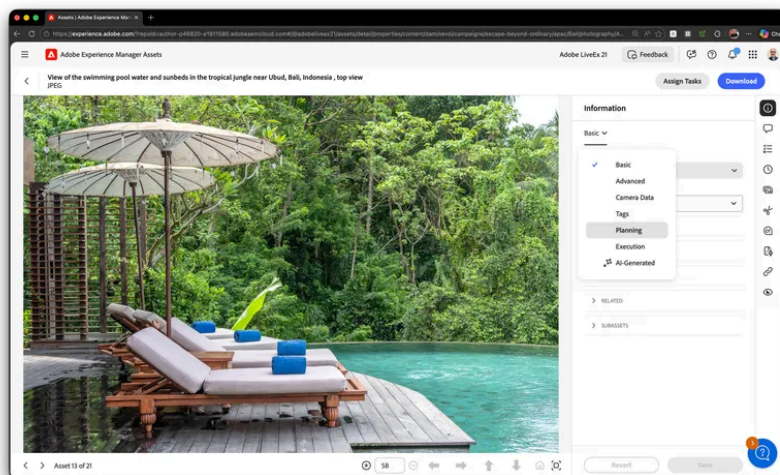
Données de référence : métadonnées enrichies de Workfront

Champ	Valeur	Source
Nom du portfolio	Laboratoire pratique	Projet Workfront
Nom du programme	Laboratoire client	Projet Workfront
Nom du projet	Ben - Échapper à l'ordinaire	Projet Workfront
Numéro de référence du projet	24345	Projet Workfront
Campagne	Échapper à l'ordinaire	Planification Workfront
Marque	Sevoi	Planification Workfront
Date de lancement	14 avril 2026	Planification Workfront
Date de fin	14 octobre 2026	Planification Workfront

Instructions étape par étape

Étape 1 : Gardez le panneau des détails de la ressource ouvert

- Avec la ressource précédemment sélectionnée toujours active
- Cliquez sur le bouton Info dans la barre latérale droite
- Ouvrez l'onglet Planning



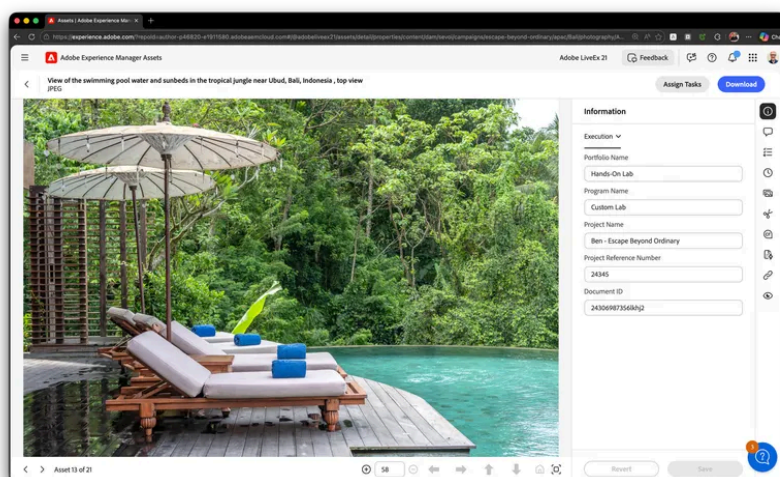
Cliquez pour prévisualiser

Étape 2 : Passez en revue les champs de Campaign Planning

- Repérez le champ Campaign affichant « Escape Beyond Ordinary »
- Ces métadonnées proviennent automatiquement du flux de travail Workfront
- Passez en revue le champ Brand affichant « Sevoi »
- Examinez les champs de synchronisation d'activation de la campagne

Étape 3 : Comparez les champs générés par l'IA et les champs de planification/exécution

- Faites défiler entre l'onglet AI Generated et les champs provenant de Workfront
- Remarquez comment l'IA vous indique le « quoi » (coucher de soleil, piscine, tropical)
- Remarquez comment Workfront vous indique le « pourquoi » (campagne, audience, synchronisation)
- Ensemble, ils créent un contexte complet pour la ressource



Cliquez pour prévisualiser

Point de contrôle : Voyez-vous comment les métadonnées de Workfront ajoutent un contexte d'affaires pour compléter les métadonnées visuelles de l'IA ?



Avantage en matière de gouvernance : Grâce à l'enrichissement automatisé des métadonnées, les équipes de conformité peuvent facilement vérifier quelles ressources sont associées à quelles campagnes, valider l'utilisation de la marque dans différents marchés et suivre l'utilisation des ressources. Lorsqu'une campagne se termine, les métadonnées de date de fin peuvent déclencher des flux de travail automatisés de révision ou d'archivage.

Exercice 1.4 : Capacités de recherche par IA

Durée : 5 minutes

Comprendre la découverte intelligente des ressources

AEM Assets offre plusieurs modes de recherche alimentés par l'IA. **Omnisearch** (la barre de recherche standard) prend en compte les Smart Tags et les champs de métadonnées générés par l'IA dans les résultats, en plus des métadonnées standard. **Recherche contextuelle** permet aux utilisateurs de saisir des requêtes en langage naturel et AEM les convertit automatiquement en filtres de recherche. Les résultats de recherche qui correspondent à tous les termes dans les champs de métadonnées s'affichent d'abord, suivis des résultats qui correspondent uniquement aux termes dans les Smart Tags.

Valeur pour votre organisation :

- Contextual Search convertit automatiquement les requêtes en langage naturel en filtres de recherche
- Les équipes mondiales peuvent effectuer des recherches dans leur propre langue et avec leur propre terminologie
- La recherche sémantique met en évidence du contenu pertinent même sans correspondances exactes de mots-clés

Données de référence : Capacités de recherche par IA

Capacité	Exemple	Fonctionnement
Compréhension des synonymes	La recherche « océan » trouve des ressources étiquetées « plage », « mer », « littoral »	Correspondance conceptuelle via les Smart Tags et les métadonnées générées
Tolérance aux fautes d'orthographe	« trupical » retourne des ressources « tropical »	Correspondance approximative dans l'index Omnisearch
Recherche multilingue	Une requête en espagnol/allemand trouve des ressources étiquetées en anglais	Traitement du langage naturel via Contextual Search
Langage naturel	« pool umbrellas for summer social media »	Contextual Search convertit la requête en filtres

Instructions étape par étape

Étape 1 : Cliquez sur la barre de recherche dans AEM Assets

- Repérez la barre de recherche en haut de la vue Assets
- Cliquez pour activer le champ de recherche
- AI Search est activé par défaut

Étape 2 : Recherchez Poolside Umbrella

- Tapez : poolside umbrella
- Appuyez sur Entrer pour exécuter la recherche

Étape 3 : Observez les résultats de correspondance des synonymes



- Passez en revue les résultats de recherche qui s'affichent
- Remarquez les ressources liées aux piscines, aux parasols de patio et aux scènes de détente extérieures
- L'IA a compris que « poolside umbrella » inclut des concepts connexes via Smart Tags et les métadonnées générées

Étape 4 : Effacez la recherche et essayez une faute d'orthographe

- Effacez le champ de recherche
- Tapez : trupical paradise (faute d'orthographe intentionnelle)
- Appuyez sur Entrer pour lancer la recherche
- Observez que les résultats affichent toujours du contenu tropical/paradis

Étape 5 : Essayez une recherche en langage naturel avec Contextual Search

- Effacez le champ de recherche
- Tapez : pool umbrellas for summer social media
- Appuyez sur Entrer et passez en revue les résultats

Étape 6 : Effectuez une recherche dans une autre langue (espagnol)

- Effacez le champ de recherche
- Tapez : sombrilla de piscina (espagnol pour « pool umbrella »)
- Appuyez sur Entrer et observez les résultats

Étape 7 : Essayez une recherche en allemand

- Effacez le champ de recherche
- Tapez : Poolschirm (allemand pour « pool umbrella »)
- Vérifiez que les résultats affichent du contenu de complexe hôtelier de luxe étiqueté en anglais

Point de contrôle : Avez-vous pu trouver des ressources pertinentes même avec la faute d'orthographe ? La recherche multilingue a-t-elle retourné les résultats attendus ?

Exemple de cas d'utilisation : Une entreprise multinationale de biens de consommation exerce ses activités dans 40 pays. Auparavant, les responsables marketing régionaux ne pouvaient pas trouver des ressources à moins de connaître les balises exactes en anglais. Grâce à la recherche IA multilingue, l'équipe du Japon effectue des recherches en japonais, l'équipe du Brésil effectue des recherches en portugais, et tout le monde trouve les mêmes ressources pertinentes.

Exercice 1.5 : Collections pour la collaboration d'équipe interne

Durée : 3 minutes

Comprendre les Collections dans AEM Assets

Les Collections permettent aux équipes d'organiser et de partager des ensembles de ressources organisés sans déplacer les fichiers ni perturber la gouvernance des dossiers. Contrairement aux dossiers, une collection peut inclure des ressources provenant de différents emplacements dans le DAM. AEM prend en charge trois types : **Visionneuse** (une collection personnelle par utilisateur), **Collections standard** (organisées manuellement et partageables), et **Collections intelligentes** (alimentées dynamiquement selon des critères de recherche enregistrés).

**Valeur pour votre organisation :**

- Les Collections standard peuvent être partagées avec des autorisations configurables d’affichage ou de modification
- Les Collections intelligentes affichent automatiquement les plus récentes ressources correspondant aux critères définis
- Les collections fonctionnent au-delà des limites des dossiers, permettant une organisation intercampagne

Données de référence : Types de collections

Type de collection	Comportement	Meilleur cas d'utilisation
Collection standard	Organisée manuellement, statique, partageable	Livrables de projet ponctuels, ensembles de ressources organisés
Collection intelligente	Dynamique, mise à jour automatique selon une requête de recherche enregistrée	Campagnes continues, ensembles régionaux de ressources approuvées

Instructions étape par étape**Étape 1 : Accédez à la vue Collections**

- Dans AEM Assets, repérez l’option Collections dans la navigation de gauche
- Cliquez pour ouvrir l’interface Collections
- Vous verrez les collections existantes organisées par types Standard Collection et Smart Collection

Étape 2 : Créer une nouvelle Smart Collection

- Cliquez sur le bouton Create Collection
- Sélectionnez Smart Collection comme type de collection
- Nommez-la : APAC Campaign Approved Assets

Étape 3 : Ajouter des critères de filtre

- Dans la configuration de la Smart Collection, appliquez les filtres suivants :
 - Brand = Sevoi
 - Region = APAC
 - Status = Approved
- Ces critères incluront dynamiquement toutes les ressources correspondantes provenant de l’ensemble du référentiel

Étape 4 : Enregistrer la collection

- Cliquez sur Save pour créer la Smart Collection
- La collection sera immédiatement remplie avec les ressources correspondantes
- Notez le nombre de ressources incluses

Étape 5 : Afficher le contenu de la collection

- Cliquez sur votre collection nouvellement créée
- Examinez toutes les ressources qui correspondent aux critères
- Notez comment les ressources provenant de différents dossiers apparaissent ensemble dans une seule vue



Étape 6 : Observer l'agrégation inter-dossiers

- Remarquez que les ressources des dossiers photography et banners apparaissent ensemble
- Les utilisateurs n'ont pas besoin de connaître la structure des dossiers pour trouver les bonnes ressources

Étape 7 : Tester la mise à jour dynamique

- Retournez à un dossier de ressources
- Approuvez une ressource APAC supplémentaire (si elle n'est pas déjà approuvée)

Étape 8 : Actualiser la collection

- Retournez à votre Smart Collection et actualisez-la
- Vérifiez que la ressource nouvellement approuvée apparaît automatiquement

Point de contrôle : Pouvez-vous voir comment votre Smart Collection inclut automatiquement des ressources APAC approuvées provenant de plusieurs dossiers ?

Exemple de cas d'utilisation : Un responsable marketing régional à Singapour ouvre sa Smart Collection « APAC Approved » et voit immédiatement chaque ressource approuvée étiquetée pour sa région. Lorsque de nouvelles ressources sont approuvées, elles apparaissent automatiquement dans un portail de ressources vivant et toujours à jour.

Exercice 1.6 : Approbation des ressources pour la distribution

Durée : 3 minutes

Comprendre le statut d'approbation comme gouvernance

Toutes les ressources du référentiel ne sont pas prêtes pour une distribution à grande échelle. Dans AEM Assets as a Cloud Service, chaque ressource peut se voir attribuer l'un des trois états de révision via le menu déroulant **Statut** dans l'onglet Basic des Propriétés de la ressource : **Approuvé**, **Rejeté**, ou **Aucun statut** (par défaut). Lorsque vous définissez une ressource comme Approved, un champ **Cible d'approbation** apparaît, vous permettant de diriger la ressource vers Content Hub, Delivery (Dynamic Media with OpenAPI), ou les deux, selon votre environnement.

Valeur pour votre organisation :

- Gouvernance précise sur les ressources qui atteignent les canaux de distribution externes
- Le champ Approval Target dirige les ressources vers le bon canal en aval
- Les indicateurs visuels rendent le statut d'approbation immédiatement clair

Données de référence : statut d'approbation



Statut	Objectif	Niveau d'accès
Approuvé	Finalisé, prêt pour la distribution	Utilisateurs de Content Hub + utilisateurs de AEM Assets (selon la cible d'approbation)
Rejeté	Signalé comme non conforme ou hors marque	Utilisateurs internes de AEM Assets uniquement
Aucun statut	Travail en cours ou en attente de révision (par défaut)	Utilisateurs internes de AEM Assets uniquement

Remarque : Lors du réglage d'une ressource sur Approuvé, le menu déroulant Approval Target détermine si la ressource est publiée dans Content Hub, Dynamic Media with OpenAPI (Delivery), ou les deux. La cible par défaut est déterminée par les services activés dans votre environnement.

Instructions étape par étape

Étape 1 : Sélectionnez une ressource photographique de complexe hôtelier à Bali

- Accédez au dossier de photographies de campagne
- Sélectionnez l'une des images de complexe hôtelier de haute qualité
- Choisissez une image que vous jugez prête pour la campagne

Étape 2 : Ouvrez le panneau des propriétés de la ressource

- Avec la ressource sélectionnée, accédez au panneau des propriétés/détails
- Accédez à l'onglet Basic

Étape 3 : Repérez le champ Status

- Faites défiler l'onglet Basic pour trouver le menu déroulant Status
- Notez l'état actuel (Aucun statut par défaut)
- Ce champ contrôle l'admissibilité à la distribution

Étape 4 : Modifiez le statut à Approuvé

- Cliquez sur le menu déroulant Status
- Sélectionnez « Approved » parmi les options (Approved, Rejected, No Status)
- Si le champ Approval Target apparaît, sélectionnez Content Hub, Delivery, ou les deux selon votre environnement

Étape 5 : Enregistrez les propriétés

- Cliquez sur Save ou Apply pour valider la modification
- Attendez la confirmation que la modification a été enregistrée
- La ressource est maintenant approuvée pour la distribution vers la cible sélectionnée

Étape 6 : Notez l'indicateur visuel

- Observez le badge visuel indiquant le statut approuvé sur la carte de la ressource
- Cela rend le statut immédiatement visible pendant la navigation
- Les autres utilisateurs peuvent rapidement identifier les ressources approuvées



Étape 7 : Sélectionnez 2 à 3 ressources de campagne supplémentaires

- Maintenez Ctrl/Cmd enfoncé et cliquez pour sélectionner plusieurs ressources
- Choisissez des ressources prêtes pour la campagne

Étape 8 : Approuvez en lot les ressources sélectionnées

- Avec plusieurs ressources sélectionnées, accédez au panneau Bulk Metadata Edit
- Modifiez le champ Status à « Approved » pour toutes les ressources sélectionnées
- Enregistrez pour appliquer la modification à toutes les ressources simultanément

Point de contrôle : Pouvez-vous voir l'indicateur de statut approuvé sur les ressources sélectionnées ?

Avantage de gouvernance : Le statut d'approbation crée une séparation claire entre le contenu en cours de réalisation et les ressources prêtes pour la distribution. Les équipes créatives peuvent continuer à itérer sans risque d'utilisation prématurée. Lorsque les ressources reçoivent une approbation, elles deviennent automatiquement accessibles à un public plus large par l'intermédiaire de Content Hub, permettant le libre-service tout en maintenant le contrôle de la marque.



4. copier-coller de test



4.1. 5.4.1 Exigences des éléments de DocumentRetrieveRequest

Informations API

oknjknjknj

Le DocumentRetrieveRequest est un message GET qui utilisera les informations de point de terminaison API de service web suivantes pour récupérer un document via Surescripts.

Opération	Description du point de terminaison API	Description
Récupération de document	GET [base]/Binary?documentId=&repositoryId>&homeCommunityId=	Retourne le document sous forme binaire.

Remarque: En tant que répondeur, vous serez responsable de définir les informations de point de terminaison et l'URL afin que d'autres puissent récupérer des documents depuis votre système. Ces informations sont retournées dans la DocumentSearchResponse de votre système.

Types de contenu et encodages

Selon la section 3.1.0.1.9 de la spécification FHIR® Content Types and encodings,

« Le type MIME formel pour les ressources FHIR® est application/fhir+json ou application/fhir+xml. Le type MIME correct DOIT être utilisé par les clients et les serveurs. » Plus loin dans cette même section, « L'encodage UTF-8 DOIT être utilisé pour le type MIME application/fhir. Cela PEUT être spécifié comme paramètre de type MIME pour le type MIME application/fhir, mais n'est pas requis » parce que FHIR® utilise UTF-8 pour tous les corps de requête et de réponse. Puisque la spécification HTTP (section 3.7.1) définit un encodage de caractères par défaut de ISO-8859-1, les requêtes et réponses DOIVENT explicitement définir l'encodage de caractères sur UTF-8 à l'aide du paramètre charset du type MIME dans l'en-tête Content-Type. Les requêtes PEUVENT également spécifier ce paramètre charset dans l'en-tête Accept et/ou utiliser l'en-tête Accept-Charset.

En raison de ces exigences, les requêtes/réponses devront inclure un Content-Type dans l'en-tête HTTP. Ci-dessous se trouve un exemple :

- Content-Type: application/fhir+json; charset=utf-8

Remarque: charset=utf-8 n'est pas requis par Surescripts, mais est pris en charge s'il est envoyé.

Contenu de l'en-tête Accept c. paramètre de requête the_format

Les spécifications FHIR R4 indiquent que les documents doivent être retournés selon la façon dont ils ont été demandés, soit via des Bundles avec des contenus encodés en Base64 dans un bundle, soit comme contenu diffusé en continu. Le champ d'en-tête de requête Accept dans une requête de récupération de document peut être utilisé pour spécifier les types de médias acceptables pour chaque réponse.

- Si « application/fhir+xml » ou « application/fhir+json » est spécifié dans le champ Accept de l'en-tête d'une requête, le document sera retourné via des bundles FHIR.
 - Les documents (ex. : CCDAs, PLQs, fichiers binaires) retournés aux demandeurs de documents via des bundles dans les réponses FHIR doivent être encodés en Base64



- Si un autre type est utilisé, ex. : « application/pdf » (ou d'autres types binaires), le document demandé sera diffusé directement au demandeur

Comme alternative, le produit FHIR® RLE prend également en charge l'utilisation du paramètre « _format », tel que décrit dans la section 3.1.0.1.11 de la spécification FHIR®, permettant aux demandeurs de documents de spécifier leur préférence pour recevoir les documents demandés dans un format autre que celui inclus dans l'en-tête HTTP (dans les situations où une application demandeuse peut contrôler le contenu de ses requêtes, mais pas l'en-tête HTTP en raison de l'utilisation d'un logiciel de communication tiers ou d'autres limitations).

Exemple de requête avec le paramètre « _format » :

`https://rle.surescripts.net/fhir/R4/Binary?documentId=CCDA123&repositoryId=urn:`

`oid:44.5555.66=urn:oid:77.8888.99&`

`_format=application/fhir+json`

Veuillez noter que l'utilisation du paramètre « _format » dans la requête remplace tous les types de médias de requête spécifiés dans le champ Accept de l'en-tête de message de la requête.

5.4.10 Exemple de DocumentRetrieveRequest

L'exemple suivant d'URL et d'en-tête HTTP retournerait une réponse groupée au format JSON.

`https://rle.surescripts.net/fhir/Binary?`

`documentId=2.16.840.1.113883.3.2054.2.1.128118651&repositoryId=2.16.840.1.113883.3.2054.2.1&homeCommunityId=2.16.840.1.113883.3.2054.2.1`

```
Accept: application/fhir+json
Accept-Encoding: gzip, deflate
Connection: keep-alive
Content-Type: application/fhir+json
Originator-Assigning-Authority: 2.11.222.3.444444.5.666.7.2
Originator-Home-Community-Id: urn:oid:aFRVAy
Originator-Message-Id: 3bb7cae9-e835-4844-a3ae-1c41abcea564
Originator-Organization-Name: Logiciel médical ABC
Originator-Patient-Identifier: 1002318246
Originator-User-Id: SMJ01234579
Originator-User-Name: Utilisateur ABC
Originator-User-Role-Code: 106289002
Purpose-Of-Use-Code: TREATMENT
User-Agent: Apache-HttpClient/4.5.13 (Java/11.0.5)
X-SSL-Client-Cert-Issuer: Émetteur de certificat ABC
X-SSL-Client-Cert-Subject: Sujet du certificat ABC
```

5.4.11 Exigences des éléments de DocumentRetrieveResponse

Les documents seront retournés selon la façon dont ils ont été demandés (voir DocumentRetrieveRequest), soit via des Bundles avec des contenus encodés en Base64 dans un bundle, soit comme contenu diffusé directement au demandeur dans le format (sans encodage) spécifié par le contentType du document demandé.

Le tableau ci-dessous présente les exigences pour la ressource Bundle utilisée dans le message DocumentRetrieveResponse.



Remarque : Ce guide inclut des éléments de données uniquement lorsque Surescripts a des exigences spécifiques ou fournit des explications supplémentaires sur l'utilisation du champ. Reportez-vous à la ressource Bundle FHIR pour obtenir une liste complète des champs.



Give this API endpoint a title...

Obtenir un gâteau par son ID

GET<https://api.cakes.com>**Request**

Response

BODY PARAMETERS

resource type

String

required

Identifie ceci comme une ressource Bundle.

id

String

optional

Identifiant logique du bundle de réponse.

type

String

required

Indique l'objectif de ce bundle – comment il doit être utilisé.

Valeur fixe : searchset

entry

String

required

Résultats de recherche renvoyés par l'opération DocumentSearch.

resource

String

optional

Une ressource qui représente les données d'un artefact brut unique sous forme de contenu numérique accessible dans son format natif. Une ressource Binary peut contenir n'importe quel contenu, qu'il s'agisse de texte, d'image, de pdf, d'archive zip, etc.

Commentaires

En général, les ressources Binary sont utilisées pour gérer du contenu tel que :

- Documents CDA (c.-à-d. avec XDS)
- Documents PDF
- Images (la ressource Media est privilégiée pour la gestion des images, mais ce n'est pas possible lorsque le contenu est déjà binaire - p. ex. XDS).

resourceType

String

optional

Valeur fixe : Binary

id

String

optional

Identifiant logique de la ressource Binary.

contentType

String

optional

Type MIME du document (p. ex., image/jpeg, application/pdf).



data String optional

Contenu du document encodé en Base64.

JS JS

Responses

200

```
{
  "name": "Nom du gâteau",
}
```

5.4.12 Exemple de message DocumentRetrieveResponse

JS JS

```
{
  "resourceType": "Bundle",
  "id": "f98511eb-a5a2-4ecf-8ee9-c6d35bb3bb99",
  "type": "searchset",
  "entry": [
    {
      "resource": {
        "resourceType": "Binary",
        "id": "16dade3d-e45b-4be4-889b-c0ca65bc9367",
        "contentType": "image/jpeg",
        "data": "iVBORw0KGgoAAAANSUhEUgAAmQAAAC0CAYAAAA6sDC6...(Base64 content truncated for readability)...SUVORK5CYII="
      }
    }
  ]
}
```



4.2. vas duplicate

Code Section Title

Apex

Responses



5. Test des signes diacritiques

- début
- chanson

- moutarde
- lacet

Ce tableau fournit un aperçu des nouvelles fonctionnalités et améliorations disponibles dans la nouvelle plateforme Cadent.



Fonctionnalité	Détails	Hérité	Nouveau Cadeant	Améliorations
Audiences de santé	Capacité à créer une audience prédictive avec des données de santé.	✓	✓	Recherche basée sur LLM, logique ET-OU, balisage direct des annonceurs et attribution à un groupe d'annonces
Intégration 1P	Téléchargez et activez directement des données primaires.		✓	Intégration directe sans devoir passer par LiveRamp
Informations sur l'audience	Informations clés sur les ensembles de données et les audiences, y compris le sexe, l'emplacement, le revenu et l'appareil.		✓	Informations en temps réel sur les données intégrées et la composition de l'audience
Créateur d'audience	Créez une audience personnalisée basée sur des données 1P et 3P.	✓	✓	Portée en temps réel et nombre d'informations
Détails des campagnes et des éléments de campagne*	Configuration des campagnes et des éléments de campagne	✓	✓	UX rationalisée et représentation des données pour une consultation plus rapide des données. La désignation du canal des éléments de campagne automatise les paramètres en aval.
Détails du groupe d'annonces	Données de configuration de base et de performance pour les groupes d'annonces.	✓	✓	Mappage des KPI au modèle et mappage de l'optimisation à l'optimiseur
API client	Possibilité de créer et de gérer une campagne publicitaire via des API externes.		✓	API pour Advertiser, Campaign, Line Item, Ad Group et Creatives.
Diffusion d'annonces / Gestion des créations*	Créez, prévisualisez et gérez les créations dans toutes les campagnes.	✓	✓	La configuration de créations basée sur des modèles permet de rationaliser le trafic des créations
Gestion des pixels*	Créez et gérez des pixels 1P pour le suivi et la mesure des conversions de campagne.	✓	✓	Le mappage des pixels au niveau des line items permet un mappage plus granulaire des conversions KPI
Détails/journal des enchères*	Les rapports granulaires et les journaux des demandes d'enchères optimisent les stratégies d'enchères et facilitent le dépannage de la diffusion.	✓	✓	Possibilité de consulter les détails des enchères aux niveaux des line items et des groupes d'annonces
Gestion du ciblage*	Créez, appliquez et gérez des objets de ciblage complexes à travers les agences et les annonceurs.	✓	✓	Capacité pour les utilisateurs en libre-service de gérer des objets au niveau d'une agence ou d'un annonceur
Options de ciblage*	Une vaste bibliothèque de paramètres de ciblage pour l'activation des campagnes,	✓	✓	Expérience utilisateur cohérente à travers les options de ciblage



Fonctionnalité	Détails	Hérité	Nouveau Cadent	Améliorations
	tels que Emplacement, le ciblage des appareils et du contenu			
Connexions directes aux éditeurs	Intégrations directes avec les éditeurs pour un accès simplifié à l'inventaire.		✓	Cadent Inventory Manager



6. Page de test CrossRiver 3

Vous pouvez déployer Litmus Unify sur Microsoft Azure Kubernetes Services.

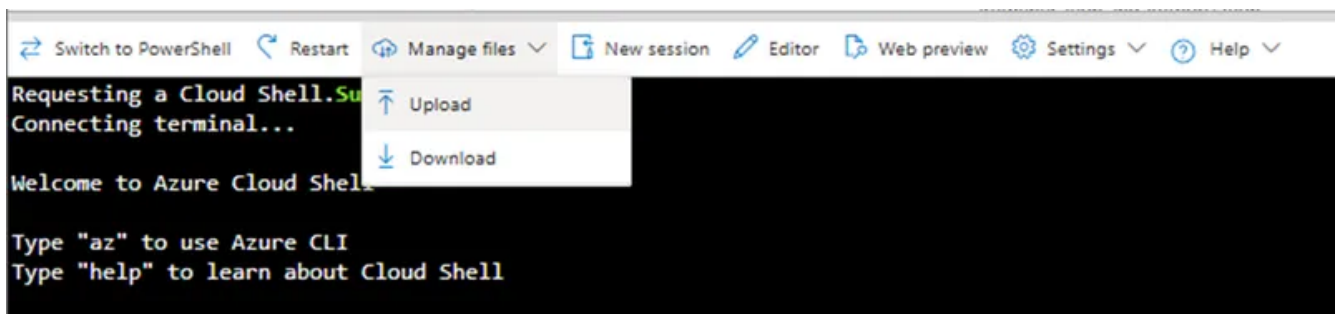
Avant de commencer

- Assurez-vous de disposer d'un cluster AKS Kubernetes en cours d'exécution.

Étapes d'installation

Pour déployer Unify sur Microsoft Azure Kubernetes Services :

- Téléchargez le fichier secret Unify **Secret K8S** (`litmusautomation-uns-centralportal-secret.yml`) depuis [Litmus Central Portal](#).
- Connectez-vous à l'instance du cluster **K8S** sur **AKS** via *Cloud Shell* ou d'autres méthodes. Téléversez ensuite le fichier `yml` ci-dessus dans le répertoire de travail.



Cloud Shell

- Créez l'espace de noms Unify en exécutant la commande `kubectl create namespace uns` dans bash.
- Créez le secret Unify en exécutant la commande `kubectl create -f litmusautomation-uns-centralportal-secret.yml --namespace=uns` dans bash.
- Exécutez la commande Helm pour l'installation avec `helm install uns oci://quay.io/litmusautomation/charts/litmus-uns --version=1.3.4 --wait --namespace uns --create-namespace --set "imagePullSecrets[0].name=litmusautomation-uns-centralportal-pull-secret"`.
- Une fois le processus d'installation terminé, l'application Unify devrait être en cours d'exécution.

Vous pouvez accéder à la webui via l'*IP externe* du service Load Balancer. Récupérez-la en exécutant la commande suivante :

Bash

```
kubectl get all -n uns
```



7. Dossier de test

7.1. recettes

sdsadasdsa		sadasdasd

asasdasdas		



Get Cakes

Obtenir un gâteau par son ID

GET<https://api.cakes.com>

Request

BODY PARAMETERS

id

String

required

ID du gâteau à obtenir

JS

```
var myHeaders = new Headers();
myHeaders.append("Accept", "application/json");
myHeaders.append("Content-Type", "application/json");

var raw = JSON.stringify({
  "id": "string"
});

var requestOptions = {
  method: 'GET',
  headers: myHeaders,
  body: raw,
  redirect: 'follow'
};

fetch("https://api.cakes.com", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

Responses

200

```
{
  "name": "Nom du gâteau",
}
```

7.2. test d'fr-CA-tête TOC manquant

Les notes de version fournissent des mises à jour sur les nouvelles fonctionnalités et les changements apportés à la plateforme Cadent dans le cadre d'une version donnée. Nous rendons les nouvelles notes de version disponibles dès qu'une nouvelle version est publiée.

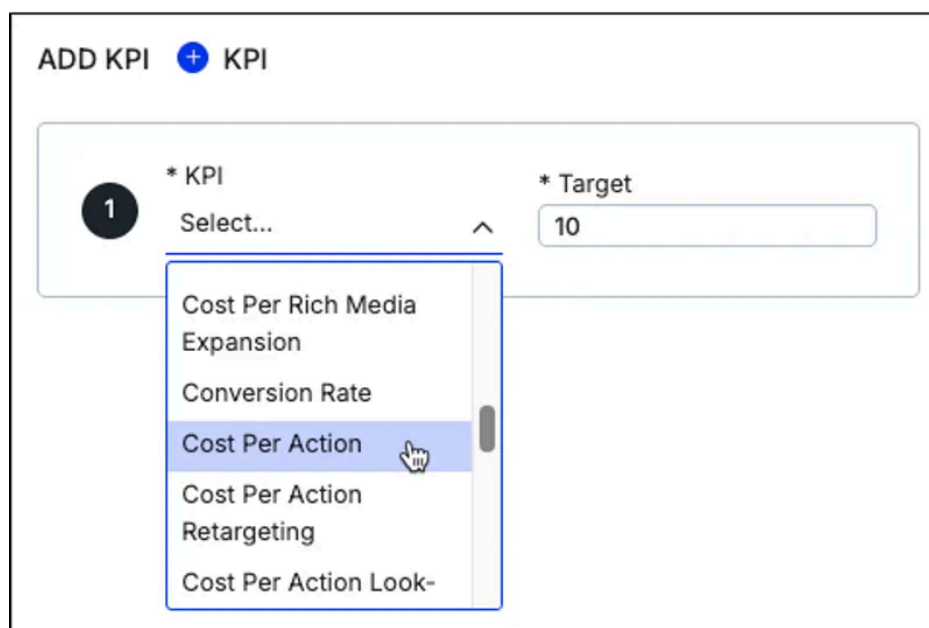
✓ 21 avril 2026

Fonctionnalité Ad Manager

KPI Cost Per Action pour les éléments de ligne

Ad Manager prend désormais en charge les modèles de coût par action (CPA), qui sont des modèles spécifiques à une campagne créés automatiquement lorsque vous sélectionnez **CPA** comme KPI pour un élément de ligne, puis mappez les activités de la campagne.

Les modèles CPA sont un outil clé pour obtenir de solides résultats KPI en fournissant des modèles basés sur la performance entraînés sur des données de conversion, comme une vente ou une installation d'application. Cela diffère des autres modèles, tels que les modèles de clics, qui optimisent les clics sans tenir compte du fait qu'ils mènent ou non à une conversion.



Cost Per Action est un KPI d'élément de ligne



✓ 20 avril 2026

Fonctionnalités d'Ad Manager

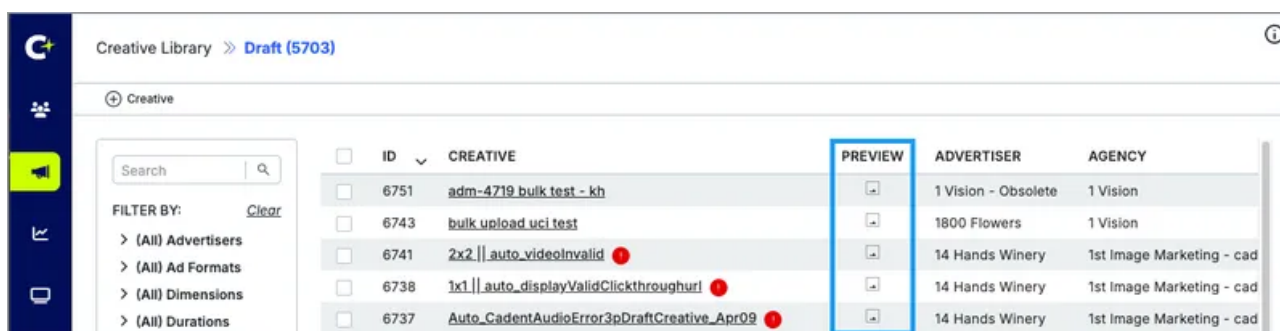
Prévisualisez et testez vos créations publicitaires brouillon

Vous pouvez maintenant prévisualiser et tester vos créations publicitaires brouillon. Cette étape de contrôle de la qualité garantit que vos créations publicitaires brouillon s'affichent et fonctionnent exactement comme prévu avant que vous ne **approuviez vos créations publicitaires brouillon**.

Apprenez-en davantage sur [la prévisualisation et le test des créations publicitaires](#).

Comment prévisualiser et tester une création publicitaire

1. Accédez à la Creative Library, puis cliquez sur l'icône **Aperçu** située à côté de la création publicitaire. Cela ouvre le panneau de prévisualisation et de test.

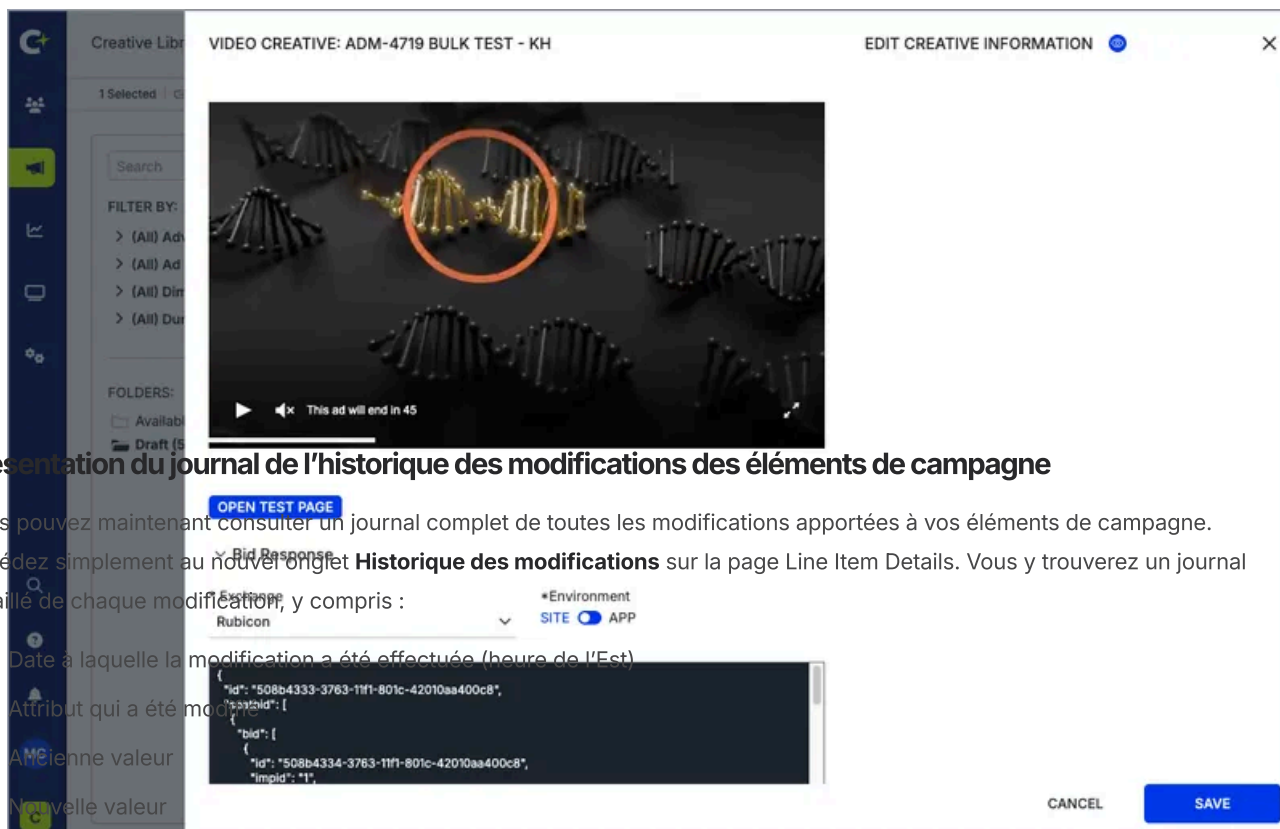


Icône Preview dans la Creative Library

2. Ici, vous pouvez effectuer les actions suivantes :
 - Cliquez sur la prévisualisation de la création publicitaire pour tester l'URL de redirection.
 - Affichez la réponse d'enchère.







VIDEO CREATIVE: ADM-4719 BULK TEST - KH EDIT CREATIVE INFORMATION

OPEN TEST PAGE

Présentation du journal de l'historique des modifications des éléments de campagne

Vous pouvez maintenant consulter un journal complet de toutes les modifications apportées à vos éléments de campagne. Accédez simplement au nouvel onglet **Historique des modifications** sur la page Line Item Details. Vous y trouverez un journal détaillé de chaque modification, y compris :

- Date à laquelle la modification a été effectuée (heure de l'Est)
- Attribut qui a été modifié
- Ancienne valeur
- Nouvelle valeur
- Personne ayant effectué la modification

Mindful campaign >> Line Item Details: Mindy's campaign_CTV (1000016730)

Details **Change History**

Change History 2

Search...

Changed On (Eastern Time)	Updated Attributes	Changed By
04/15/2026 03:25:05 PM	KPIs	Mindy Carpenter
04/15/2026 03:25:05 PM	Basics Frequency Cap Budget & Pacing 1 More	Mindy Carpenter
	Attribute	Old Value New Value
	Budget	-- \$50,000
	Channel	-- CTV
		05.01.2026 - 06.01.2026 / spend cap \$25,000 / imp
04/15/2026 03:25:05 PM	Basics Frequency Cap Budget & Pacing 1 More	Mindy Carpenter
	Flight 2	06.01.2026 - 07.01.2026 / spend cap \$25,000 / imp
	Flight Pacing	cap --
	Frequency Capping Enabled	Automatic, ahead 115%
	Metrics Primary Partner	False
	Name	Cadent
	Pacing Towards	Mindy's campaign_CTV
	Status	Spend Disabled

Onglet Change History

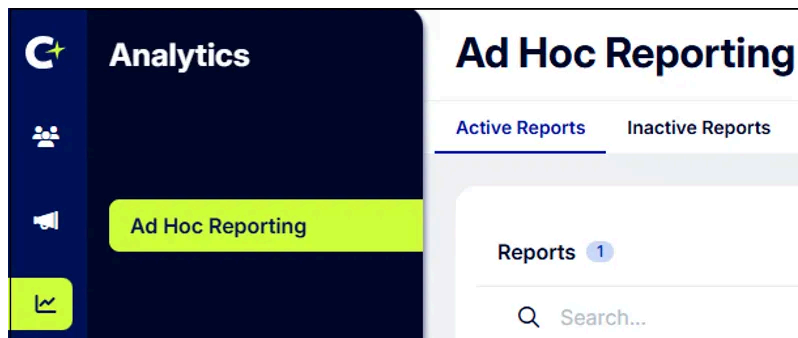
Cette fonctionnalité vous offre une plus grande transparence et facilite le dépannage en fournissant une piste d'audit claire de tous les ajustements.

Apprenez-en davantage sur le [Journal de l'historique des modifications](#).

Analytics - Fonctionnalités de rapport Ad Hoc

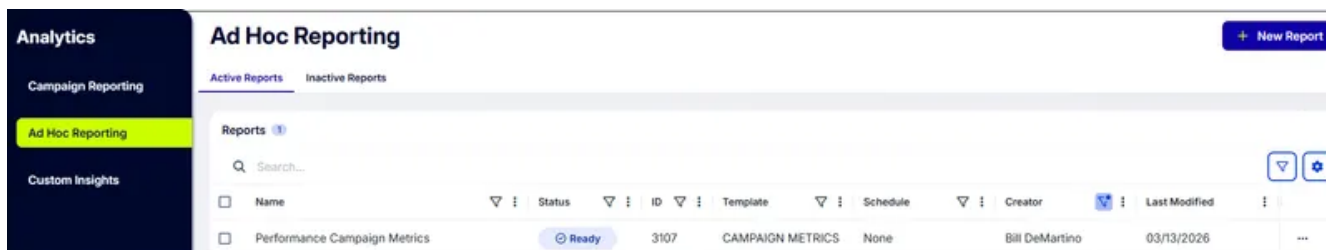
Ad Hoc Reporting a été déplacé

Vous pouvez maintenant trouver Ad Hoc Reporting en cliquant sur l'icône Analytics  dans la barre de navigation de gauche. Auparavant, il se trouvait sous Ad Manager.



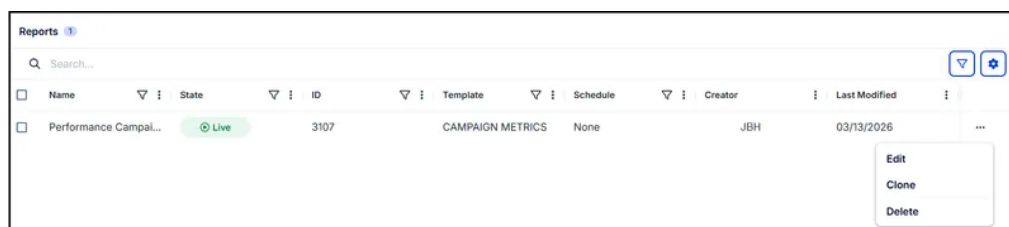
Améliorations de la page Ad Hoc Reporting

La page Ad Hoc Reporting a été mise à jour afin d'améliorer son flux de travail et son efficacité. Vous avez la possibilité d'ajouter ou de supprimer des colonnes. Des filtres plus puissants permettent d'afficher un ensemble précis de rapports et de les trier selon vos besoins. [En savoir plus.](#)



Cloner des rapports pour effectuer des modifications plus rapidement

Sur la page Ad Hoc Reporting, vous pouvez maintenant cloner un rapport. Le clonage vous permet de gagner du temps lors de la création d'un nouveau rapport en utilisant un rapport existant comme base pour le nouveau. Après avoir créé le clone, vous pouvez ensuite modifier les détails du clone afin de répondre aux besoins du nouveau rapport. [En savoir plus.](#)



✓ 6 avril 2026

Nouvelle action groupée : Mettre à jour les paramètres d'optimisation pour plusieurs groupes d'annonces

Vous pouvez désormais appliquer des modifications des paramètres d'optimisation à plusieurs groupes d'annonces simultanément, ce qui simplifie votre flux de travail de gestion de campagne et garantit l'uniformité entre vos groupes d'annonces. Auparavant, la mise à jour de ces paramètres nécessitait la modification de chaque groupe d'annonces individuellement.

Vous trouverez cette option de mise à jour groupée sur les pages suivantes :

- **Détails des campagnes** : Pour la liste des éléments de ligne qui y apparaissent
- **Liste des éléments de ligne** : Sur la page Campaigns
- **Page des détails des éléments de ligne** : Pour la liste des groupes d'annonces qui y apparaissent

Pour mettre à jour l'optimisation des groupes d'annonces de façon groupée :

1. Sélectionnez un ou plusieurs groupes d'annonces. Notez que les groupes d'annonces doivent être associés au même annonceur pour effectuer cette mise à jour groupée.
2. Cliquez sur **Optimisation de la mise à jour** pour ouvrir le panneau **Optimisation**.
3. Cliquez sur **Enregistrer** pour appliquer vos modifications à tous les éléments sélectionnés.

The screenshot displays the 'Audience Assignment Test Line Item' page. At the top, there's a navigation bar with '1B' and 'Audience Assignment Test Line Item' (Paused). Below this, a toolbar contains various action buttons like 'View Bidding Details', 'View Rate Card', 'Allocate', 'Update Status', 'Clone', 'Delete', 'Update Frequency Cap', 'Update Targeting', 'Update Optimization' (highlighted), and 'Update Audience'. The main content area is divided into sections: 'Summary' (Overview, Pacing, Insights), 'Ad Groups (3)', and a table of ad groups. The table has columns: ID, AD GROUPS, CHANNEL, ALLOCATION, YESTERDAY (USD), DAILY GOAL EST. (USD), Impressions, and CPI. The first three rows are selected, and the 'Update Optimization' button is highlighted in the top right corner.

ID	AD GROUPS	CHANNEL	ALLOCATION	YESTERDAY (USD)	DAILY GOAL EST. (USD)	Impressions	CPI
1000021001	Audience Assignment Test Ad Group 1	CTV	0%	—	0	—	—
1000021002	Audience Assignment Test Ad Group 1...	CTV	0%	—	0	—	—
1000021003	Audience Assignment Test Ad Group 1...	CTV	0%	—	0	—	—
TOTAL	—	—	0%	—	—	—	—

Mise à jour groupée de l'optimisation

Référenciez facilement vos tarifs avec le visualiseur de grilles tarifaires



Nous sommes heureux d'annoncer le lancement du nouveau visualiseur de grilles tarifaires dans Ad Manager. Cette nouvelle vue offre un accès rapide et transparent à votre grille tarifaire, réduisant le temps consacré à la recherche de vos tarifs.

Pour accéder à la grille tarifaire, cliquez sur **Voir la grille tarifaire** dans l'un des écrans suivants :

- Détails de la campagne
- Détails des éléments de ligne
- Détails des groupes d'annonces



 Visualiseur de grilles tarifaires

Visualiseur de grilles tarifaires



✓ 1 avril 2026

Notes de version d'Ad Manager

Problème résolu : Écart d'impressions du partenaire principal

La méthodologie utilisée pour calculer la différence entre les impressions suivies par Cadent et les impressions du partenaire principal (par exemple, celles suivies par un serveur publicitaire comme CM360) affichait des chiffres inexacts. Nous avons résolu ce problème, donc maintenant la colonne

Écart d'impressions PP affiche des chiffres exacts.

Pour afficher la colonne PP Imp Discrepancy :

1. Accédez à la page **Détails de la campagne** ou **Détails des postes**.
2. Sélectionnez **Écart PP** dans le sélecteur de vue.

Cadent PROD Self-Service Verification >>

Campaign Details:

Line Item

SPEND (USD)

BUDGET: 40.00
SPEND: 20.41

3 DAYS REMAINING

51%

OVERVIEW

Lifetime Yesterday Today

IMPRESSIONS	CPM (USD)	CLICKS	CTR (%)	VCR (%)
13,064	1.56	43	0.32%	19.54%

LINE ITEMS (2)

PP Discrepancy

Id	Line Item	Impressions	PP Impressions	PP Imp Discrepancy	PP Clicks	PP Conversions
1000000536	Cadent_GCM_Validation_Video	8,280	2526	-227.79	4	-
1000000535	Cadent_GCM_Validation_Dis...	10,667	3031	-251.93	3	-
TOTAL		18,947	5,557	-239.86 %	7	0

DATE RANGE: Preset Custom Lifetime

FILTER BY: Clear

> (All) Status

> (All) Advertisers

SETTINGS

BASICS BUDGET & DATES FREQUENCY CAP



✓ 23 mars 2026

Notes de version d'Ad Manager

Les filtres de rapports pour Ad Hoc Reporting ont été simplifiés

Lors de la création d'un nouveau rapport, vous pouvez désormais filtrer les rapports uniquement par agence. Aucun autre filtre n'est requis. Cela permet de produire des rapports plus complets que ce qui était auparavant possible. [En savoir plus.](#)

The screenshot shows a filter configuration window. At the top, it says 'FILTERS' with a plus icon and the word 'Filter'. Below this, there is a section titled '* Filter Type' with a dropdown menu currently set to 'Agency/Id'. To the right of the dropdown is a lock icon and the text 'equals to'.

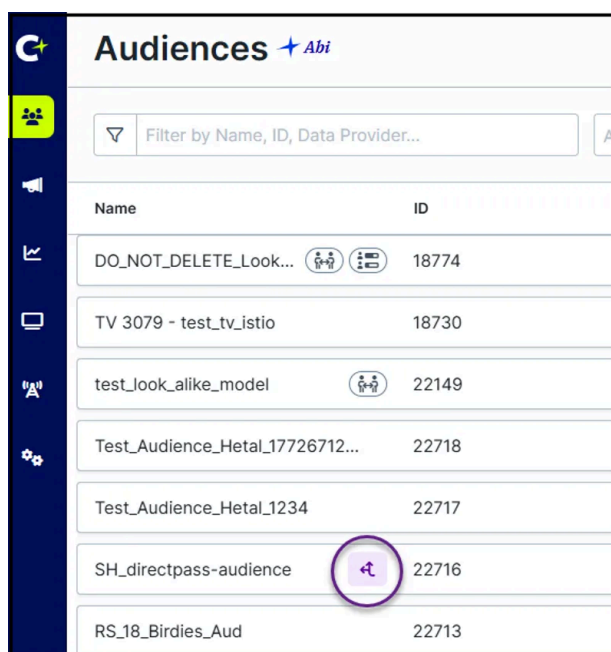
Notes de version d'Audience Manager

Des icônes DirectPass ont été ajoutées dans Audience Manager

Une nouvelle icône DirectPass  s'affiche désormais à côté des ensembles de données DirectPass dans Audience Manager afin de les rendre plus faciles à repérer. DirectPass utilise un flux de travail spécialisé d'intégration et de déploiement des ensembles de données qui permet l'ingestion et le déploiement directs d'identifiants d'appareils bruts (tels que les Device IDs et les adresses IP) sans devoir d'abord les résoudre dans le graphe des ménages de Cadent. Les ensembles de données DirectPass n'incluent pas les informations au niveau du ménage.

Les icônes DirectPass s'affichent aux emplacements suivants dans Audience Manager :

- Page Audiences
- Page Deployments
- New Audience, onglet My Datasets
- New Audience, onglet Marketplace Datasets
- New Audience, onglet Shared with Me



Name	ID
DO_NOT_DELETE_Look...	18774
TV 3079 - test_tv_istio	18730
test_look_alike_model	22149
Test_Audience_Hetal_17726712...	22718
Test_Audience_Hetal_1234	22717
SH_directpass-audience	22716
RS_18_Birdies_Aud	22713

Icône DirectPass sur la page Audiences

Les balises d'audience s'affichent sur la page Audiences

Vous pouvez désormais voir les balises d'audience avec les audiences sur la page Audiences. Les balises permettent de créer des catégories d'audiences qui partagent des caractéristiques communes. Bien que les balises ne s'affichent pas par défaut, vous pouvez les ajouter à la page à l'aide de l'icône  sur la page Audiences.

L'indicateur % Reach sur la page New Audience a été mis à jour

Dans la section **Portée totale** de la page New Audience, la colonne **% portée** a été renommée **Variation en pourcentage** afin d'améliorer la clarté. Cette colonne affiche le pourcentage de changement de la portée totale au fur et à mesure que vous



configurez une nouvelle audience. Elle est mise à jour en temps réel lorsque vous apportez des modifications à l'audience que vous créez. En savoir plus



✓ 18 mars 2026

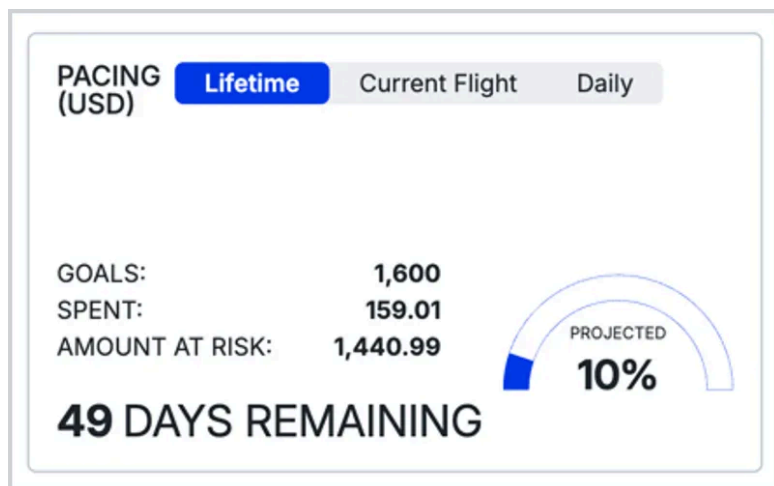
Notes de version du gestionnaire de publicités

Nouvelle vue « Lifetime » pour le rythme des éléments de ligne

Nous avons ajouté une nouvelle période **À vie** à la tuile de rythme des éléments de ligne. Cela vous permet de voir les métriques de rythme sur l'ensemble des diffusions d'un élément de ligne, vous offrant une vue plus globale des performances et vous aidant à mieux comprendre et analyser les performances de vos campagnes à long terme.

Voici un aperçu des métriques que vous verrez dans cette nouvelle vue :

Métrique	Description
Objectif	Il s'agit de l'objectif global pour l'élément de ligne, selon la manière dont votre élément de ligne est rythmé : • Si le rythme est basé sur les impressions, l'objectif reflète les impressions du vol. • Si le rythme est basé sur les dépenses, l'objectif reflète le plafond des dépenses du vol.
Dépensé	Le montant total dépensé jusqu'à hier.
Montant à risque	Cela montre la portion de l'objectif qui devrait être manquée, vous offrant une vue rapide des performances du rythme. Le montant à risque est calculé à l'aide de cette formule : $(100\% - \% \text{ projeté}) * \text{Objectif} = \text{Montant à risque}$
Jours restants	Le nombre de jours restants dans le vol actuel, sans inclure aujourd'hui.



Méthode simplifiée pour désactiver les campagnes, les éléments de ligne et les groupes d'annonces

Nous avons simplifié le processus de désactivation des éléments dans votre compte afin de rendre la gestion de vos campagnes plus rapide et plus intuitive.

Désactivation des campagnes et des éléments de ligne





Vous pouvez désormais définir directement le statut d'une campagne entière ou d'un élément de ligne sur **Désactivé** sans modifier sa configuration budgétaire sous-jacente. Auparavant, vous deviez accéder à chaque groupe d'annonces individuellement pour désactiver l'allocation. Ce changement permet de mettre vos campagnes en pause beaucoup plus rapidement, en une seule étape.

Désactivation des groupes d'annonces

Vous pouvez définir un groupe d'annonces individuel sur **Désactivé**, mais uniquement si l'allocation budgétaire active de l'élément de ligne parent totalise déjà 100 %.



Pour vérifier l'allocation budgétaire d'un élément de ligne, suivez simplement ces étapes :

1. Accédez à l'écran **Détails des postes**.
2. Sélectionnez le **Groupe d'annonces** et cliquez sur **Attribuer**. Cela ouvre le panneau de modification de l'allocation du groupe d'annonces.
3. Examinez les détails de l'allocation budgétaire et, si nécessaire, apportez des modifications afin que l'allocation totale atteigne 100 %.

EDIT AD GROUP ALLOCATION

1 Selected

☐	ID	AD GROUP	CURRENT ALLOCATION (%)	NEW ALLOCATION (%)
☐	1000000010	MW Test Campaign_Video_AdGroup 1	25	25
☐	1000000635	MW Test Campaign_Video_AdGroup 1_Clone-1	25	25
☐	1000000636	MW Test Campaign_Video_AdGroup 1_Clone-2	25	25
☐	1000000637	MW Test Campaign_Video_AdGroup 1_Clone-3	25	25
TOTAL				100%

Gagnez du temps grâce aux téléversements de pixels en lot

Nous sommes ravis d'annoncer une nouvelle fonctionnalité qui rendra votre flux de travail beaucoup plus efficace. Vous pouvez désormais créer des pixels en lot — y compris des balises universelles, des activités et des activités d'action — directement dans Ad Manager, ce qui vous évite les efforts manuels liés à leur création un par un.

Comment téléverser des pixels en lot

Suivez ces étapes pour commencer :

1. Accédez à l'écran **Pixels** à partir du menu principal.
2. Cliquez sur le bouton **+Pixels** et sélectionnez l'option **En vrac**. Cela ouvrira le panneau Add Pixels in Bulk.
3. Sélectionnez l'**Annonceur** auquel vous souhaitez associer les pixels.
4. Téléchargez le **fichier d'exemple** fourni. Ce modèle garantit que vos données sont correctement formatées pour un téléversement réussi. Ajoutez les informations de vos pixels dans ce fichier et enregistrez-le sur votre ordinateur. Si vous essayez de créer de nouvelles activités pour une Universal Tag existante, vous devriez télécharger un nouveau modèle depuis le panneau afin de vous assurer que vos Universal Tags existantes sont préremplies dans la liste déroulante de l'onglet Activities du modèle.
5. Glissez-déposez votre fichier rempli dans la zone de téléversement, ou cliquez pour le rechercher sur votre ordinateur.
6. Cliquez sur **Ajouter**.

Une fois le téléversement terminé, vous serez automatiquement redirigé vers l'écran principal Pixels, où vous verrez un message de réussite confirmant que vos nouveaux pixels ont été ajoutés.

Simplifiez votre ciblage grâce aux actions en lot

Vous pouvez désormais mettre à jour le ciblage **Catégories personnalisées**, **Catégories d'approvisionnement**, et **Réseau et chaîne** pour plusieurs groupes d'annonces en même temps.

Détails importants à retenir

- **Custom Categories** : Vous pouvez sélectionner des catégories personnalisées prédéfinies lors d'une mise à jour en lot. Les nouvelles catégories doivent être créées au préalable.
- **Network & Channel** : Cette option de ciblage est spécifiquement destinée au CTV.
- **Supply Categories** : Il s'agit de catégories d'inventaire définies par Jounce.

Fonctionnement :

1. Accédez à l'écran Line Items Details et sélectionnez le ou les groupes d'annonces que vous souhaitez mettre à jour, puis cliquez sur **Mettre à jour le ciblage**. Cela ouvre le panneau Add Targeting.

Auto_R_Campaign_Cadent ID_Mar11 >>
Line Item Details: Test_500_targeting (1000015978)

View Bidding Details View Rate Card 2 Selected Allocate Update Status Clone Delete Update Frequency Cap **Update Targeting**

SUMMARY

OVERVIEW Lifetime Yesterday Today

IMPRESSIONS CPM (USD) WIN RATE (%)

KPI 1 - KPI (USD) VCR (%)

PACING (USD) Lifetime Current Flight Daily

GOALS: SPENT: 0 AMOUNT AT RISK: 0

DAYS REMAINING PROJECTED

INSIGHTS KPI 1 KPI 2 KPI 3 BIDDING

LIFETIME AVG. THROUGH YESTERDAY: -

AD GROUPS (2) Overview

ID	AD GROUPS	CHANNEL	ALLOCATION	YESTERDAY (USD)	DAILY GOAL EST. (USD)	Impres:
1000020619	Test_500_targeting_AdGroup 1	CTV	0%	—	0	
1000020621	Test_500_targeting_AdGroup 2	CTV	0%	—	0	
TOTAL			0%	—	—	

DATE RANGE: Preset Custom
 Lifetime

2. Dans le panneau **Ajouter un ciblage**, effectuez vos nouvelles sélections de ciblage et fermez le panneau.
3. Tout est prêt ! Les nouvelles options de ciblage seront affichées dans l'écran **Détails du groupe d'annonces** pour chaque groupe d'annonces que vous avez mis à jour.



✓ 19 février 2026

Notes de version d'Ad Manager

Mise à jour des éléments de campagne : paramètre Days Ahead

Vous avez maintenant la possibilité d'indiquer si un élément de campagne doit viser une diffusion complète un certain nombre de jours avant la date de fin.

* FLIGHT PACING

Automatic ☒ Manual ☐ Ahead ☒ Even ☐ 115 %

DAYS AHEAD 1

* FLIGHTS: + Flight

FLIGHT	START & END DATE	* IMPRESSION GOAL	* IMPRESSION CAP	SPEND CAP (OPTIONAL)
1	03/01/2026 - 03/31/2026			
TOTAL				

Mise à jour du ciblage : ciblage par catégories

Vous pouvez maintenant cibler des catégories de contenu IAB ou Peer39, y compris la création et la sélection de segments Peer39 personnalisés et avancés, afin de garantir que vos publicités apparaissent sur du contenu pertinent.

Vous pouvez cibler des catégories IAB spécifiques ou des segments Peer39, ou vous pouvez exclure des catégories afin d'empêcher vos publicités d'apparaître sur du contenu indésirable.

ADD CATEGORIES Market Rate CPM

Custom Advanced Custom

SEGMENT ID 0/30

Search Selected (6)

ID	NAME	TYPE	RATE (
296077542	Custom Segment	Peer39	0.3
680163062	Custom Segment	Peer39	0.3
989898	Custom Segment	Peer39	0.3
111222333	Custom Segment	Peer39	0.3
414141	Custom Segment	Peer39	0.3
434343	Advanced Custom Segment	Peer39	0.35
7654321	Custom Segment	Peer39	0.3
6754321	Advanced Custom Segment	Peer39	0.35

Mise à jour du ciblage : ciblage par genre

Le ciblage par genre vous permet d'aligner vos campagnes sur des types spécifiques de contenu vidéo CTV. Cette fonctionnalité vous permet de cibler ou d'exclure du contenu selon le genre — par exemple, comédie, sports, drame, documentaires et true crime. Vous pouvez soit inclure des genres spécifiques à cibler, soit exclure des genres que vous souhaitez éviter.











ADD GENRE

Search

Q

GENRE

+

Action/ Adventure

+

Adult

+

Animation

+

Comedy

+

Crime

+

Documentary

+

Drama

Search

Selected (6)

Mise à jour du ciblage : ciblage par langue de l'appareil

Le ciblage par langue de l'appareil vous permet de diffuser des publicités spécifiquement aux utilisateurs dont les appareils sont configurés dans une langue particulière. C'est une façon simple de vous assurer que votre message est pertinent et compris par votre public cible, particulièrement dans les régions multilingues ou les campagnes mondiales.

ADD DEVICE LANGUAGE

Search

☒ **DEVICE LANGUAGE NAME**

- ☒ Dutch
- ☒ English
- ☐ Estonian
- ☐ Filipino
- ☒ Finnish
- ☐ French

Search Selected (5)

Mise à jour du ciblage : ciblage par langue du contenu

Cette méthode vous permet de cibler les spectateurs en fonction de la langue principale du contenu CTV lui-même. Par exemple, vous pouvez choisir de faire apparaître vos publicités uniquement pendant des programmes en espagnol, indépendamment des paramètres de l'appareil du spectateur.

Vous pouvez choisir soit de cibler une langue, soit de l'exclure.

ADD CONTENT LANGUAGE +\$2.45 USD

Search

CONTENT LANGUAGE

- Chinese
- Croatian
- Czech
- Danish
- Dutch
- English
- Estonian
- Filipino

Search Selected (6)

Mise à jour du ciblage : ciblage Geo POI

Vous pouvez cibler des POI via le ciblage géographique, lesquels sont présentés à l'utilisateur sous forme agrégée plutôt qu'en sélection de points uniques. Par exemple, en un clic, vous pouvez cibler 4 000 Starbucks au lieu de devoir les sélectionner un à un.

EDIT GEOGRAPHY

Mise à jour du ciblage : reciblage de plateforme

Le reciblage de plateforme vous permet de cibler des utilisateurs qui ont déjà été exposés à vos campagnes.

ADD PLATFORM RETARGETING

- Vous pouvez choisir soit un ciblage positif (inclure), soit un ciblage négatif (supprimer) pour l'audience sélectionnée.
- Fréquence utilisateur : Cela vous permet de définir une plage minimale et maximale pour le nombre de fois qu'une personne a vu votre publicité. Par exemple, si vous définissez un minimum de 2 et un maximum de 5, le système ciblera uniquement les utilisateurs qui ont vu la publicité au moins 2 fois, mais pas plus de 5 fois.

- Cible : Il s'agit de l'identifiant utilisé pour compter la fréquence. Vous pouvez choisir de suivre les utilisateurs par leur adresse IP, un Cookie de navigateur ou Tous (qui utilise les deux méthodes).

Mise à jour du ciblage : Actions en lot

Vous pouvez maintenant sélectionner plusieurs groupes de publicités dans la vue des détails de l'élément de ligne afin d'appliquer des actions de ciblage en lot (voir la capture d'écran ci-dessous). Les sélections que vous effectuez remplaceront les sélections précédentes.

Les options de ciblage que vous pouvez mettre à jour en lot sont :

- Liste d'applications/sites
- Packages
- Géographie
- Ententes
- Mots-clés

Test Camp For Display Status 2.23 >>>
Line Item Details: NewCampaignFeb26_Display_Clone-1 (1000015312)

View Bidding Details View Rate Card 2 Selected Allocate Update Status Clone Delete Update Frequency Cap **Update Targeting** Update Optimization

SUMMARY

OVERVIEW Lifetime Yesterday Today

IMPRESSIONS CPM (USD) WIN RATE (%)

KPI 1 - KPI (USD) CTR (%)

0 -

PACING (USD) Current Flight Daily

GOALS: 1
 SPENT:
 AMOUNT AT RISK: 1.00

43 DAYS REMAINING PROJECTED

INSIGHTS KPI

AD GROUPS (2)

Search

DATE RANGE: Preset Custom
 Lifetime

☒	ID	AD GROUPS	CHANNEL	ALLOCATION	YESTERDAY (USD)	DAILY GOAL
☒	1000019999	NewCampaignFeb26_Display_AdGroup...	Display	0%	—	—
☒	1000020000	NewCampaignFeb26_Display_AdGroup...	Display	0%	—	—
TOTAL		—	—	0%	—	—

Notes de version des rapports

Moins de filtres sont requis lors de la création d'un rapport ad hoc

Nous avons réduit le nombre de filtres que vous devez appliquer lors de la création d'un rapport Ad Hoc. Désormais, vous devez uniquement sélectionner une agence. Cela permet de créer des rapports plus larges que ceux qui étaient possibles auparavant.

En savoir plus sur [Création de rapports ad hoc](#).

BASICS

* Name

FILTERS Filter

* Filter Type

Select Type

▼

Nouvelle dimension de rapport ajoutée aux modèles Performance Insights

Pour vous permettre d'obtenir plus de détails sur vos créations dans les rapports, une nouvelle dimension de rapport, **Type de création**, a été ajoutée. Un type de création définit la façon dont un utilisateur interagit avec une publicité numérique. Par exemple, **vidéo**, **affichage**, et **bannière** sont des types de créations qui s'appliquent aux publicités en ligne. La dimension du type de création est maintenant disponible dans les modèles de rapports suivants :

- Mesures de campagne
- Mesures CTV
- Mesures vidéo
- Mesures de ciblage géographique
- Portée et fréquence
- Mesures HCP

En savoir plus sur [Modèles Performance Insights](#).

DIMENSIONS (0 selected)

Search

Search Selected

☐ Select All 29 Dimensions

☐ Agency / Partner

☐ Agency ID / Partner ID

☐ Advertiser

☐ Advertiser ID

☐ Campaign

METRICS (0 selected)

Search

Search Selected

☐ Select All 20 Metrics

☐ Clicks

☐ Completes

☐ Impressions

☐ Total Spend

☐ Agency Margin

Bidding Snapshot renommé Bidding Details

Lorsque vous accédez au rapport Bidding Snapshot, vous remarquerez qu'il a été renommé **Détails de l'offre**. Le nouveau nom s'affiche dans la barre de navigation de gauche et en haut de la page dans Ad Manager. Il s'agit du même rapport qu'auparavant, mais avec deux nouveaux ajouts décrits ci-dessous.

Bidding Details

BIDDING SNAPSHOTS

* Advertiser: AutoZone

* Campaign: RTB_Bidding_Campaign1

HOURLY VIEW

DAILY VIEW Refresh

HOUR (ET)	ELIGIBLE REQUESTS	REQUESTS REJECTED	BIDS	BID RATE (%)	W
8 AM	0	0	0	—	—
9 AM	4	0	4	100	0

Le rapport Bidding Details ajoute une vue horaire

En plus des vues Daily et Troubleshooting, vous avez maintenant l'option de sélectionner une **Vue horaire** des détails des éléments de ligne et des groupes de publicités. La vue horaire affiche l'activité de l'heure en cours et peut être actualisée pour afficher les dernières mises à jour.

Bidding Details

BIDDING SNAPSHOTS

* Advertiser: Select Advertiser

* Campaign: Select Campaign

* Line Item: Select Line Item

Ad Group: Select Ad Group

VIEW

HOURLY VIEW

MINUTE	ELIGIBLE REQUESTS	REQUESTS REJECTED	BIDS	BID RATE (%)	WIN RATE (%)	IMP GOAL	IMPS SERVED	CLICKS/COMPLETES	CTR/VCR (%)	EC
No Results Found.										

De nouveaux champs s'affichent dans le rapport Bidding Details

Les nouveaux champs suivants ont été ajoutés aux résultats du rapport Bidding Details pour les résultats de rapport horaires et quotidiens :

- **Objectif de dépenses** - La cible monétaire qu'un annonceur ou un acheteur média vise à atteindre dans le délai d'un élément de campagne ou d'un groupe d'annonces.
- **Dépenses totales** - Le montant d'argent que votre campagne ou élément de campagne a dépensé à partir du budget pour remporter des impressions.



✓ 4 février 2026

Notes de version d'Audience Manager

Ingestion de données DirectPass

DirectPass utilise un type spécialisé d'intégration de jeu de données et un flux de travail de déploiement qui permet l'ingestion directe et le déploiement d'identifiants d'appareils bruts, tels que les Device IDs et les adresses IP, sans d'abord les résoudre au graphique des foyers de Cadent. Les jeux de données DirectPass n'incluent pas les informations au niveau du foyer.

DirectPass permet des flux de travail spécialisés qui nécessitent l'utilisation d'identifiants bruts et non modifiés. Vous pouvez utiliser DirectPass dans les conditions suivantes :

- Ciblage pour des régions à l'extérieur des États-Unis, comme des segments canadiens à utiliser dans AdManager.
- Transmission directe des identifiants bruts aux partenaires de destination qui possèdent leurs propres systèmes de résolution d'identité.

Remarque : Vous devez avoir DirectPass activé pour votre organisation afin d'effectuer ce type d'intégration de données. Si cela vous intéresse, contactez votre gestionnaire de compte.

Filtre d'ID ajouté à la page Deployments

Le filtre de la page Deployments a ajouté l'option permettant de rechercher par ID de déploiement. [Détails de la page Deployments](#)

Deployments		
Deployments 2541		
Q Filter by Name, ID, Destination		
Name		ID
29_10		53447



✓ 23 janvier 2026

Note de version de Health Audience

Attribuer une audience HCP à un groupe de publicités

Qu'est-ce qu'une audience HCP?

Une audience Healthcare Professional (HCP) est un groupe cible composé de professionnels de la santé tels que des médecins, infirmières, pharmaciens et dentistes. Lorsque vous créez et enregistrez une audience HCP dans le [Générateur d'audience Health Audience](#), elle devient automatiquement disponible pour attribution dans l'interface utilisateur, tant que l'audience et le groupe de publicités sont tous deux associés au même annonceur.

Comment attribuer une audience HCP à un groupe de publicités

Vous pouvez maintenant attribuer une audience HCP à votre groupe de publicités pendant la configuration du groupe de publicités. Consultez [Ajouter un groupe de publicités](#) pour obtenir des instructions.

ADD AUDIENCES (1)		CUSTOM AUDIENCES (0)		HCP AUDIENCES (1)		TOTAL CPM: 24.00 USD	
Search		ID	AUDIENCE	HCPS	MATCHED %		
		20687	NoDeployHCPTest ✓	260,805	88%		

Écran d'attribution d'audience dans Ad Manager



8. Page de test CrossRiver

asopdkjasoifnsdijfnsadjlfnsad]

asdsadsadsadsa

Document de test



9. Page de test CrossRiver 2

Webhooks pertinents

Pour ce endpoint, assurez-vous d'avoir **Comptes, cartes et événements de paiement** pour ce webhook :

- **Rtp.Payment.Received**
- **Rtp.Payment.Expired**
- **Rtp.Payment.Rejected**

Référence API

Initie une demande de paiement (RfP)

Permet au bénéficiaire d'autoriser et de déclencher un virement de crédit par paiement instantané. Les RfP peuvent être initiées à partir de :

- Compte principal
- Compte de dépôt (DDA)
- Sous-grand livre associé



Endpoint: /rtp/v1/payments/payment-request

POST

https://sandbox.crbcos.com/rtp/v1/payments/payment-request

Request

Response

BODY PARAMETERS

accountNumber

String

required

Account from which the RfP is originated

amount

Integer

required

Requested payment amount in positive integral cents. For example, represent \$1.00 as 100.

purpose

String

optional

Internal CR field used to include a short description or purpose for the payment request. The RDFI does not see this field. 50 characters maximum.

requestedExecutionDate

String

required

Date when the payment request needs to be processed by the Debtor. Formatted as yyyy-mm-ddThh:mm:ss[.mmm].

expiryDate

String

required

Expiration date and time of the RfP. If no time is specified, it expires at 12:00 AM on the given date. If a time is provided, it expires within one hour after the specified time on the same date. Formatted as yyyy-mm-ddThh:mm:ss[.mmm].

clientIdentifier

String

optional

Use this attribute to add your own unique identifying string to a payment call or COS record. This attribute is useful for idempotency purposes. 50 characters maximum.

remittanceData

String

optional

A description of the payment request, meaning, the remittance advice. Use this field for any remittance information that doesn't have its own field. 140 characters maximum.

remittanceLocation ▶

Object

optional

Remittance location information

**debtor ▶**Object **required**

Debtor Personal/Business information. Debtor is the individual receiving the Request for Payment. Please note: For the debtor, name and one additional element like [birthCity, birthCountry, birthdate], [orgLegalEntityId/LEI], OR [addressStreetName, addressCity, addressState, addressPostalCode, and addressCountryare] required as per RfP business rules. [edit](#)

ultimateDebtor ▶

Object optional

Ultimate Debtor Business/Individual Information. The final party receiving the RfP and responsible for initiating the credit transfer upon authorization.

creditor ▶

Object optional

Creditor Business/Individual Information. The recipient of the payment, typically the party at CRB initiating the RfP. Depending on the configuration, this information may be populated automatically.

ultimateCreditor ▶

Object optional

Ultimate Creditor Business/Individual Information. The final recipient of the payment. Depending on the configuration, this information may be populated automatically.

useCaseString **required**

Use one of the following values for the useCase field:

- BusinessToBusiness
- ConsumerBillPay
- AccountToAccount
- DownPaymentOrFinalPayment
- ConsumerToGovernment

industryCategory

String optional

Use one of the following values for the industryCategory field (optional when useCase is BusinessToBusiness or ConsumerToGovernment):

- HomeUtilities
- TelecomUtilities
- Mortgage
- Rent
- InsurancePremiums
- AutoLoans
- PersonalLoans
- CreditCardPayment
- MediaSubscriptions
- Memberships
- FinancialInstitution
- BrokerOrDealer
- Medical
- OnlineGaming
- RealEstate
- Government

**senderID**String **required**

Use the senderID field (optional when useCase is BusinessToBusiness or ConsumerToGovernment), provided by Cross River and generated by The Clearing House (TCH).

Valid senderID's for Sandbox Testing:

-1000000000001
-1000000000002
-1000000000003
-1000000000004
-1000000000005

amountModAllowedBoolean **optional**

Indicates whether the recipient is allowed to modify the requested amount in response to an RfP. If the field is not provided, the system defaults to true.

If true, the recipient may respond with a different amount

If false, the recipient must match the original amount; any mismatch will cause the payment to be rejected.



Curl

```
curl --location 'https://sandbox.crbcos.com/rtp/v1/payments/payment-request' \
--header 'Content-Type: application/json' \
--data '{
  "accountNumber": "2564791479",
  "amount": 20000,
  "discount": 1000,
  "debtor": {
    "routingNumber": "071212128",
    "accountNumber": "2758463448",
    "name": "Terry Tester",
    "birthCity": "Pine Brook",
    "birthCountry": "US",
    "birthDate": "1996-08-09",
    "addressStreetName": "Linwood Ave",
    "addressBuildingNumber": "2115",
    "addressCity": "Pine Brook",
    "addressState": "NJ",
    "addressPostalCode": "07024",
    "addressCountry": "US"
  },
  "creditor": {
    "accountNumber": "2564791479",
    "name": "the creditor"
  },
  "purpose": "Test Payment 2",
  "clientIdentifier": "123456",
  "requestedExecutionDate": "2025-08-10",
  "expiryDate": "2025-08-16",
  "senderId": "100000001",
  "useCase": "BusinessToBusiness",
  "industryCategory": "OnlineGaming"
}'
```



Responses

200



```
{
  "id": "8ee74810-45fe-47ee-9569-b30500b8d6ae",
  "originalPaymentId": "8ee74810-45fe-47ee-9569-b30500b8d6ae",
  "referenceId": "R1740I808VYXJ",
  "accountNumber": "2955057589",
  "clientIdentifier": "123456",
  "amount": 19000,
  "direction": "Outbound",
  "status": "Created",
  "paymentType": "RequestForPayment",
  "source": "Api",
  "transactionAccountContext": "NotSubmitted",
  "debtor": {
    "routingNumber": "011000138",
    "accountNumber": "2758463448",
    "name": "Terry Tester",
    "birthCity": "Pine Brook",
    "birthCountry": "US",
    "birthDate": "1996-08-09",
    "addressStreetName": "Linwood Ave",
    "addressBuildingNumber": "2115",
    "addressCity": "Pine Brook",
    "addressState": "NJ",
    "addressPostalCode": "07024",
    "addressCountry": "US"
  },
  "creditor": {
    "routingNumber": "021214891",
    "accountNumber": "2955057589",
    "name": "Dim Mak",
    "addressStreetName": "400 Kelby St",
    "addressCity": "Fort Lee",
    "addressState": "NJ",
    "addressPostalCode": "07024",
    "addressCountry": "US"
  },
  "network": {
    "clearingSystemCode": "TCH",
    "currency": "USD",
    "endToEndId": "6909c29416f0453fbdfdb30500b8d6ae",
    "instructionId": "20250623021214273T1B19TS07136537534",
    "interbankSettlementAmount": 19000.
  }
}
```



9.1. Untitled



9.2. Test

klnklk

asdasdasda



9.3. Le texte alternatif de l'image Ashbyk disparaît

Découvrez comment accélérer l'évaluation des candidats en utilisant l'IA pour évaluer les postulants selon des critères personnalisés. Cette fonctionnalité aide votre équipe à identifier plus rapidement les meilleurs talents en automatisant le processus initial de présélection des CV directement dans Ashby.

Permissions et accès

- **Forfaits** : Disponible sur Foundations, Legacy Plus, Plus et Enterprise.
- **Permissions** : Vous devez être Org Admin pour activer la fonctionnalité. Pour modifier les critères au niveau du poste, vous devez disposer de Elevated Access - Admin ou Elevated Access - Admin Private pour ce rôle précis.
- Il s'agit d'une **fonctionnalité à activer** qui doit être activée. Vous pouvez l'activer en accédant à **Admin > Configuration de l'organisation > Fonctionnalités optionnelles** et en activant **Examen des candidatures assisté par l'IA**.

Étape 1 : Activer l'évaluation des candidatures assistée par IA

1. Accédez à **Admin > Configuration de l'organisation > Fonctionnalités d'adhésion**.
2. Si **Activer les fonctionnalités d'IA** n'est pas activé, activez-le maintenant. Vous devrez accepter les conditions d'Ashby pour les fonctionnalités IA afin de continuer.
3. Ensuite, activez **Examen des candidatures assisté par l'IA** dans la liste des tuiles de fonctionnalités IA disponibles.

Étape 2 : Définir vos critères d'évaluation

Les critères que vous ajoutez au poste définissent ce que l'IA évalue. Vous pouvez définir des critères de CV que l'IA recherchera dans le CV d'un candidat ainsi que vérifier si le candidat a répondu à des questions précises du formulaire de candidature.

Pour définir vos critères, accédez à **Emplois > Emplois** et sélectionnez un poste. Dans les paramètres du poste, accédez à l'onglet **Fonctionnalités d'IA** et sélectionnez **Définir des critères**.

Critères de CV

Dans la section Resume criteria, vous pouvez ajouter des critères de plusieurs façons : ajouter des critères suggérés par l'IA selon l'offre d'emploi, ajouter des critères utilisés précédemment pour d'autres postes, ou ajouter des critères personnalisés que vous créez spécifiquement pour ce rôle.

Pour commencer à ajouter des critères, sélectionnez **+ Ajouter des critères** sous la section **Critères du CV**.



Group Product Manager

AI-Assisted Application Review Criteria

Learn more ↗

Resume criteria

No criteria have been set for this job yet.

+ Add criteria ▾

✦ Add from 6 suggestions

🕒 Add previously used

📝 Add custom

Application form criteria

Add a question of type *Long Unformatted Answer* to the job's application form to create criteria based on applicant answers.

Ajouter des critères suggérés ou des critères déjà utilisés

Sélectionnez **Ajouter à partir de # suggestions** pour obtenir des suggestions générées par l'IA à partir de votre offre d'emploi. Vous pouvez également réutiliser des critères d'autres rôles auxquels vous avez accès en sélectionnant **Ajouter des éléments déjà utilisés**.

Pour ajouter une suggestion, sélectionnez **Ajouter** dans la colonne **Action**.



Search for relevant criteria based on prompts, titles, or jobs...

All

Suggestions 6

Previously Used

Auto-Generated Suggestion	Source	Action
Product Management Experience Candidate has 3–5 years of experience in a product management role with titles such as Product Manager, Senior Product Manager, or Group Product Manager, as shown on the...	Job Posting	Add
Led Product Initiatives Has led at least one product launch or feature roadmap with documented outcomes or scope described in the resume.	Job Posting	Add
Hiring Experience Resume lists specific activities related to recruiting or hiring product team members, such as writing job postings or conducting interviews.	Job Posting	Add
People Management or Mentoring Has held a position with direct reports or formal mentorship responsibilities as indicated in job titles or descriptions.	Job Posting	Add
Authored Job Descriptions Resume includes examples of having written job descriptions or hiring requisitions for product or related roles.	Job Posting	Add
Cross-functional Collaboration Experience leading cross-functional meetings or representing products in stakeholder forums, as described on the resume	Job Posting	Add

Créer des critères personnalisés efficaces

Sélectionner **Ajouter personnalisé** au lieu des options suggérées ou déjà utilisées vous permet de créer vos propres critères. Vous devrez définir les éléments suivants :

- Titre court :** Le nom de votre critère. Celui-ci apparaît dans les paramètres du poste, sur la page d'évaluation des candidatures et comme nom de filtre.
- Invite :** Les instructions utilisées pour évaluer les candidats. Des invites précises et claires produisent de meilleures performances de l'IA.

Conseils sur l'efficacité des critères

Rédigez vos critères comme si vous donniez des instructions à un recruteur junior. Assurez-vous que l'invite est formulée de manière à ce que tout lecteur arrive à la même conclusion.



- **Autonome** : Les critères doivent être compréhensibles sans contexte supplémentaire. Au lieu de « Peut assimiler les commentaires », utilisez « Intègre efficacement les commentaires liés au rendement professionnel personnel ».
- **Unique** : Concentrez-vous sur une compétence par exigence. Au lieu de « Expérience en Python et AWS », créez deux critères distincts pour chacun.
- **Clair** : Évitez les termes vagues comme « formation technique ». Précisez l'expertise exacte requise.
- **Spécifique** : Utilisez des exigences mesurables, telles que le nombre d'années d'expérience ou des intitulés de poste précédents précis.
- **Objectivement vérifiable** : Assurez-vous que l'exigence peut être démontrée en consultant un CV. Au lieu de « a bon goût », utilisez « a dirigé le design pour des produits SaaS destinés aux consommateurs ».

Critères du formulaire de candidature

Vous pouvez évaluer si un candidat a répondu aux questions longues du formulaire de candidature dans la section **Critères du formulaire de demande**.

Activez l'option **La demande a-t-elle répondu à la question ?** pour des questions individuelles. Celles-ci sont ajoutées comme critères dans le processus d'évaluation.

Cette vérification détermine uniquement si la question a reçu une réponse. Les réponses vides ou sans lien avec la question sont signalées comme ne répondant pas aux critères. Cette fonctionnalité est uniquement disponible pour les champs de texte longs non formatés.

Étape 3 : Validation et révision des critères

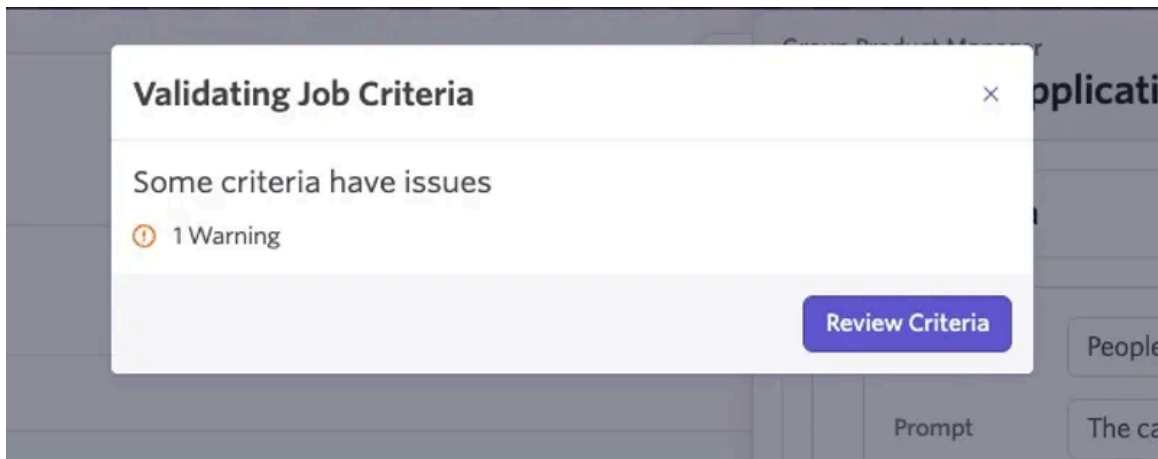
Une fois vos critères configurés, ils doivent être validés. Ashby vérifie vos invites sur deux dimensions :

- **Clarté et objectivité** pour s'assurer que les critères sont clairs et objectifs, ce qui signifie qu'ils peuvent être évalués adéquatement par l'IA.
- **Violations de l'égalité des chances en matière d'emploi (EEO)** : l'IA recherche également les violations EEO implicites ou explicites et les signale.

Validation des critères

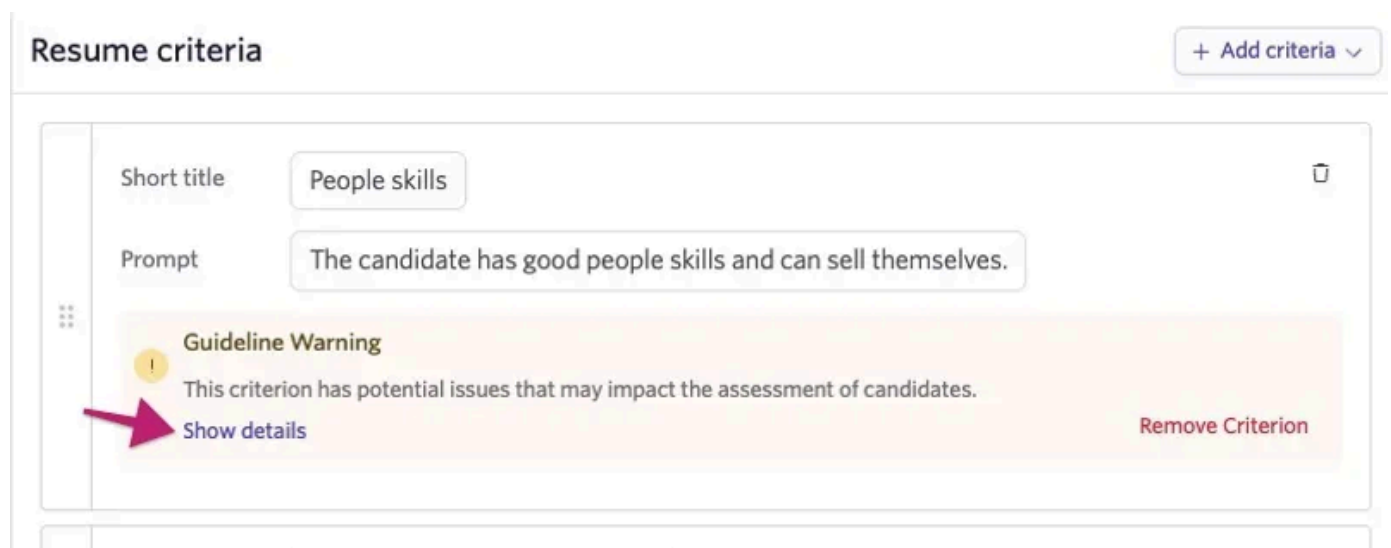
Pour valider vos critères, sélectionnez **Valider**. Si tous les critères passent la validation, vous verrez une fenêtre contextuelle l'indiquant. Cliquez sur **Fermer** et passez à la section suivante sur l'enregistrement des critères et l'évaluation des candidats.

Si le système identifie des problèmes, sélectionnez **Critères d'évaluation** dans la fenêtre contextuelle pour afficher les avertissements.



Révision des critères signalés

Les critères signalés seront identifiés par un avertissement. Sélectionnez **Afficher les détails** sous n'importe quel avertissement pour comprendre pourquoi le critère a été signalé.



Nous vous recommandons soit de mettre à jour le critère jusqu'à ce qu'il ne soit plus signalé, soit de supprimer complètement le critère avant de continuer.

Vous pouvez continuer avec des critères signalés, mais cela n'est pas recommandé.

Étape 4 : Enregistrement des critères et évaluation des candidats

Une fois la validation réussie, sélectionnez **Enregistrer et évaluer tous les candidats actifs** pour commencer l'évaluation initiale. Selon le nombre de candidats, cela peut prendre de quelques secondes à quelques minutes.

Vous verrez un écran de chargement affichant le nombre total de candidats en cours d'évaluation ainsi qu'une barre de progression.



AI-Assisted Application Review		0 / 6	Edit
Cross-functional Collaboration	Experience leading cross-functional meetings or representing products in stakeholder forums, as described on the resume		
Led Product Initiatives	Has led at least one product launch or feature roadmap with documented outcomes or scope described in the resume.		
Product Management Experience	Candidate has 3-5 years of experience in a product management role with titles such as Product Manager, Senior Product Manager, or Group Product Manager, as shown on the resume with dates/tenure.		

Si vous mettez à jour des critères existants, sélectionnez **Enregistrer et réévaluer toutes les considérations d'emploi actives** une fois vos mises à jour effectuées.

Examiner les évaluations des candidats

L'IA d'Ashby évalue les CV des candidats par rapport à chaque critère spécifique et renvoie un statut d'évaluation. Le statut peut indiquer que le candidat répond au critère, n'y répond pas ou que l'IA n'a pas pu se prononcer sur le critère. Les résultats de l'évaluation peuvent être consultés dans le profil du candidat ainsi que dans l'étape Application Review du pipeline de candidats pour le poste.

Afficher les résultats dans le profil du candidat

Affichez les évaluations complètes dans l'onglet **Commentaires** d'un profil de candidat.

Feed | Notes | **Feedback** | Emails | Texts | Referrals | Forms | Assistant

AI GENERATED

Criteria Evaluation

Meets 1 of 3

Cross-functional Collaboration

Led Product Initiatives

Product Management Experience

?

✓

✗

>

>

>

Show Less

Cliquez sur n'importe quel critère pour voir le raisonnement associé à un statut « répond au critère », « ne répond pas au critère » ou « indéci ». Sélectionnez l'icône permettant de tout développer pour voir le raisonnement de tous les critères à la fois.



Feed | Notes | Feedback | Emails | Texts | Referrals | Forms | Assistant

AI GENERATED

Criteria Evaluation

Meets 1 of 3

Cross-functional Collaboration

“Experience leading cross-functional meetings or representing products in stakeholder forums, as described on the resume”

The candidate has experience leading a product team of 11 product managers and has established a product analytics framework, which suggests they have likely led cross-functional meetings and represented products in stakeholder forums. However, the resume does not explicitly mention specific instances of leading such meetings or forums.

Led Product Initiatives

“Has led at least one product launch or feature roadmap with documented outcomes or scope described in the resume.”

The candidate has successfully launched 5 major products while serving as a Group Product Manager, generating significant annual revenue and demonstrating documented outcomes related to product launches. This aligns well with the criterion of having led a product launch with documented outcomes.

Product Management Experience

“Candidate has 3-5 years of experience in a product management role with titles such as Product Manager, Senior Product Manager, or Group Product Manager, as shown on the resume with dates/tenure.”

Résultats d'évaluation dans l'Application Review

L'étape Application Review du pipeline affiche une colonne pour les résultats d'évaluation qui fournit un résumé de la façon dont le candidat a été évalué. Survolez la ligne des résultats pour voir plus de détails.

Leads | Application Review | Active | Pending Offer | Hired | Archived

Filters

Search by candidate name...

Review 2 Applications

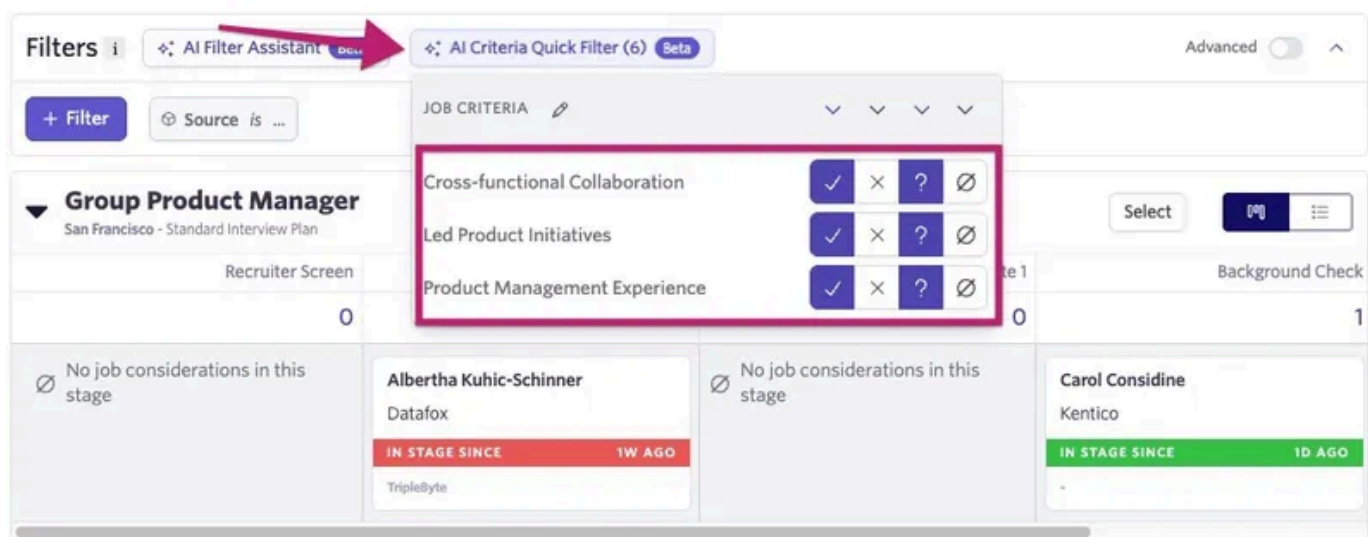
Name / ...	Status	Job Title	Source	Submitted ...	Avg. Sc...	Evaluations	Ashby Job Posting ID
Alex Ball...	In Stage...	Group Product Ma...	TripleByte	Apr 03, 2026, 5:0...	-	Cross-functional Collaboration +2	-
Minnie B...	In Stage...	Group Product Ma...	Hired.com	Mar 21, 2026, 5:0...	-	Cross-functional Collaboration +2	-

Filtrage par critères d'IA



Vous pouvez filtrer les candidats selon leur statut d'évaluation dans les étapes de votre pipeline :

1. Sélectionnez l'option **Filtre rapide des critères d'IA**.
2. Sélectionnez les critères et statuts que vous souhaitez afficher, comme tout candidat ayant réussi ou pour lequel l'IA est restée indécise sur un critère particulier.



3. Effectuez des actions groupées sur la liste filtrée selon vos besoins.

Avis juridique sur le traitement automatisé

Vous pouvez configurer un message à afficher aux candidats susceptibles d'être affectés par des processus de révision automatisés par l'IA en suivant les étapes ci-dessous.

1. Accédez à **Admin > Gestion des données > Confidentialité des données et conformité > Confidentialité et affaires juridiques** pour configurer un avis juridique destiné aux candidats.
2. Repérez la section **Avis juridique sur le traitement automatisé** et cliquez sur **Mentions légales pour tous les emplois** affiché.
3. Configurez les paramètres indiquant quand ce message sera affiché :
 - **Est activé?:** Lorsqu'elle est désactivée, les candidats ne voient pas l'avis.
 - **Portée de la règle :** Définissez ici des critères lorsque vous souhaitez que le message s'affiche uniquement selon certains critères, par exemple pour certains postes seulement. N'ajoutez pas de portée si vous voulez qu'il s'applique à tous les postes.
 - **Masquer les scores :** Si cette option est activée, les scores générés par l'IA sont masqués aux utilisateurs si le candidat n'a pas vu l'avis.
 - **Texte de l'avis juridique :** Le texte précis affiché aux candidats. Celui-ci peut être modifié.
4. Cliquez sur **Enregistrer** une fois que vous avez ajusté les paramètres selon vos besoins.

Expérience du candidat

Les candidats voient un lien **En savoir plus** sous le bouton **Soumettre la candidature**. En cliquant dessus, ils peuvent consulter l'avis, refuser le traitement automatisé ou vérifier s'ils l'ont déjà refusé. Vous pouvez voir à quoi ressemble cette page en sélectionnant



Vue dans le coin supérieur droit lors de la configuration du message.

Crédits IA

AI-Assisted Application Review est facturé par crédit. Un crédit équivaut à une évaluation d'un seul candidat (considération d'emploi), avec prise en charge d'un maximum de 50 critères par évaluation.

Des crédits supplémentaires peuvent être achetés par blocs. Pour plus d'informations, consultez notre guide sur .

FAQ

Si je modifie mes critères, puis-je réévaluer les candidats ?

Oui. Si vous modifiez une invite ou changez des critères, sélectionnez l'option Save and Re-Evaluate all active Job Considerations pour actualiser les évaluations.

Puis-je signaler des évaluations IA si elles sont incorrectes ?

Oui. Sélectionnez les trois points dans le coin supérieur droit d'une évaluation pour la signaler comme incorrecte ou trompeuse.

Puis-je ajouter des évaluations comme colonne dans Application Review ?

Oui. Sélectionnez **Ajouter une colonne** sur la page de revue des candidatures et choisissez **Considération d'emploi > Critères > Évaluations**.

Quelles autorisations sont nécessaires pour modifier les critères d'emploi ?

Vous devez disposer de Elevated Access - Admin ou Elevated Access - Admin Private pour le rôle spécifique, ou être un administrateur de l'organisation.

L'IA évalue-t-elle la formation et les antécédents professionnels s'il n'y a pas de CV ?

Oui, si l'information figure dans le profil Ashby du candidat, l'IA en tient compte.



9.3.1. Untitled

Get Cakes

Get a cake by its ID

GET

https://api.cakes.com

Request

PATH PARAMETERS

Parameter name	String	optional
----------------	--------	----------

BODY PARAMETERS

id	String	required
ID of the cake to get		

JS JS

```
var myHeaders = new Headers();
myHeaders.append("Accept", "application/json");
myHeaders.append("Content-Type", "application/json");

var raw = JSON.stringify({
  "id": "string"
});

var requestOptions = {
  method: 'GET',
  headers: myHeaders,
  body: raw,
  redirect: 'follow'
};

fetch("https://api.cakes.com", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

Responses

200

```
{
  "name": "Cake's name",
}
```




10. c'est un titre h1

klmklm



Endpoint: /core/v1/cm/customers

POST<https://sandbox.crbcos.com/core/v1/cm/customers>**Request**

Response

BODY PARAMETERS

partnerId

String

required

Your unique ID in the Cross River system. This ID is in GUID format.

name ▶

Object

required

Customer name in detail

classification

String

required

The classification type of the account:
Use Personal (meaning, an individual person's account)

profile ▶

Object

required

Customer banking-relevant information

dueDiligence ▶

Object

optional

Details for due diligence compliance for an individual customer. Speak to your compliance liaison to find out any requirements for Due Diligence fields.

clientIdIdentifier

String

optional

Use this attribute to add your own unique identifying string to a payment call or COS record. This attribute is useful for idempotency purposes.
50 characters maximum

primaryAddress ▶

Object

optional

Main address for the customer. This can be added later. When you use this object, some attributes are required (as indicated).

primaryPhone ▶

Object

optional

Main phone number for the customer. This can be added later. When you use this object, some attributes are required (as indicated).

primaryEmail ▶

Object

optional

Main email address for the customer. This can be added later. When you use this object, some attributes are required (as indicated).



primaryIdentification ▶

Object

optional

Identification metadata of customer-identifying documents, such as a driver's license or passport. This can be added later. When you use this object, some attributes are required (as indicated).



 Curl



```
curl --location 'https://sandbox.crbcos.com/core/v1/cm/customers' \
--header 'Content-Type: application/json' \
--data-raw '{
  "partnerId": "1101a33d-fbd9-4b9e-b169-b29500f12407",
  "name": {
    "firstName": "Brandon",
    "lastName": "Sanderson"
  },
  "classification": "Personal",
  "profile": {
    "reg0": false,
    "citizenshipCountryCode": "US",
    "politicallyExposedPerson": false,
    "enableBackupWithholding": false,
    "backupWithholdingPercent": 0,
    "taxResidencyCountryCode": "US"
  }
}
```



Responses

200



```
{
  "id": "6c545b91-f401-4308-bad3-b300007b7885",
  "cifNumber": "245085933838",
  "classification": "Personal",
  "status": "Active",
  "ofac": "Pending",
  "pepScan": "Pending",
  "internalList": "Pending",
  "name": {
    "firstName": "Brandon",
    "lastName": "Sanderson",
    "fullName": "Brandon Sanderson"
  },
  "profile": {
    "reg0": false,
    "citizenshipCountryCode": "US",
    "politicallyExposedPerson": false,
    "enableBackupWithholding": false,
    "backupWithholdingPercent": 0,
    "taxIdType": "Ssn",
    "taxId": "312860737",
    "birthDate": "1980-03-06",
    "riskRating": "Low",
    "privacyOptOut": true
  },
  "createdAt": "2025-06-18T03:29:32.6027543-04:00",
  "lastModifiedAt": "2025-06-18T03:29:32.6027543-04:00",
  "partnerId": "1101a33d-fbd9-4b9e-b169-b29500f12407",
  "dueDiligence": { asdasd asdsad
    "employmentStatus": "Employed",
    "occupation": "Technology",
    "occupationDescription": "Support",
    "industry": "Finance",
    "incomeSource": "Salary",
    "incomeState": "NJ",
    "incomeCountryCode": "US",
    "employer": "Cross River",
    "employerState": "NJ",
    "employerCountryCode": "US",
    "sourceOfFunds": "Salary",
    "wealthSource": "Salary",
    "wealthSourceDescription": "Salary",
    "annualIncome": 4500000
  }
}
```



11.1. Document de test

Nabuchodonosor

asdasdas

1.0.7 OAS 2.0

Swagger Petstore

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on irc.freenode.net, #swagger. For this sample, you can use the api key **special-key** to test the authorization filters.

- [Terms of service](#)
- [Contact the developer](#)
- [Apache 2.0](#)
- [Find out more about Swagger](#)

Schemes

HTTPS

Authorize

pet Everything about your Pets

Find out more ^

POST

/pet/{petId}/uploadImage uploads an image

POST

/pet Add a new pet to the store

PUT

/pet Update an existing pet

GET

/pet/findByStatus Finds Pets by status



GET /pet/findByTags Finds Pets by tags

Warning: Deprecated

GET /pet/{petId} Find pet by ID

POST /pet/{petId} Updates a pet in the store with form data

DELETE /pet/{petId} Deletes a pet

store Access to Petstore orders

GET /store/inventory Returns pet inventories by status



POST

/store/order Place an order for a pet

^

GET

/store/order/{orderId} Find purchase order by ID

^

DELETE

/store/order/{orderId} Delete purchase order by ID

^

user Operations about user

Find out more about our store ^

POST

/user/createWithList Creates list of users with given input array

^

GET

/user/{username} Get user by user name

^



PUT	/user/{username}	Updated user	⌵
↻			
DELETE	/user/{username}	Delete user	⌵
↻			
GET	/user/login	Logs user into the system	⌵
↻			
GET	/user/logout	Logs out current logged in user session	⌵
↻			
POST	/user/createWithArray	Creates list of users with given input array	⌵
↻			
POST	/user	Create user	⌵
↻			



Models



```
ApiResponse {
  code      integer($int32)
  type      string
  message   string
}
```

```
Category {
  id        integer($int64)
  name      string
}
```

```
Pet {
  id          integer($int64)
  category    Category {
    id        integer($int64)
    name      string
  }
  name*       string
  photoUrls*  [
    string
    example: doggie
    xml: { "name": "photoUrl" }
    xml: {
      name: photoUrl
    }
  ]
  tags        [
    Tag {
      id        integer($int64)
      name      string
    }
  ]
  status      string
  pet status in the store
  Enum:
    [ available, pending, sold ]
}
```

```
Tag {
  id        integer($int64)
  name      string
}
```



```
Order {
  id              integer($int64)
  petId           integer($int64)
  quantity        integer($int32)
  shipDate        string($date-time)
  status          string
  Order Status
  Enum:
  [ placed, approved, delivered ]
  complete        boolean
}
```

```
User {
  id              integer($int64)
  username         string
  firstName        string
  lastName         string
  email           string
  password        string
  phone           string
  userStatus       integer($int32)
  User Status
}
```

Bucarest

Lorem Ipsum : est simplement un texte factice de l'industrie de l'impression et de la composition typographique. Lorem Ipsum est le texte factice standard de l'industrie depuis les années 1500, lorsqu'un imprimeur inconnu prit une galée de caractères et la mélangea pour créer un livre spécimen de polices. Il a survécu non seulement à cinq siècles, mais aussi au passage à la composition électronique, demeurant essentiellement inchangé. Il a été popularisé dans les années 1960 avec la publication de feuilles Letraset contenant des passages de Lorem Ipsum, et plus récemment avec des logiciels de publication assistée par ordinateur comme Aldus PageMaker incluant des versions de Lorem Ipsum.



12. Retrait

12.1. Une autre page

asdasdsadasdas



12.1.1. Filtrage avancé

Dans la section « faceting », nous mettons en évidence comment il est possible d'obtenir les filtres disponibles pour une requête de recherche et d'appliquer des filtres pour affiner les résultats de recherche. Par exemple, après avoir recherché « shoes », un acheteur peut vouloir affiner les résultats en appliquant un filtre sur l'attribut de couleur et en choisissant une valeur telle que rouge. Cela affichera uniquement les « red » shoes dans les résultats. Si à ce stade, l'acheteur choisit encore une autre couleur, par exemple bleu, Klevu affichera maintenant à la fois les « red » et les « blue » shoes. Ainsi, il appliquera l'opération OR entre les valeurs du même attribut. Si à cette étape, l'acheteur choisit « cotton » comme matériau, le système affichera à la fois les chaussures rouges et bleues, mais il s'assurera également que tous les produits ont « cotton » comme matériau. En d'autres termes, il applique l'opération AND parmi les différents types d'attributs sélectionnés.

Cependant, il arrive un moment où l'on peut vouloir effectuer des opérations de filtrage complexes, ou en d'autres termes, des opérations de filtrage avancées. Par exemple, vous pouvez vouloir afficher un groupe de chaussures où la « color is red » et où la « brand is NOT Adidas ». Vous pouvez également vouloir ajouter un autre groupe de chaussures à cet ensemble de résultats. Par exemple, où la « color is blue or black », la « brand is Nike » et où la « price range is between 100 and 400 ». Comme vous pouvez le constater, contrairement au comportement du « faceting », dans notre exemple ici, nous souhaitons avoir un ou plusieurs groupes de produits correspondant à différents groupes de conditions pour que l'ensemble final de produits soit renvoyé.

C'est dans cette intention de permettre des opérations de filtrage avancées sur l'ensemble de résultats que nous avons introduit une nouvelle fonctionnalité appelée « advanced filtering » dans notre API.

Avant l'introduction du filtrage avancé, si un utilisateur demandait que des filtres soient renvoyés ou que certains soient appliqués, Klevu déterminait auparavant un ensemble de produits pertinents correspondant à la requête de l'utilisateur. Il incluait/excluait ensuite tous les produits explicitement demandés dans cet ensemble de résultats. Ce n'est qu'après cette étape qu'il calculait tous les filtres possibles à renvoyer et appliquait également les filtres sélectionnés pour affiner les résultats de recherche. Ainsi, si un produit ajouté explicitement ne correspondait pas au filtre appliqué, il était exclu de l'ensemble final des résultats.

Avec l'ajout du filtrage avancé, Klevu identifie maintenant un ensemble de produits pertinents correspondant à la requête. Il applique ensuite le filtrage avancé à celui-ci. Ce n'est qu'après cette étape qu'il inclut/exclut les produits explicitement demandés, calcule les filtres en fonction de cet ensemble de résultats et applique finalement les filtres sélectionnés pour affiner les résultats de recherche. De cette façon, si un filtrage doit être appliqué avant l'inclusion/exclusion explicite de produits, le filtrage avancé doit être utilisé.

Veuillez noter que le filtrage avancé peut également être utilisé avec les API de Category Merchandising et Recommendation.

L'objet **condition de groupe** peut être utilisé pour appliquer les conditions de filtrage avancé afin que vos clients puissent affiner leurs résultats en fonction d'attributs pertinents.



Apply Advanced Filtering

Example is as below:

POST

https://{hostname}/cs/v2/search

Request

BODY PARAMETERS

groupOperator

String

optional

Defining the condition of filtering. Here are the options available.

ALL_OF: This is used to filter data using AND query i.e. the search results are matching with all the attributes.

ANY_OF: This is used to filter data using OR query i.e. the search results are matching with at least one attributes.

NONE_OF: This is used as negation AND query i.e. when search results are matching with none of the attribute values.

key

String

optional

The ID of the attribute to filter by, eg. color

valueOperator

String

optional

Defining an attribute's operation, Here are the options available.

INCLUDE : This is used to include specific attribute for the filter purpose.

EXCLUDE : This is used to exclude specific attribute for the filter purpose.

EXISTS: This is used to find if the specific indexed attribute value exists with product.

NOT_EXISTS : This is used to find if the specific indexed attribute value does not exist at all.

singleSelect

Boolean

optional

The behaviour when specifying multiple filters or values.

If it is set to "true", the products returned must have both the values.

If it is set to "false", the products returned must have at least one of the values.

excludeValuesInResult

Boolean

optional

For numeric filter like price, can be used to exclude data of the starting or ending values. For example if the range is defined as "1200 - 1232" in values, and excludeValuesInResult is set to true, Klevu will use range as "1201 - 1231".

This can be used to achieve greater than or less than condition. For example, if you want to get the products greater than price 200, the values field will be "200 - *" and the excludeValuesInResult will be set to true.

**values**

Array

optional

An array of values to filter by, eg. Red, Blue. When using range filters, specify the first value as the minimum value and the second as the max value. For example to retrieve records with prices between 60 to 80, use: "key": "klevu_price", "values": [60, 80].

To retrieve steps of prices, for example those with values between 0-50 or 150-200, use: "key": "klevu_price", "values": ["0-50", "150-200"].

To retrieve exact values, for example records with exact values 100 or 200, use: "key": "klevu_price", "values": ["100-100", "200-200"].

By default all attributes submitted to Klevu are indexed as STRING attributes, which means they cannot be used as range filters. The product sale price field is the only exception to this rule, which is filtered using the key klevu_price. If you have explicitly requested and Klevu has approved that certain attributes be indexed as numerical attributes, you can also retrieve those as range filters.



JS Request Body (JSON)



```
{
  "context": {
```



Responses

200



```
{
  "meta": {
    "qTime": 9,
    "responseCode": 200
  },
  "queryResults": [
    {
      "id": "productSearch",
      "meta": {
        "qTime": 7,
        "noOfResults": 2,
        "totalResultsFound": 6,
        "typeOfSearch": "WILDCARD_AND",
        "offset": 0,
        "debuggingInformation": {},
        "notificationCode": 1,
        "searchedTerm": "tee",
        "apiKey": "klevu-156925593843210765",
        "isPersonalised": false
      },
      "records": [
        {
          "discount": "",
          "hideGroupPrices": "",
          "type": "Tees",
          "itemGroupId": "4384027344958",
          "freeShipping": "",
          "storeBaseCurrency": "GBP",
          "price": "24.00",
          "toPrice": "",
          "imageUrl": "https://cdn.shopify.com/s/files/1/0116/9457/1582/products/ws05-yellow_main_fe09e105-6b7e-4269-9f",
          "currency": "GBP",
          "inStock": "yes",
          "id": "31366447038526",
          "imageHover": "",
          "sku": "WS05-XS-Yellow",
          "brand": "KKE",
          "basePrice": "24.0",

```



▼ CodeSandbox

try_out_apis

T Team Klevu

487.8k 0 21

Edit Sandbox

▼ Files

.codesandbox

public

src

App.js

index.js

payload.js

payloadList.js

styles.css

package.json

▼ Dependencies

< > ↺ https://4ktfrp.csb.app/?payload=advanceFiltering

❖ 📄

Open Sandbox

Console 0

Problems 0

React DevTools 0

^



12.2. Nouvelle page

asdasdasdas



13. page de test Ocient

Le prend en charge les requêtes utilisant la syntaxe SQL conforme aux normes ANSI SQL.

Syntaxe SQL Ocient

@virtual

Ce bloc de code montre la syntaxe générale pour effectuer des requêtes SQL dans ainsi que l'ordre dans lequel les commandes doivent être placées.

Pour obtenir des descriptions et la syntaxe spécifiques des commandes, consultez les sections respectives des instructions SQL sur cette page.

Syntaxe

SQL

```
[ WITH ... ]
SELECT ...
    [ EXCEPT(...) ]
[ FROM ...
    [ JOIN ... ] ]
[ WHERE ... ]
[ GROUP BY ...
    [ HAVING ... ] ]
[ ORDER BY ... ]
[ LIMIT ... ]
[ OFFSET ... ]
[ INTERSECT ... ]
[ EXCEPT ... ]
[ UNION ... ] ]
[ USING ... ]
[ TRACE ... ]
[ TAG ... ]
```

Schéma par défaut

Ocient identifie chaque table à l'aide d'une base de données et d'un schéma. Par exemple, le chemin pleinement qualifié vers la table `films` est `cinema.adventure.movies`, où `cinéma` est la base de données et `aventure` est le schéma.

Lorsque vous ne qualifiez pas entièrement un nom de table, le système Ocient utilise un schéma par défaut. Lorsque vous ouvrez une session dans le système pour la première fois, le schéma par défaut correspond à votre nom d'utilisateur pleinement qualifié. Vous pouvez modifier le schéma par défaut à l'aide de la commande [DÉFINIR LE SCHÉMA](#).

Référence des instructions de requête SQL

Ocient prend en charge les instructions SQL suivantes.

WITH



Attribue un nom à une expression de table commune, permettant à une requête auxiliaire d'être utilisée dans la requête principale. Cela est utile pour décomposer des requêtes complexes en parties plus petites.

Syntaxe

SQL

```
WITH
  <cte_name> [ ( <cte_column> [ ,... ] ) ]
  AS ( <sub_query> )
<select_query>
```

Paramètres

Paramètre	Description
<cte_name>	Nom de l'expression de table commune remplie avec l'ensemble de résultats de l'instruction <sub_query> .
<cte_column>	Facultatif. Il s'agit d'une liste de noms de colonnes pour les expressions de table communes remplies avec l'ensemble de résultats <sub_query> . Si aucun nom de colonne n'est fourni, les noms de colonnes sont les mêmes que ceux de la table référencée dans l'instruction <sub_query> .
<sub_query>	Une requête auxiliaire qui collecte des données pour une expression de table commune. <sub_query> peut utiliser des commandes de requête régulières, y compris ORDER BY, LIMIT, OFFSET, UNION, INTERSECT, EXCEPT, WHERE, GROUP BY et HAVING. Consultez les sections de référence respectives des instructions SQL pour connaître les règles d'utilisation.
<select_query>	La requête SELECT principale. Notez que l'instruction DE dans la requête principale SELECT doit également inclure <cte_name> si la requête utilise ses valeurs.

Exemple

Dans cet exemple, la sous-requête calcule le budget moyen pour toutes les lignes de la table movies . La requête principale utilise cette moyenne pour trouver tous les films qui ont dépensé davantage.

SQL

```
WITH avg_budget_table (average_budget) AS (
  SELECT AVG(budget)
  FROM movies
)
SELECT title,
       budget,
       revenue
FROM movies,
     avg_budget_table
WHERE budget > avg_budget_table.average_budget;
```



Résultat

titre	budget	revenu
CGI Why	237000000	2787965087
Titania	200000000	1845034188
Véhicule de marchandise 5	200000000	1066969703
Le Tentpole	220000000	1519557910
Pirates de Palm Springs	140000000	655011224
Espion 16	200000000	1108561013
Glacial	150000000	1274219009
Furie 7	190000000	1506249360
Superhéros 23	250000000	1084939099
Monde Triassique	150000000	1513528810
Chef de fer 3	200000000	1215439994
Le dernier aviateur	150000000	318502923

SELECT

Initie une instruction query statement ou une clause de sous-requête dans d'autres instructions.

Vous pouvez interroger les données des tables pour lesquelles vous disposez du privilège SELECT.

Pour obtenir des informations sur l'utilisation de SELECT comme sous-requête pour filtrer ou trier les résultats, consultez les sections [OÙ](#) et [AYANT](#) .

Pour obtenir des informations sur l'utilisation de SELECT comme sous-requête pour une expression de table commune, consultez la section [AVEC](#) .

Syntaxe

SQL

```
SELECT [ ALL | DISTINCT ]
      [ * [ EXCEPT ( column_name [ , ... ] ) ] ]
      | <select_list_entry> [ , ... ] ]
      [ <from_clause> ]
```

Paramètres



Paramètre	Description
TOUT	<code>SELECT</code> et <code>TOUT SÉLECTIONNER</code> sont identiques. Renvoie toutes les lignes de données valides de la base de données qui répondent aux critères de votre requête.
DISTINCT	<code>SELECT DISTINCT</code> renvoie uniquement des lignes uniques qui ne correspondent pas à d'autres lignes selon les critères de votre requête.
*	Renvoie toutes les colonnes des tables spécifiées dans l'ensemble de résultats de la requête.
SAUF	Lorsqu'il est utilisé avec <code>*</code> , <code>SAUF</code> permet d'exclure des colonnes spécifiques des résultats de la requête.
nom_de_colonne	Nom d'une ou de plusieurs colonnes de la table spécifiée que vous souhaitez exclure (à l'aide de <code>SAUF</code>) des résultats de votre requête.

<select_list_entry>

L'élément `select_list_entry` définit une colonne ou une expression à inclure dans l'ensemble de résultats de votre requête.

Syntaxe

SQL

```
<select_list_entry> ::=  
    column_name | expression [ AS new_name ]
```

Paramètres

Paramètre	Description
nom_de_colonne	Le nom d'une ou de plusieurs colonnes de la table spécifiée que vous souhaitez inclure dans les résultats de votre requête.
expression	Une ou plusieurs expressions que vous souhaitez inclure dans les résultats de votre requête. Les expressions peuvent être n'importe quelle combinaison de valeurs littérales, de noms de colonnes, d'expressions arithmétiques, de parenthèses et d'appels de fonction.
nouveau_nom	Un alias, traité comme un identifiant, pour un nom alternatif de la colonne ou des résultats de l'expression.

<clause_from>

Pour obtenir des informations sur `<from_clause>`, consultez la documentation [DE](#).

Exemples

Utilisation de `SELECT *`

Cet exemple utilise `SELECT *` pour renvoyer toutes les colonnes de la table `movies`.



SQL

```
SELECT * FROM movies LIMIT 5;
```

Sortie

movie_id	title	budget	popularity	release_date	revenue	runtime	movie_status	vote_average	vote_count
211672	Minnows	74000000	875.581305	2015-06-17	1156730962	91	Sorti	6.40	4571
24	Billy the Killy	30000000	79.754966	2003-10-10	180949000	111	Sorti	7.70	4949
19995	CGI Why	237000000	150.43757	2009-12-10	2787965087	162	Publié	7.20	11800
37724	Spyman 16	200000000	93.004993	2012-10-25	1108561013	143	Publié	6.90	7604
24428	The Tentpole	220000000	144.448633	2012-04-25	1519557910	143	Publié	7.40	11776

Utilisation de SELECT * EXCEPT

Cet exemple utilise `SELECT * EXCEPT` pour exclure certaines colonnes du jeu de résultats.

SQL

```
SELECT *  
EXCEPT (movie_id, runtime, vote_average, vote_count)  
FROM movies  
LIMIT 5;
```

Résultat

title	budget	release_date	revenue
Swords & Scabbards	94000000	2003-12-01	1118888979
Billy the Killy	30000000	2003-10-10	180949000
Merchandise Vehicle 5	200000000	2010-06-16	1066969703
CGI Why	237000000	2009-12-10	2787965087
Space Odyssey 6000	10500000	1968-04-10	68700000

Utilisation des alias de colonnes latéraux

Les requêtes Ocient SQL prennent en charge les alias de colonnes latéraux, ce qui signifie que vous pouvez réutiliser immédiatement des alias pour des calculs dans la même requête comme nouvelles entrées. Par conséquent, vous pouvez simplifier les requêtes qui



nécessitent normalement des sous-requêtes et des expressions de table communes.

Ces exemples utilisent la table `produits` avec ces colonnes :

- `product_id` — Identifiant du produit sous forme d'entier
- `product_name` — Nom du produit sous forme de chaîne
- `prix` — Prix sous forme de nombre à virgule flottante

Créez cette table à l'aide de l'instruction SQL `CRÉER UNE TABLE` .

SQL

```
CREATE TABLE products (  
  product_id INT,  
  product_name VARCHAR(100),  
  price DECIMAL(10, 2)  
);
```

Insérez quatre enregistrements dans la table `produits` .

SQL

```
INSERT INTO products (product_id, product_name, price) VALUES  
(1, 'Laptop', 1000.00),  
(2, 'Tablet', 500.00),  
(3, 'Smartphone', 800.00),  
(4, 'Monitor', 300.00);
```

Créez une requête pour déterminer les prix après réductions et taxes en utilisant une sous-requête d'expression de table commune

`Prix réduits` . Utilisez le mot-clé `AVEC` pour créer la sous-requête.

SQL

```
WITH DiscountedPrices AS (  
  SELECT  
    product_id,  
    product_name,  
    price,  
    price * 0.90 AS discounted_price,  
    price * 0.90 * 1.05 AS total_price_after_tax  
  FROM  
    products  
)  
SELECT  
  product_id,  
  product_name,  
  price,  
  discounted_price,  
  total_price_after_tax  
FROM  
  DiscountedPrices;
```



Les alias latéraux permettent de regrouper les mêmes calculs dans une seule requête.

Cette requête plus simple est essentiellement identique à l'exemple plus long d'expression de table commune, mais la logique est condensée parce que vous pouvez référencer immédiatement l'alias `prix_remisé` pour calculer la valeur `prix_total_après_taxe` dans la même requête.

SQL

```
SELECT
  product_id,
  product_name,
  price,
  price * 0.90 AS discounted_price,
  discounted_price * 1.05 AS total_price_after_tax
FROM
  products;
```

FROM

Spécifie la table ou vue à utiliser dans une instruction `SELECT` .

Syntaxe

SQL

```
<select_clause> FROM
  { table_name | ( <sub_query> ) } [ , ... ]
  [ <join_clause> [ , ... ] ]
```

Paramètres

Paramètre	Description
<code><select_clause></code>	Une instruction de requête utilisant l'instruction SQL <code>SELECT</code> . Pour plus de détails, consultez la section SELECT .
<code>nom_de_table</code>	Les noms d'une ou plusieurs tables ou vues que vous souhaitez interroger. Si vous spécifiez plusieurs tables dans la liste <code>DE</code> , séparées par des virgules, l'effet est le même que si vous effectuiez explicitement un JOINTURE CROISÉE sur toutes ces sources.
<code><sub_query></code>	Une ou plusieurs sous-requêtes utilisant l'instruction SQL <code>SELECT</code> . Une sous-requête génère une table à partir de laquelle la requête <code>clause_select</code> référence des données. Chaque sous-requête doit être placée entre parenthèses avec un nom de corrélation facultatif. Pour plus de détails, consultez la section SELECT .
<code><join_clause></code>	Une clause <code>REJOINDRE</code> utilisée pour récupérer des données à partir de deux tables ou plus pour votre requête. Pour plus de détails, consultez la section REJOINDRE .

JOIN

Combine des lignes provenant de plusieurs tables afin qu'elles puissent être accessibles par une requête.



Syntaxe

SQL

```
<select_query>
  FROM <table_reference1> <join_operation> <table_reference2>
  ON table1_column <boolean_operator> table2_column
```

Paramètre

Paramètre	Description
<select_query>	Une requête <code>SELECT</code> valide. Pour plus de détails, consultez SELECT .
<table_reference1>	Soit un nom de table ou une requête <code>SELECT</code> complète entre parenthèses <code>table_reference1</code> a priorité pour le retour des lignes pour <code>JOINTURE EXTERNE GAUCHE</code> , <code>SEMI JOIN</code> , et <code>ANTI JOIN</code> . Pour les règles spécifiques, consultez Types d'opérations JOIN .
<table_reference2>	Soit un nom de table ou une fonction <code>SELECT</code> complète entre parenthèses. <code>table_reference2</code> a priorité pour le retour des lignes pour <code>JOINTURE EXTERNE DROITE</code> . Pour les règles spécifiques, consultez Types d'opérations JOIN .
table1_column	Une colonne provenant de <code>référence_du_tableau</code> L'opération <code>REJOINDRE</code> utilise cette colonne pour faire correspondre les lignes avec <code>table2_column</code> afin de combiner les données pour l'ensemble de résultats.
<boolean_operator>	Les conditions de jointure peuvent utiliser toute expression booléenne, y compris <code>=</code> , <code>!=</code> , <code><</code> , <code>></code> , <code>=></code> , et <code><=</code>
table2_column	Une colonne provenant de <code>référence_du_tableau</code> non utilisée pour <code>table1_column</code> L'opération <code>REJOINDRE</code> utilise cette colonne pour faire correspondre les lignes en fonction de <code>table1_column</code> afin de combiner les données pour l'ensemble de résultats.

Types d'opérations JOIN (<join_operation>)

Ocient prend en charge les types suivants d'opérations `REJOINDRE`.

Syntaxe

SQL

```
<join_operation> ::=
{ INNER JOIN
| LEFT [ OUTER ] JOIN
| RIGHT [ OUTER ] JOIN
| FULL [ OUTER ] JOIN
| CROSS JOIN
| SEMI JOIN
| ANTI JOIN }
```

Descriptions des types JOIN



Type de JOIN	Description
JOINTURE INTERNE	Retourne toutes les lignes qui existent dans les deux tables. Ce type de jointure est le type de jointure par défaut.
JOINTURE EXTERNE GAUCHE	Retourne toutes les lignes de la première table, qu'il y ait ou non des lignes correspondantes dans la deuxième table. <code>LEFT JOIN</code> effectue la même opération que <code>JOINTURE EXTERNE GAUCHE</code> .
JOINTURE EXTERNE DROITE	Retourne toutes les lignes de la deuxième table, qu'il y ait ou non des lignes correspondantes dans la première table. <code>RIGHT JOIN</code> effectue la même opération que <code>JOINTURE EXTERNE DROITE</code> .
FULL OUTER JOIN	Retourne toutes les lignes correspondantes et non correspondantes. <code>FULL JOIN</code> effectue la même opération que <code>FULL OUTER JOIN</code> .
JOINTURE CROISÉE	Retourne le produit cartésien des tables jointes. Chaque ligne de la première table est jointe à toutes les lignes de la deuxième table. L'ensemble de résultats contient le nombre de lignes de la première table multiplié par le nombre de lignes de la deuxième table.
SEMI JOIN	Retourne les lignes de la première table qui ont au moins une correspondance dans la deuxième table. Aucune colonne de la deuxième table n'est disponible dans la requête.
ANTI JOIN	Retourne les lignes de la première table qui n'ont aucune correspondance dans la deuxième table. Aucune colonne de la deuxième table n'est disponible dans la requête.

Par défaut, les instructions `REJOINDRE` qui impliquent des sous-requêtes peuvent fonctionner latéralement si nécessaire. Ce comportement permet aux sous-requêtes de référencer les colonnes jointes des éléments précédents inclus dans la clause `DE`.

Par exemple, les deux parties de cette opération de jointure utilisent des sous-requêtes qui référencent la table `x`. Le mot-clé `LATÉRAL` est facultatif; l'opération de jointure agit de la même façon, qu'il soit inclus ou non.

```
SELECT * FROM
(
  SELECT
    1 AS cx1,
    2 AS cx2) AS x
INNER JOIN LATERAL (
  SELECT
    x.cx1 AS cy1,
    x.cx2 AS cy2) AS y
ON
  x.cx1 = y.cy1;
```

Les jointures latérales sont principalement utiles lorsqu'une colonne référencée de façon croisée est nécessaire pour calculer les lignes à joindre. Une application courante consiste à fournir une valeur d'argument pour une fonction retournant un ensemble.

Exemples

Ces exemples joignent deux tables :



- `jeux` — Une table des titres de jeux vidéo et de leurs `genre_ids`. Notez que certains jeux ont une valeur NULL assignée à leur `genre_id`.

nom_du_jeu	id_du_genre
Bataille stellaire 6000	9
Simulateur de nain	NULL
Futbol 2002	11
Attaque spatiale!	9
Fou fantastique	8
Blasto l'écureuil	5
Quête du camionneur	7
Bloquer, empiler et pleurer!	6
Journée de basketball	11
Mutant pizza	1
Plombier italien	5
Zombies!	1
Ninjas de l'espace	12
Maire de la lune	10
Héros des feuilles de calcul	NULL
Magnat du zoom	12
Braconnier de porche	6

- `genre` — Une table des genres de jeux vidéo, qui sont identifiés par des ID.



genre_name	id
Course	7
Puzzle	6
Aventure	2
Simulation	10
Tir	9
Divers	4
Action	1
Sports	11
Jeu de rôle	8
Plateforme	5
Stratégie	12
Combat	3

INNER JOIN

Cet exemple utilise une opération **JOINTURE INTERNE** pour capturer uniquement les lignes qui existent dans les deux tables. Les jeux avec des valeurs NULL pour leur **genre_id** sont éliminés de l'ensemble de résultats.

SQL

```
SELECT game_name,  
       genre_name  
FROM video_games.game  
     INNER JOIN video_games.genre ON game.genre_id = genre.id;
```

Résultat



nom_du_jeu	nom_du_genre
Fantasy Fool	Jeu de rôle
Blasto the Squirrel	Plateforme
Italian Plumber	Plateforme
Porch Poacher	Puzzle
Space Ninjas	Stratégie
Basketball Day	Sports
2002 Futbol	Sports
Moon Mayor	Simulation
Block, Stack, and Cry!	Puzzle
Zombies!	Action
Pizza Mutant	Action
Space Attack!	Jeu de tir
Trucker Quest	Course
Zoom Tycoon	Stratégie

LEFT OUTER JOIN

Cet exemple utilise une opération **JOINTURE EXTERNE GAUCHE** pour capturer toutes les lignes de la table de gauche (game), même si elles n'ont aucune ligne correspondante dans la table de droite (genre).

SQL

```
SELECT game_name,  
       genre_name  
FROM video_games.game  
LEFT OUTER JOIN video_games.genre ON game.genre_id = genre.id;
```

Résultat



game_name	genre_name
Fantasy Fool	Jeu de rôle
Space Ninjas	Stratégie
Zoom Tycoon	Stratégie
Zombies!	Action
Pizza Mutant	Action
Trucker Quest	Course
Blasto the Squirrel	Plateforme
2002 Futbol	Sports
Basketball Day	Sports
Italian Plumber	Plateforme
Block, Stack, and Cry!	Puzzle
Star Battle 6000	Shooter
Dwarf Simulator	NULL
Moon Mayor	Simulation
Porch Poacher	Puzzle
Spreadsheet Hero	NULL
Space Attack!	Shooter

RIGHT OUTER JOIN

Cet exemple utilise une opération **DROITE** **JOINTURE EXTERNE** pour capturer toutes les lignes de la table de droite (genre), même si elles n'ont aucune ligne correspondante dans la table de gauche (game).

SQL

```
SELECT game_name,  
       genre_name  
FROM video_games.game  
      RIGHT OUTER JOIN video_games.genre ON game.genre_id = genre.id;
```

Résultat



nom_du_jeu	nom_du_genre
NULL	Aventure
Trucker Quest	Course
Pizza Mutant	Action
Moon Mayor	Simulation
Fantasy Fool	Jeu de rôle
Space Attack!	Jeu de tir
NULL	Divers
Star Battle 6000	Jeu de tir
Space Ninjas	Stratégie
Porch Poacher	Réflexion
Block, Stack, and Cry!	Réflexion
Blasto the Squirrel	Plateforme
Italian Plumber	Plateforme
Zoom Tycoon	Stratégie
NULL	Combat
Zombies!	Action
2002 Futbol	Sports
Basketball Day	Sports

JOINTURE EXTERNE COMPLÈTE

Cet exemple utilise une opération `FULL OUTER JOIN` pour capturer toutes les lignes des deux tables, même si elles ne correspondent pas.

SQL

```
SELECT game_name,  
       genre_name  
FROM video_games.game  
      FULL OUTER JOIN video_games.genre ON game.genre_id = genre.id;
```

Résultat



nom_du_jeu	nom_du_genre
Journée Basketball	Sports
2002 Futbol	Sports
Zombies!	Action
Blasto l'écureuil	Plateforme
Maire de la Lune	Simulation
Quête du camionneur	Course
NULL	Divers
Bataille stellaire 6000	Jeu de tir
Attaque spatiale!	Jeu de tir
Héros du tableur	NULL
NULL	Aventure
NULL	Combat
Plombier italien	Plateforme
Mutant pizza	Action
Simulateur de nain	NULL
Fou fantastique	Jeu de rôle
Bloquez, empilez et pleurez !	Puzzle
Ninjas de l'espace	Stratégie
Magnat du zoom	Stratégie
Voleur de porche	Puzzle

CROSS JOIN

Cet exemple utilise une opération `JOINTURE CR0ISÉE` pour capturer toutes les combinaisons possibles de lignes des deux tables, qu'elles correspondent ou non. Notez que contrairement aux autres opérations `REJOINDRE`, `JOINTURE CR0ISÉE` ne nécessite pas d'instruction `ACTIVÉ`.

SQL

```
SELECT game_name,  
       genre_name  
FROM video_games.genre  
CROSS JOIN video_games.game;
```

Résultat

Comme l'ensemble de résultats pour cet exemple de CROSS JOIN contient plus de 200 lignes, les résultats sont abrégés.



nom_du_genre	id_du_genre
Héros du tableur	Jeu de rôle
Héros du tableur	Divers
Héros du tableur	Sports
Héros du tableur	Plateforme
Héros du tableur	Stratégie
Héros du tableur	Jeu de tir
Héros du tableur	Combat
Héros du tableur	Action
Héros du tableur	Simulation
Héros du tableur	Course
Héros du tableur	Puzzle
Héros du tableur	Aventure
Attaque spatiale!	Jeu de rôle
Attaque spatiale!	Divers
Attaque spatiale!	Sports
Attaque spatiale !	Plateforme
Attaque spatiale !	Stratégie
Attaque spatiale !	Jeu de tir
Attaque spatiale !	Combat
Attaque spatiale !	Action
Attaque spatiale !	Simulation
Attaque spatiale !	Course
Attaque spatiale !	Puzzle
Attaque spatiale !	Aventure
...	...

SEMI JOIN

Cet exemple utilise une opération `SEMI JOIN` pour récupérer les noms de genres ayant au moins une correspondance dans la table des jeux. Les lignes ne sont incluses qu’une seule fois, même s’il existe plusieurs correspondances.



SQL

```
SELECT genre_name,  
       id  
FROM video_games.genre  
SEMI JOIN video_games.game ON genre.id = game.genre_id;
```

Résultat

genre_name	genre_id
Simulation	10
Shooter	9
Course	7
Stratégie	12
Puzzle	6
Plateforme	5
Action	1
Sports	11
Jeu de rôle	8

ANTI JOIN

Cet exemple utilise une opération `ANTI JOIN` pour récupérer toutes les lignes de la table `game` qui ne correspondent à aucune ligne de la table `genre`.

SQL

```
SELECT game_name  
FROM video_games.game  
ANTI JOIN video_games.genre ON game.genre_id = genre.id;
```

Résultat

game_name
Spreadsheet Hero
Dwarf Simulator

WHERE

Filtre les lignes en fonction d'une condition spécifiée.

Syntaxe



SQL

```
<select_query>
WHERE { column_name <filter_condition> filter_value |
[ NOT ] EXISTS ( <sub_query> ) }
```

Paramètres

Paramètre	Description
<select_query>	Une requête <code>SELECT</code> valide.
nom_de_colonne	Une colonne utilisée pour le regroupement qui est évaluée par la condition <code>condition_de_filtre</code> .
valeur_du_filtre	Une valeur utilisée pour évaluer le <code>nom_de_colonne</code> spécifié. Pour plus de détails, consultez la section <filter_condition> .
<sub_query>	Une sous-requête avec une condition de filtre qui suit la clause <code>EXISTE</code> . Pour plus de détails, consultez la section EXISTE .

<filter_condition>

Une combinaison logique de prédicats utilisée pour évaluer le `nom_de_colonne` référencé selon la valeur `filter_value` .

Syntaxe

SQL

```
<filter_condition> ::=
{
  | ==
  | <>
  | !=
  | [ NOT ] EQUALS
  | <
  | <=
  | >
  | >=
  | [ NOT ] IN
  | [ NOT ] LIKE
  | [ NOT ] SIMILAR TO
  | BETWEEN
  | FOR SOME
  | FOR ALL
  | IS [ NOT ] NULL
  | IS [ NOT ] DISTINCT FROM
}
```

Définitions



Opérateur	Description	Exemple
<code>=</code> , <code>==</code> , <code>ÉGALE</code>	Égal à.	<code>Auteur = 'Alcott'</code>
<code><></code> , <code>!=</code>	Différent de	<code>Département <> 'Ventes'</code>
<code>></code>	Supérieur à	<code>Date_d'embauche > '2012-01-31'</code>
<code><</code>	Inférieur à	<code>Prime < 50000.00</code>
<code>>=</code>	Supérieur ou égal à	<code>Personnes à charge >= 2</code>
<code><=</code>	Inférieur ou égal à	<code>Taux <= 0,05</code>
<code>[PAS]</code> <code>COMME</code>	Commence par un modèle de caractères à faire correspondre. Le modèle est placé entre parenthèses et peut inclure les caractères génériques <code>%</code> , représentant zéro, un ou plusieurs caractères, et <code>_</code> représentant un seul caractère. Pour avoir un caractère littéral <code>%</code> ou <code>_</code> qui n'agit pas comme caractère générique du côté droit d'une expression <code>LIKE</code> ou <code>NOT LIKE</code> , placez une barre oblique inverse avant le caractère.	<code>Nom_complet LIKE 'Will%'</code>
<code>[PAS]</code> <code>SIMILAIRE</code> <code>À</code>	Pour des informations sur l'utilisation, consultez la section Opérateur SIMILAR TO .	<code>'abc' SIMILAR TO 'abc'</code>
<code>ENTRE</code> <code>value1 ET</code> <code>value2</code>	Nécessite deux valeurs. Le filtre est évalué à TRUE si la colonne spécifiée se trouve dans la plage entre les deux valeurs. <code>ENTRE</code> inclut les bornes. Les valeurs peuvent être des heures, des nombres ou des chaînes.	<code>Ventes ENTRE 1000 ET 5000</code>
<code>FOR_SOME()</code>	Évalue un tableau pour déterminer si au moins une valeur répond aux critères du filtre. Dans l'exemple, l'opérateur <code>FOR_SOME()</code> renverrait toutes les lignes de la colonne <code>col_int_array</code> où au moins une valeur est supérieure à 10. Pour plus de détails, consultez Filtres de tableau .	<code>FOR_SOME(col_int_array) > 10</code> <code>FOR_SOME(col_int_array) > 10</code>
<code>FOR_ALL()</code>	Évalue toutes les valeurs d'un tableau pour déterminer si elles répondent toutes aux critères du filtre. Dans l'exemple, l'opérateur <code>FOR_ALL()</code> renverrait toutes les lignes de la colonne <code>col_int_array</code> où toutes les valeurs sont supérieures à 10. Pour plus de détails, consultez Filtres de tableau .	<code>FOR_ALL(col_int_array) > 10</code> <code>FOR_ALL(col_int_array) = 1</code>
<code>[PAS]</code> <code>DANS</code>	Égal à l'une de plusieurs valeurs possibles	<code>CodeDépartement IN (101, 103, 209)</code>
<code>IS [NOT] NULL</code>	Comparer à <code>NULL</code> (données manquantes)	<code>L'adresse N'EST PAS NULLE</code>



Opérateur	Description	Exemple
<code>IS [NOT] DISTINCT FROM</code>	Compare l'égalité de deux expressions pour obtenir un résultat booléen, y compris les comparaisons impliquant des valeurs <code>NULL</code> . Cet opérateur garantit des comparaisons fiables dans les scénarios où les valeurs <code>NULL</code> doivent être traitées comme des points de données significatifs. Normalement, <code>NULL = NULL</code> est évalué à <code>faux</code> parce que <code>NULL</code> représente une valeur inconnue. Cependant, l'instruction <code>a N'EST PAS DISTINCT DE b</code> traite <code>NULL</code> comme une valeur comparable et renvoie <code>vrai</code> lorsque les deux valeurs sont <code>NULL</code> ou lorsque <code>a = b</code> . Inversement, <code>a est différent de b</code> renvoie <code>vrai</code> si les valeurs sont différentes ou si une valeur est <code>NULL</code> tandis que l'autre n'est pas <code>NULL</code> .	<code>La dette n'est pas DISTINCTE DES créances</code>

EXISTS

Une clause `EXISTE` utilisée dans une instruction `où` évalue si une sous-requête retourne des lignes.

Pour chaque ligne que la base de données calcule dans la requête externe, si la sous-requête retourne au moins une ligne, la clause `EXISTS` est évaluée à `true`. Dans ce cas, la requête externe retourne sa ligne.

Si la sous-requête retourne zéro ligne, la clause `EXISTE` est évaluée à `faux`, et la requête externe exclut ces lignes.

La clause `N'EXISTE PAS` effectue la logique booléenne opposée. Dans ce cas, la requête externe retourne des lignes uniquement lorsque la sous-requête ne comporte aucune correspondance.

Exemples

Ces exemples utilisent deux tables, une pour les départements de l'entreprise et l'autre pour les employés affectés à ces départements.

Créez la table `départements` pour les données des départements.

SQL

```
CREATE TABLE departments (  
  id INT,  
  name VARCHAR(50)  
);
```

Créez la table `employés` pour les données des employés.

SQL

```
CREATE TABLE employees (  
  id INT,  
  name VARCHAR(50),  
  department_id INT  
);
```

Insérez des données de département pour trois départements, `RH`, `IT`, et `Marketing`, dans la table `départements`.



SQL

```
INSERT INTO departments (id, name) VALUES
(1, 'HR'),
(2, 'IT'),
(3, 'Marketing');
```

Insérez des données d'employés pour quatre employés, `Alice`, `Bob`, `Charlie`, et `David`, dans la table `employés`.

SQL

```
INSERT INTO employees (id, name, department_id) VALUES
(1, 'Alice', 1),
(2, 'Bob', 2),
(3, 'Charlie', 2),
(4, 'David', 4);
```

Trouver les employés appartenant à un département existant

Cet exemple utilise une clause `EXISTS` pour trouver les noms de la table `employés` qui sont affectés à un département répertorié dans la table `départements`.

SQL

```
SELECT name
FROM employees AS e
WHERE EXISTS (
  SELECT *
  FROM departments AS d
  WHERE e.department_id = d.id
);
```

La requête retourne tous les employés sauf `David`, qui est affecté à un département qui n'existe pas dans la table `départements`.

Sortie

Text

```
Bob
Charlie
Alice
```

Trouver les départements qui ont des employés

Cette requête retourne les départements qui ont au moins un employé. La sous-requête utilise `SELECT 1` pour vérifier si des lignes correspondantes existent dans la table `employés`. La sortie est la même si la requête utilise `SELECT *` à la place.



SQL

```
SELECT name
FROM departments AS d
WHERE EXISTS (
    SELECT 1
    FROM employees AS e
    WHERE e.department_id = d.id
);
```

Les résultats de la requête excluent le département `Marketing` parce qu'aucun employé ne lui appartient.

Sortie

Text

HR
IT

Filtrer les départements en fonction d'une table non corrélée

Dans cet exemple, la requête externe filtre la table `départements` où `id != 3` (en excluant le département `Marketing`). La sous-requête vérifie s'il existe au moins un employé avec un identifiant de département inférieur à `4`.

La requête est non corrélée parce que la sous-requête ne référence pas la table `départements`.

SQL

```
SELECT *
FROM departments d
WHERE d.id != 3
AND EXISTS (
    SELECT *
    FROM employees e
    WHERE e.department_id < 4
);
```

Résultat

Text

HR
IT

Trouver des employés sans département valide

Cet exemple utilise une clause `N'EXISTE PAS`. L'exemple retourne uniquement les employés attribués à un identifiant de département `department_id` qui n'est pas répertorié dans la table `départements`.



SQL

```
SELECT name
FROM employees AS e
WHERE NOT EXISTS (
  SELECT *
  FROM departments AS d
  WHERE e.department_id = d.id
);
```

Résultat : David

Opérateur SIMILAR TO

`SIMILAR TO` est un mot-clé qui étend l'opérateur `LIKE` en ajoutant davantage de fonctionnalités pour le filtrage des correspondances, y compris plusieurs métacaractères utilisés dans les expressions régulières.

`%` et `_` agissent tous deux comme des opérateurs génériques, mais les autres métacaractères pris en charge correspondent aux expressions régulières traditionnelles.

Syntaxe

SQL

```
WHERE string1 SIMILAR TO string2
```

Ce tableau décrit les métacaractères pris en charge par le mot-clé `SIMILAR TO`.



Métacaractère	Description
%	Joker répété, correspond à n'importe quels caractères zéro fois ou plus. Utilisation équivalente à <code>LIKE</code> .
_	Joker, correspond exactement à un caractère. Utilisation équivalente à <code>LIKE</code> .
	Alternative, signifiant l'une de deux possibilités (<code>a b</code> représente <code>a</code> ou <code>b</code>).
*	Répétition zéro fois ou plus.
+	Répétition une fois ou plus.
?	Répétition zéro ou une fois.
{m}	Répétition exactement <code>m</code> fois.
{m,n}	Répétition entre <code>m</code> et <code>n</code> fois (inclusivement).
{m, }	Répétition au moins <code>m</code> fois.
()	Regroupement logique.
[]	Classe de caractères, équivalente aux classes de caractères dans les expressions régulières. Exemple : <code>[a-z]</code> représente n'importe quelle lettre minuscule anglaise.
^	Ancre de début de ligne et négation dans les classes de caractères.
\$	Ancre de fin de ligne.

`SIMILAR TO` prend également en charge l'échappement de ces métacaractères avec `\` et certaines séquences d'échappement prises en charge par les langages de la famille C.



Métacaractère	Description
<code>\a</code>	Le caractère d'alerte/sonnerie.
<code>\b</code>	Le caractère de retour arrière.
<code>\B</code>	Un seul caractère <code>\</code> . Équivalent à <code>\\</code> .
<code>\cX</code>	Le caractère ayant les cinq bits de poids faible identiques à ceux de <code>x</code> avec tous les bits de poids fort à zéro.
<code>\e</code>	Le caractère de valeur numérique 27, Esc.
<code>\f</code>	Saut de page.
<code>\n</code>	Nouvelle ligne.
<code>\r</code>	Retour chariot.
<code>\t</code>	Tabulation horizontale.
<code>\uwx yz</code>	<code>wxyz</code> représente quatre chiffres hexadécimaux, incluant deux caractères : un avec la valeur numérique <code>wx</code>, et un avec la valeur numérique <code>yz</code>.
<code>\xd...</code>	<code>d</code> représente n'importe quel nombre de chiffres hexadécimaux. Si le nombre de chiffres est impair, un 0 est ajouté à gauche de la séquence. Les caractères numériques avec des valeurs représentées par chaque groupe de deux symboles de <code>d</code>.
<code>\0</code>	Caractère NULL 0.
<code>\xy</code>	<code>xy</code> sont des chiffres octaux qui représentent le caractère avec la valeur numérique <code>xy</code>.
<code>\xyz</code>	<code>xyz</code> sont des chiffres octaux qui représentent le caractère avec la valeur numérique <code>xyz</code>. Si <code>x</code> est supérieur ou égal à <code>4</code>, la base de données renvoie deux caractères : la valeur numérique <code>1</code> et la valeur numérique des 8 bits inférieurs de <code>xyz</code>.
<code>\d</code>	Correspond à n'importe quel chiffre.
<code>\s</code>	Correspond à n'importe quel espace blanc.
<code>\w</code>	Correspond à n'importe quel caractère alphanumérique ou <code>_</code> .
<code>\D</code>	Correspond à n'importe quel caractère non numérique.
<code>\S</code>	Correspond à n'importe quel caractère autre qu'un espace blanc.
<code>\W</code>	Correspond à n'importe quel caractère non alphanumérique qui n'est pas non plus <code>_</code> .
<code>\A</code>	Le début d'une ligne. Équivalent à <code>^</code> .
<code>\m</code> , <code>\M</code> , ou <code>\y</code>	La limite d'un mot. Équivalent à <code>\b</code> dans les expressions régulières normales.



Métacaractère	Description
<code>\Y</code>	Toute position qui n'est pas la limite d'un mot.
<code>\Z</code>	La fin d'une ligne. Équivalent à <code>\$</code> dans les expressions régulières normales.
<code>*</code>	Le caractère étoile.
<code>\+</code>	Le caractère plus.
<code>\?</code>	Le caractère point d'interrogation.

- La base de données ignore tout métacaractère `\` qui ne précède pas un métacaractère (y compris `\`) ou une séquence d'échappement dans la table des métacaractères.
- Les expressions non valides (quantificateurs sans préfixe, parenthèses non fermées, etc.) entraînent des erreurs de requête.
- La base de données ne reconnaît pas le caractère `.` comme un métacaractère, contrairement au comportement des expressions régulières traditionnelles.
- La base de données applique les métacaractères d'échappement qui s'appliquent aux littéraux de chaîne. Pour faire correspondre une chaîne contenant un seul `'\'`, vous devrez peut-être utiliser `'\\'`.

Exemples

SQL

```
'abc' SIMILAR TO 'abc'           true
'abc' SIMILAR TO 'a'             false
'abc' SIMILAR TO '%(b|d)%'       true
'abc' SIMILAR TO '(b|c)%'        false
'-abc-' SIMILAR TO '%\mabc\M%'   true
'xabcy' SIMILAR TO '%\mabc\M%'   false
```

Filtres de tableau

Une expression de filtre peut utiliser les fonctions `FOR_SOME()` et `FOR_ALL()` pour appliquer un prédicat à toutes les valeurs d'un tableau d'entrée.

Les fonctions de filtre de tableau suivent les règles suivantes :

- Elles peuvent uniquement évaluer des types tableau.
- Elles peuvent se trouver à gauche ou à droite d'une expression de comparaison booléenne, mais pas des deux côtés en même temps.
- Elles doivent utiliser directement un opérateur de comparaison booléenne.
- Elles ne peuvent pas utiliser les mots-clés SQL `QUELQUE` et `TOUT`.

Comportement du filtre avec des tableaux vides

Les fonctions de filtre de tableau ont un comportement unique lors de l'évaluation de tableaux vides.

- `FOR_ALL()` évalue un tableau vide comme étant `VRAI`.
- `FOR_SOME()` évalue un tableau vide comme étant `FAUX`.



Si des tableaux vides doivent être évalués pour obtenir un résultat différent, vous pouvez utiliser la fonction `LONGUEUR_TABLEAU` pour spécifier une longueur minimale de tableau. Par exemple, cette instruction évaluerait un tableau comme étant `VRAI` uniquement s'il n'est pas vide et que toutes les valeurs correspondent à `%ocient%` :

SQL

```
ARRAY_LENGTH(array_col) > 0 AND FOR_ALL(array_col) LIKE '%ocient%
```

Pour plus de détails sur les fonctions de tableau, consultez la page [Fonctions et opérateurs de tableau](#).

Comportement du filtre avec des lignes NULL

Le filtrage avec des fonctions de tableau peut produire des résultats différents selon que la ligne du tableau est NULL ou que les valeurs contenues dans le tableau sont NULL.

Si `FOR_SOME()` ou `FOR_ALL()` évaluent une ligne NULL (la ligne elle-même est NULL, et non le fait que le tableau contienne des valeurs NULL), le résultat est alors toujours `FAUX`.

Pour vérifier si un tableau contient des valeurs NULL, vous devez utiliser l'opérateur `EST NUL`. Par exemple :

SQL

```
FOR_SOME(array_col) IS NULL
```

Les opérateurs de comparaison de tableau, tels que `@>`, `<@`, et `&&` ne respectent pas la logique booléenne pour les valeurs NULL. Pour plus de détails, consultez la page [Fonctions et opérateurs de tableau](#).

GROUP BY

Regroupe les lignes ayant les mêmes valeurs dans des lignes récapitulatives, selon une fonction d'agrégation spécifiée.

Syntaxe

SQL

```
SELECT <select_list_clause> FROM table_name
  GROUP BY { column_name | expression | integer } [ , ... ]
  [ HAVING <filter_condition> ]
```

Paramètres



Paramètre	Description
<code><select_list_clause></code>	Une liste d'une ou plusieurs colonnes pour la requête <code>SELECT</code> . Pour utiliser <code>GROUPE PAR</code> afin de résumer l'ensemble de résultats de la requête, au moins une des colonnes dans <code>liste_des_colonnes</code> doit inclure une fonction d'agrégation, telle que <code>SOMME()</code> , <code>COUNT()</code> , ou <code>MAX()</code> . Pour obtenir une liste des fonctions prises en charge, consultez la page Fonctions d'agrégation .
<code>nom_de_table</code>	La table à utiliser pour la requête.
<code>nom_de_colonne</code>	Le nom d'une ou plusieurs colonnes à utiliser pour regrouper l'ensemble de résultats. Si vous spécifiez plusieurs colonnes, l'ensemble de résultats est regroupé selon chaque combinaison unique de valeurs provenant des colonnes.
<code>expression</code>	Toute combinaison de valeurs littérales, de noms de colonnes, d'expressions arithmétiques, de parenthèses et d'appels de fonction.
<code>entier</code>	Un entier représentant la position des colonnes référencées dans <code>liste_des_colonnes</code> . La première position commence à <code>1</code> .
<code><filter_condition></code>	Une clause <code>AYANT</code> qui filtre les groupes agrégés selon une condition spécifiée. Pour plus de détails, consultez la section <filter_condition> .

Exemple

Cet exemple utilise une base de données de films pour calculer le montant total dépensé pour la production de films par année.

SQL

```
SELECT YEAR(release_date),
       SUM(budget)
FROM movies
GROUP BY 1
ORDER BY 1 ASC;
```

Sortie



year(release_date)	sum(budget)
1968	10500000
1979	31500000
1981	18000000
1982	28000000
1992	14000000
1997	200000000
2003	264000000
2009	237000000
2010	350000000
2012	670000000
2013	350000000
2015	414000000

HAVING

Filtre les lignes agrégées en fonction d'une condition spécifiée.

`AYANT` fonctionne dans une instruction `GROUPE PAR` en définissant un filtre pour les lignes à agréger et à regrouper.

Pour plus d'informations sur l'utilisation de `GROUPE PAR` dans une instruction de requête, consultez [GROUPE PAR](#).

Syntaxe

SQL

```
GROUP BY column_name HAVING <filter_condition>
```

Paramètres

Paramètre	Description
<code>nom_de_colonne</code>	Une colonne utilisée pour le regroupement qui est évaluée par la <code>condition_de_filtre</code> .
<code><filter_condition></code>	Une combinaison logique de prédicats booléens. Pour obtenir des informations sur les prédicats logiques pris en charge, consultez <filter_condition> .

Exemple

Cet exemple calcule le montant total dépensé pour la production de films par année. La clause `AYANT` filtre toutes les lignes qui n'ont pas une somme d'au moins 100 millions \$.



SQL

```
SELECT YEAR(release_date),
       SUM(budget)
FROM movies
GROUP BY 1
HAVING SUM(budget) > 100000000
ORDER BY 1 ASC;
```

Résultat

year(release_date)	sum(budget)
1997	200000000
2003	264000000
2009	237000000
2010	350000000
2012	670000000
2013	350000000
2015	414000000

ORDER BY

Trie le jeu de résultats par ordre croissant ou décroissant selon une ou plusieurs colonnes spécifiées.

Si vous spécifiez plusieurs colonnes, elles sont triées hiérarchiquement de gauche à droite.

Syntaxe

SQL

```
ORDER BY { column_position | column_name } [ ASC | DESC ]
        [ NULLS FIRST | NULLS LAST ] [ , ... ]
```

Paramètres



Paramètre	Type de données	Description
<code>column_position</code>	Entier	La position d'une colonne à utiliser pour le tri <code>TRIER PAR</code> . Les positions des colonnes commencent à <code>1</code> .
<code>nom_de_colonne</code>	Chaîne	Le nom d'une colonne à utiliser pour le tri <code>TRIER PAR</code> .
<code>ASC DESC</code>	Chaîne	Facultatif. Indique si la colonne doit être triée en ordre croissant (<code>ASC</code>) ou décroissant (<code>DESC</code>). Si non spécifié, la valeur par défaut est <code>ASC</code> .
<code>NULLS FIRST NULLS LAST</code>	Chaîne	Facultatif. <code>NULLS FIRST</code> signifie que les valeurs nulles sont placées au début du jeu de résultats. <code>NULLS LAST</code> signifie que les valeurs nulles sont placées à la fin du jeu de résultats. Si non spécifié, la valeur par défaut est <code>NULLS FIRST</code> .

Exemple

Dans cet exemple, l'instruction `TRIER PAR` trie les films du plus récent au plus ancien.

SQL

```
SELECT release_date, title
FROM movies
ORDER BY release_date DESC;
```

Résultat



date_de_sortie	titre
2015-06-17	Minnows
2015-06-09	Monde triasique
2015-04-01	Furie 7
2013-11-27	Glacial
2013-04-18	Iron Chef 3
2012-10-25	Espionman 16
2012-07-16	Superhéros 23
2012-04-25	Le Tentpole
2010-06-30	Le dernier aviateur
2010-06-16	Véhicule de marchandises 5
2009-12-10	Pourquoi CGI
2003-12-01	Épées et fourreaux
2003-10-10	Billy le Killy
2003-07-09	Pirates de Palm Springs
1997-11-18	Titania
1992-08-07	Bang Bang Western
1982-06-25	Blade Walker
1981-06-12	Les aventuriers du temple de la jungle
1979-08-15	Apocalypse jamais
1968-04-10	Odyssée de l'espace 6000

LIMIT

Limite le nombre de lignes retournées par une requête à une quantité spécifiée.

Les lignes retournées sont non déterministes à moins que vous n'utilisiez une clause **TRIER PAR**.

Syntaxe

SQL

```
<select_query> LIMIT limit_number
```

Paramètres



Paramètre	Description
<code><select_query></code>	Une requête <code>SELECT</code> valide.
<code>limit_number</code>	Le nombre de lignes à retourner à partir de la requête, spécifié comme un entier positif ou toute expression scalaire constante qui s'évalue à un entier positif.

Exemples

Limiter les lignes retournées à l'aide d'un nombre

Cet exemple utilise `LIMIT` pour restreindre le nombre de titres de films retournés à seulement trois.

SQL

```
SELECT title FROM movies LIMIT 3;
```

Sortie

SQL

```
Minnows
CGI Why
Billy the Killy
```

Limiter les lignes retournées à l'aide d'une expression

L'instruction SQL `LIMIT` peut également accepter des expressions. Cette requête utilise l'expression `1+2` avec la table `sys.dummy` pour créer une table de trois entiers incréments. Pour plus de détails, consultez [Générer des tables à l'aide de sys.dummy](#).

SQL

```
SELECT c1 FROM sys.dummy10 LIMIT 1+2;
```

Sortie

SQL

```
1
2
3
```

OFFSET

Ignore un nombre spécifié de lignes du jeu de résultats.

Les lignes retournées sont non déterministes à moins que vous n'utilisiez une clause [TRIER PAR](#).



Syntaxe

SQL

```
<select_query> OFFSET offset_number
```

Paramètres

Paramètre	Description
<select_query>	Une requête SELECT valide.
offset_number	Le nombre de lignes à ignorer lors du retour des résultats de la requête, spécifié comme un entier positif ou toute expression scalaire constante qui s'évalue à un entier positif.

Exemples

Ignorer des lignes du jeu de résultats à l'aide d'un nombre

Dans cet exemple, l'instruction `TRIER PAR` trie les résultats par ordre chronologique. Par conséquent, la clause `DÉCALAGE` supprime les six films les plus anciens du jeu de résultats.

SQL

```
SELECT release_date,
       title
FROM movies
ORDER BY release_date OFFSET 6;
```

Résultat



release_date	title
2003-07-09	Pirates of Palm Springs
2003-10-10	Billy the Killy
2003-12-01	Swords & Scabbards
2009-12-10	CGI Why
2010-06-16	Merchandise Vehicle 5
2010-06-30	The Last Airman
2012-04-25	The Tentpole
2012-07-16	Superhero 23
2012-10-25	Spyman 16
2013-04-18	Iron Chef 3
2013-11-27	Frigid
2015-04-01	Fury 7
2015-06-09	Triassic World
2015-06-17	Minnows

Ignorer des lignes du jeu de résultats à l'aide d'une expression

L'instruction SQL `DÉCALAGE` peut également accepter des expressions. Cette requête utilise l'expression `3+4` avec la table `sys.dummy` pour créer une colonne d'entiers incrémentiels en ignorant les sept premières lignes sur 10. Pour plus de détails, consultez [Générer des tables à l'aide de sys.dummy](#).

SQL

```
SELECT c1 FROM sys.dummy10 OFFSET 3+4;
```

Résultat

SQL

```
8
9
10
```

INTERSECT

Retourne toutes les lignes qui correspondent entre deux requêtes `SELECT` distinctes. Pour utiliser `INTERSECT`, les deux requêtes doivent être compatibles, ce qui signifie qu'elles doivent retourner le même nombre de colonnes et avoir des types de données similaires.

Par défaut, l'instruction SQL élimine les lignes dupliquées sauf si la requête inclut le mot-clé facultatif `TOUT`. L'utilisation du mot-clé `DISTINCT` équivaut à ce comportement par défaut.



Syntaxe

SQL

```
<select_query1> INTERSECT [ ALL | DISTINCT ] <select_query2>
```

Paramètres

Paramètre	Description
<select_query1>	Une requête SELECT valide.
<select_query2>	Une requête SELECT valide.

Exemple

Cet exemple comporte deux requêtes `SELECT` avec une instruction `INTERSECT`. La requête complète récupère les films ayant eu un budget d'au moins 200 millions de dollars, mais ayant également généré plus d'un milliard de dollars de revenus.

SQL

```
SELECT *
FROM movies
WHERE revenue > 1000000000
INTERSECT
SELECT *
FROM movies
WHERE budget > 200000000;
```

Résultat



movie_id	title	budget	popularity	release_date	revenue	runtime	movie_status	vote_average	vote_count
597	Titania	20000000	100.025899	1997-11-18	1845034188	194	Sorti	7.5	7562
19995	CGI Why	23700000	150.437577	2009-12-10	2787965087	162	Sorti	7.2	11800
49026	Superhéros 23	25000000	112.31295	2012-07-16	1084939099	165	Publié	7.6	9106
10193	Véhicule de marchandise 5	20000000	59.995418	2010-06-16	1066969703	103	Publié	7.6	4597
211672	Ménés	74000000	875.581305	2015-06-17	1156730962	91	Publié	6.4	4571
122	Épées et fourreaux	94000000	123.630332	2003-12-01	1118888979	201	Publié	8.1	8064
37724	Spyman 16	20000000	93.004993	2012-10-25	1108561013	143	Publié	6.9	7604
109445	Glacial	15000000	165.125366	2013-11-27	1274219009	102	Publié	7.3	5295
168259	Furious 7	19000000	102.322217	2015-04-01	1506249360	137	Sorti	7.3	4176
68721	Iron Chef 3	20000000	77.68208	2013-04-18	1215439994	130	Sorti	6.8	8806
135397	Triassic World	15000000	418.708552	2015-06-09	1513528810	124	Sorti	6.5	8662
24428	Le chapiteau	22000000	144.448633	2012-04-25	1519557910	143	Publié	7.4	11776

EXCEPT

Le `SAUF` mot-clé retourne l'ensemble de résultats d'une première requête `SELECT` moins toute ligne correspondante d'une deuxième requête `SELECT` .

`SAUF` exige que les deux requêtes soient compatibles, ce qui signifie qu'elles doivent retourner le même nombre de colonnes et avoir des types de données similaires. Par défaut, l'ensemble de résultats élimine les lignes en double, sauf si la requête inclut le mot-clé facultatif `TOUT` . L'utilisation du mot-clé `DISTINCT` correspond à ce comportement par défaut.

Le `SAUF` mot-clé peut également être utilisé pour exclure des colonnes spécifiques d'une requête `SELECT *` . Pour obtenir des informations sur cette utilisation alternative, consultez la syntaxe et l'exemple de [SELECT](#) .

Syntaxe



SQL

```
<select_query1> EXCEPT [ ALL | DISTINCT ] <select_query2>
```

Paramètres

Paramètre	Description
<code><select_query1></code>	Une requête SELECT valide qui est comparée à <code>select_query2</code> .
<code>1></code>	Les résultats d'une instruction EXCEPT sont les lignes dans <code>select_query1</code> qui ne correspondent à aucune ligne dans <code>select_query2</code> .
<code><select_query2></code>	Une requête SELECT valide qui est comparée à <code>select_query1</code> .
<code>2></code>	

Exemple

Dans cet exemple, la première instruction `SELECT` interroge toutes les lignes de la table `movies`. La clause `SAUF` et la deuxième instruction `SELECT` éliminent des résultats toutes les lignes contenant des films dont le budget était supérieur à `20000000`.

SQL

```
SELECT *  
FROM movies  
EXCEPT  
SELECT *  
FROM movies  
WHERE budget > 20000000;
```

Résultat

titre	budget	popularité	date_de_sortie	revenus	durée	statut_du_film	moyenne_des_votes	nombre_de_votes
Bang Bang Western	14000000	37.380435	1992-08-07	159157447	131	Sorti	7.7	1113
Space Odyssey 6000	10500000	86.201184	1968-04-10	68700000	149	Sorti	7.9	2998
Raiders of the Jungle Temple	18000000	68.159596	1981-06-12	389925971	115	Sorti	7.7	3854

UNION

Retourne l'ensemble de résultats combiné de deux requêtes `SELECT` ou plus.

Par défaut, `UNION` élimine les lignes dupliquées de l'ensemble de résultats sauf si vous spécifiez le mot-clé `TOUT`. L'utilisation du mot-clé `DISTINCT` correspond à ce comportement par défaut.

Syntaxe



SQL

<select_query1> UNION [ALL | DISTINCT] <select_query2> [...]

Paramètres

Paramètre	Type de données	Description
select_query1	String	Une requête <code>SELECT</code> valide qui est combinée avec <code>select_query2</code> .
select_query2	String	Une requête <code>SELECT</code> valide qui est combinée avec <code>select_query1</code> .

Exemple

Cet exemple utilise `UNION` pour fusionner deux requêtes distinctes pour des colonnes identiques dans le même ensemble de résultats.

SQL

```
SELECT title,
       popularity,
       vote_average
FROM movie
WHERE popularity > 800
UNION
SELECT title,
       popularity,
       vote_average
FROM movies.movie
WHERE vote_average > 7.5;
```

Sortie

titre	popularité	moyenne_des_votes
Les aventuriers du temple de la jungle	68.159596	7.7
Ménés	875.581305	6.4
Aucune apocalypse	49.973462	8
Superhéros 23	112.31295	7.6
Billy le Tueur	79.754966	7.7
Bang Bang Western	37.380435	7.7
Marcheur de lames	94.056131	7.9
Épées et fourreaux	123.630332	8.1
Véhicule de marchandises 5	59.995418	7.6
Odyssée de l'espace 6000	86.201184	7.9



UTILISATION

Remplace diverses configurations système pour traiter une requête spécifiée.

Le mot-clé `UTILISATION` est requis uniquement pour la première substitution de requête, et non pour les suivantes.

Pour obtenir des descriptions des configurations de requête prises en charge, consultez le tableau des paramètres ci-dessous.

Syntaxe

SQL

```
<select_query>
USING
  [ SCHEDULING_PRIORITY = priority_value ]
  [ MAX_ROWS_RETURNED = max_rows ]
  [ MAX_ELAPSED_TIME = max_elapsed_time ]
  [ MAX_TEMP_DISK_USAGE = max_temp_disk_usage ]
  [ CACHE_MAX_TIME = cache_max_time ]
  [ CACHE_MAX_BYTES = cache_max_bytes ]
  [ SERVICE CLASS service_class ]
```

Paramètres



Paramètre	Description
<code><select_query></code>	Une requête <code>SELECT</code> valide.
<code>value_priority</code>	Une clause facultative qui spécifie la priority à laquelle la requête doit s'exécuter. La valeur de priorité est un nombre à virgule flottante qui indique la priorité par rapport aux autres requêtes. La valeur de priorité ne spécifie aucun pourcentage des ressources utilisées par la requête ni aucun degré spécifique de différence entre les requêtes. Si elle n'est pas spécifiée, la valeur par défaut est <code>SCHEDULING_PRIORITY = 1.0</code>.
<code>max_rows</code>	Une clause facultative qui limite le nombre de lignes renvoyées par une requête. Si une requête dépasse ce nombre de lignes, le système l'interrompt. Par exemple, <code>max_rows_returned = 10</code> interrompt toute requête renvoyant plus de dix lignes.
<code>max_elapsed_time</code>	Une clause facultative qui limite la durée d'exécution d'une requête. Si une requête dépasse cette limite de temps en secondes, le système l'interrompt. Par exemple, <code>max_elapsed_time = 10</code> interrompt toute requête prenant plus de 10 secondes.
<code>max_temp_disk_usage</code>	Une clause facultative qui limite le pourcentage d'espace disque temporaire utilisé par une requête. Si une requête dépasse cette limite de disque temporaire, le système l'interrompt. Par exemple, <code>max_temp_disk_usage = 10</code> interrompt toute requête utilisant plus de 10 % de l'espace disque temporaire.
<code>cache_max_time</code>	Une clause facultative qui détermine si une requête spécifiée utilise les résultats du cache plutôt que d'exécuter la requête. S'il existe un résultat mis en cache avec le même texte de requête, exécuté il y a moins de <code>cache_max_time</code> secondes, la base de données utilise ce résultat mis en cache. La valeur par défaut est 0 et, par conséquent, la base de données ne renvoie aucun résultat mis en cache par défaut. Le cache prend en compte tous les SQL Nodes et, si un résultat potentiellement mis en cache est disponible uniquement sur un autre SQL Node, la base de données redirige la requête vers ce nœud. Si vous spécifiez l'attribut force sur la connexion, ce qui désactive l'équilibrage de charge et la redirection, la base de données prend en compte uniquement les résultats mis en cache sur le nœud actuel.
<code>cache_max_bytes</code>	Une clause facultative qui contrôle si la base de données stocke les résultats d'une requête spécifiée dans le cache. Le système stocke dans le cache les résultats de toutes les requêtes exécutées à l'aide d'une classe de service spécifiant cet attribut si la taille du résultat est inférieure à cette valeur. Vous pouvez déterminer la taille d'un ensemble de résultats en octets en interrogeant le champ <code>bytes_returned</code> de la table virtuelle <code>completed_queries</code>. La valeur par défaut est 0 et, par conséquent, la base de données ne met aucun résultat en cache par défaut. La base de données met les résultats en cache dans la mémoire des SQL Nodes. Ces résultats ne sont pas mis en cache si la mémoire disponible est insuffisante.
<code>service_class</code>	Une clause facultative à la fin d'une requête qui définit une classe de service spécifique pour exécuter cette requête. Pour plus de détails, consultez CRÉER UNE CLASSE DE SERVICE.

Exemple



SQL

```
SELECT * FROM sys.dummy10
USING
  SCHEDULING_PRIORITY = 1.0
  MAX_ROWS_RETURNED = 10
  MAX_ELAPSED_TIME = 10
  MAX_TEMP_DISK_USAGE = 10;
```

Sortie

SQL

```
c1
-----
1
2
3
4
5
6
7
8
9
10
```

TRACE

Une clause facultative qui exécute la requête, mais ignore l'ensemble de résultats d'origine. À la place, `TRACE` renvoie un ensemble de résultats de données de traçage qui décrivent l'exécution de la requête.

Ajoutez l'instruction SQL `TRACE` à la fin d'une requête. L'instruction peut inclure des paramètres facultatifs pour contrôler sa fréquence et son niveau de détail.

Pour comprendre les résultats de cette instruction, consultez [Résultats TRACE](#).

Les requêtes de trace ont le même effet sur la charge de travail qu'une requête exécutée sans la clause.

Contactez le support Ocint pour obtenir des conseils sur l'utilisation de l'instruction SQL TRACE.

Syntaxe

SQL

```
<select_query> TRACE
  [ FREQUENCY frequency_int ]
  [ RESOLUTION resolution_int ]
```

Paramètres



Paramètre	Description
<code><select_query></code>	Une requête <code>SELECT</code> valide.
<code>frequency_int</code>	Contrôle la fréquence à laquelle la base de données échantillonne les événements de trace pendant l'exécution de la requête, en millisecondes. Si non spécifié, la valeur par défaut est de 500 millisecondes.
<code>resolution_int</code>	Détermine le niveau de détail capturé pour une trace ou un événement. Si la résolution d'une trace est supérieure à la résolution d'un événement, la base de données enregistre l'événement. Si non spécifié, la valeur par défaut est de 100 millisecondes.

Résultats TRACE

Chaque ligne de la sortie de trace fournit des informations sur ce qu'une instance d'opérateur a fait pendant la période comprise entre l'échantillon de trace précédent et l'échantillon le plus récent.

Colonne	Description
<code>plan_parent_id</code>	Le UUID du parent de l'instance d'opérateur dans le plan de requête. Si un opérateur possède plusieurs parents, ce champ spécifie le UUID de l'un des parents.
<code>plan_op_id</code>	Le UUID de l'opérateur dans le plan de requête.
<code>operator_type</code>	Le type d'opérateur que la base de données utilise pour échantillonner les événements de trace.
<code>node_id</code>	Le UUID du nœud où cet opérateur s'exécute.
<code>code_id</code>	L'identifiant du cœur VM où cette instance d'opérateur s'exécute. Les identifiants de cœur sont uniques pour chaque nœud.
<code>op_id</code>	L'identifiant de l'instance d'opérateur. Les identifiants sont uniques pour chaque nœud.
<code>temps</code>	Le temps en millisecondes auquel la base de données collecte les informations de l'événement de trace par rapport au début de la requête.
<code>groupe</code>	Regroupe les événements de trace associés.
<code>échantillon</code>	L'échantillon des données d'événement de trace. Pour les valeurs possibles, consultez le tableau suivant.
<code>valeur</code>	La valeur des données d'événement de trace.

Colonnes Sample et Value

Les colonnes `échantillon` et `valeur` décrivent ce qui s'est produit pendant une période de trace donnée.



Exemple	Description
ROWS_IN	Le nombre de lignes qu'une instance d'opérateur a lues à partir de ses opérateurs enfants.
ROWS_OUT	Le nombre de lignes renvoyées par l'instance d'opérateur.
BLOOM_FILTERED_ROWS	Le nombre de lignes que cette instance d'opérateur ignore à l'aide de filtres Bloom.
SCHEDULE_CYCLE	Le nombre de cycles dans la planification pour cette instance d'opérateur.
OOM_CYCLE	Le nombre de cycles de mémoire insuffisante dans la planification pour cet opérateur.
INITIALISER	Le moment où la base de données initialise l'opérateur.
FINALISER	Le moment où la base de données finalise l'opérateur.

Exemple

SQL

```
SELECT * FROM sys.dummy10 TRACE;
```

Sortie

plan_parent_id	plan_op_id	operator_type	node_id	cor_e_id	op_id	ti_m_e	group	sample	value
77041d2b-f7ca-49d5-a3f5-4b75b0782ed1	498690cf-4a9b-4c1f-affa-417f31fddc2d	RENAME_OPERATOR	08ca7b05-1e4d-455f-9f46-2174ed048d33	4	8441	0	xg::db::vm::operators::operatorTraceEvents	ROWS_IN	1
77041d2b-f7ca-49d5-a3f5-4b75b0782ed1	498690cf-4a9b-4c1f-affa-417f31fddc2d	RENAME_OPERATOR	08ca7b05-1e4d-455f-9f46-2174ed048d33	4	8441	0	xg::db::vm::operators::operatorTraceEvents	ROWS_OUT	1
77041d2b-f7ca-49d5-a3f5-4b75b0782ed1	498690cf-4a9b-4c1f-affa-417f31fddc2d	RENAME_OPERATOR	08ca7b05-1e4d-455f-9f46-2174ed048d33	4	8441	0	xg::db::vm::operators::operatorTraceEvents	BLOOM_FILTERED_ROWS	0

TAG

Clause facultative qui ajoute une ou plusieurs balises à la requête SQL. Vous pouvez également ajouter des balises à une expression de table commune dans la clause `AVEC`. Après avoir défini des balises, vous pouvez les trouver dans les tables du catalogue système `sys.queries` et `sys.completed_queries`.

Syntaxe

SQL

```
<select_query> <tag> [ ... ]

<tag> ::= TAG <tag_identifieur>
```



Paramètres

Paramètre	Description
<code><select_quer</code> <code>y></code>	Une requête <code>SELECT</code> valide.
<code>identifiant_</code> <code>tag</code>	Identifiant de la balise pour la requête. Placez l'identifiant entre guillemets doubles s'il contient des caractères spéciaux, comme des espaces.

Exemples

Requête SQL avec une balise

Sélectionnez le nombre d'une table avec une ligne et balisez la requête avec le nom `count1` .

SQL

```
SELECT COUNT(*) FROM sys.dummy1 TAG count1;
```

Requête SQL avec plusieurs balises

Sélectionnez le modèle d'appareil et balisez cette requête avec les deux noms `device_model` et `téléphones` .

SQL

```
SELECT device_model
FROM system.adtech_flat
LIMIT 1
TAG device_model
TAG phones;
```

Expression de table commune avec une balise

Définissez une expression de table commune avec la balise `average_budget` . Vous pouvez également ajouter une autre balise pour l'ensemble de la requête `overall_budget` .

SQL

```
WITH avg_budget_table (average_budget) AS (
  SELECT AVG(budget)
  FROM movies
  TAG average_budget
)
SELECT title,
       budget,
       revenue
FROM movies,
     avg_budget_table
WHERE budget > avg_budget_table.average_budget
TAG overall_budget;
```



Littéraux de chaîne et séquences d'échappement

Tous les littéraux de chaîne dans les instructions SQL doivent être placés entre guillemets simples. Pour utiliser un guillemet simple dans une chaîne, vous pouvez utiliser un autre guillemet simple comme échappement, `' '`.

Pour les autres séquences d'échappement, incluez un caractère `\` avant le littéral de chaîne, c'est-à-dire avant le guillemet simple ouvrant. Cela indique au système de reconnaître les séquences d'échappement dans le littéral de chaîne. Cela signifie :

- Tous les caractères `\` simples dans la chaîne s'échappent maintenant eux-mêmes.
- Tout caractère suivant le `\` est également échappé s'il correspond à une séquence d'échappement (voir le tableau).

Si vous utilisez des séquences d'échappement, préparez-vous à devoir modifier les chaînes qui incluent `\`, comme les chemins de répertoire.

Séquences d'échappement prises en charge

Séquence d'échappement	Description
<code>' '</code>	Guillemet simple
<code>\"</code>	Guillemet double
<code>\n</code>	Caractère de nouvelle ligne
<code>\r</code>	Caractère de retour chariot
<code>\f</code>	Caractère de saut de page (c'est-à-dire saut de page)
<code>\b</code>	Retour arrière
<code>\\</code>	Barre oblique inverse
<code>\t</code>	Tabulation

Exemples

Ces exemples montrent comment le système interprète les chaînes avec et sans séquences d'échappement.

Chaîne sans séquence d'échappement

Cet exemple sélectionne le littéral de chaîne simple `'\my\directory\path'` qui n'utilise pas de séquences d'échappement.

SQL

```
SELECT '\my\directory\path';
```

Sortie : `\my\directory\path`

Chaîne avec une séquence d'échappement

Cet exemple sélectionne le littéral de chaîne simple `'\\my\\directory\\path'` et utilise une séquence d'échappement. Le résultat omet les barres obliques inverses.



SQL

```
SELECT e'\my\directory\path';
```

Sortie : mydirectorypath

Chaîne avec une séquence d'échappement pour conserver les barres obliques inverses

Cet exemple utilise la chaîne `'\\my\\directory\\path'` avec une séquence d'échappement pour inclure des barres obliques inverses dans le résultat.

SQL

```
SELECT e'\\my\\directory\\path';
```

Sortie : \my\directory\path

Séquences d'échappement avec des expressions régulières

Soyez prudent lorsque vous utilisez des séquences d'échappement avec des chaînes qui utilisent également des expressions régulières. Les séquences d'échappement dans la syntaxe SQL remplacent les séquences d'échappement des expressions régulières.

Dans cet exemple, la fonction [REGEXP_SUBSTR](#) utilise l'expression régulière `\w+` pour correspondre à n'importe quels caractères de mot.

SQL

```
SELECT REGEXP_SUBSTR(  
    'abcdefghijklmnopqrstuvwxy',  
    '\w+'  
);
```

Sortie : abcdefghijklmnopqrstuvwxy

La fonction se comporte différemment si l'expression régulière est une séquence d'échappement, car elle remplace le caractère `\`. L'expression régulière recherche uniquement le caractère `w`.

SQL

```
SELECT REGEXP_SUBSTR(  
    'abcdefghijklmnopqrstuvwxy',  
    e'\w+'  
);
```

Sortie : w



Liens connexes

[Référence SQL](#)

[Référence des instructions Data Control Language \(DCL\)](#)

[Référence des instructions Data Definition Language \(DDL\)](#)



14. Tally Platform API



14.1. member

14.1.1. Find Members / Verify Member

GET

<https://api.b.preview.loyaltyservices.io/api/v3/members>

Try it ▶

Search for members by name, email, phone, or other demographic attributes.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 2 ▾

Required Scopes: api:member:verify, api:member:search

Authorize

Use Your Own Token

Query Parameters

> memberIdentifier string

> memberIds integer[]

> firstName string

> middleName string

> lastName string

> includeReversedName boolean

> statusCode string

> company string

> accountType string

> tierCode string

> email string

> phoneNumber string

> city string

> stateCode string

0

+



- > **zip** `string`
- > **countryCode** `string`
- > **aliasIdentifier** `string`
- > **aliasType** `string`
- > **testAccount** `boolean`
- > **includeAddresses** `boolean`
- > **includeEmails** `boolean`
- > **includePhones** `boolean`
- > **includePaymentCardAccounts** `boolean`
- > **includeAliases** `boolean`
- > **includeCurrencies** `boolean`
- > **includePreferences** `boolean`
- > **includeAuthorizedMembers** `boolean`
- > **includeMemberLists** `boolean`
- > **includeOnlyDisplayableLists** `boolean`
- > **includeInactive** `boolean`
- > **matchMode** `string`

Responses

▼ ● 200 Success



- ▼ MemberProfile[]
- > memberIdentifier string
 - > accountNumber string
 - > mergeSurvivorMemberIdentifier string
 - > mergeSurvivorAccountNumber string
 - > accountTypeCode string
 - > title string
 - > firstName string
 - > middleName string
 - > lastName string
 - > suffix string
 - > birthDate string<date>
 - > enrollmentCode string
 - > enrollmentDateTime string<date-time>
 - > enrollmentDeviceTypeCode string
 - > enrollmentLocation string
 - > enrollmentEmployee string
 - > enrollmentSourceCode string
 - > accountStatusCode string
 - > accountStatusCodeDescription string
 - > accountStatusUpdateDateTime string<date-time>
 - > accountStatusUpdateUser string
 - > closedReasonCode string
 - > closedStatusComment string
 - > closedDate string<date>
 - > companyName string
 - > genderCode string
 - > language string
 - > testAccount boolean



- > scrubbed boolean
- > onlineActivationDateTime string<date-time>
- > updateDateTime string<date-time>
- > asciiFirstName string
- > asciiMiddleName string
- > asciiNickName string
- > asciiLastName string
- > asciiSuffix string
- > asciiTitle string
- > asciiLanguage string
- > closeable Closeable
- > currentTier MemberTier
- > addresses MemberAddress[]
- > aliases MemberAlias[]
- > currencies MemberCurrency[]
- > emails MemberEmail[]
- > phones MemberPhone[]
- > preferences MemberPreference[]
- > paymentCardAccounts MemberPaymentCardAccount[]
- > authorizedMembers AuthorizedMember[]
- > lists ListMember[]
- > LinkedAdminAccounts LinkedAccount[]
- > linkedAccounts LinkedAccount[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members?includeReversedName=boolean&includeOnlyDisplayableLists=boolean&  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "memberIdentifier": "",
    "accountNumber": "",
    "mergeSurvivorMemberIdentifier": "",
    "mergeSurvivorAccountNumber": "",
    "accountTypeCode": "",
    "title": "",
    "firstName": "",
    "middleName": "",
    "lastName": "",
    "suffix": "",
    "birthDate": "",
    "enrollmentCode": "",
    "enrollmentDateTime": "",
    "enrollmentDeviceTypeCode": "",
    "enrollmentLocation": "",
    "enrollmentEmployee": "",
    "enrollmentSourceCode": "",
    "accountStatusCode": "",
    "accountStatusCodeDescription": "",
    "accountStatusUpdateDateTime": "",
    "accountStatusUpdateUser": "",
    "closedReasonCode": "",
    "closedStatusComment": "",
    "closedDate": "",
    "companyName": "",
    "genderCode": "",
    "language": "",
    "testAccount": false,
    "scrubbed": false,
    "onlineActivationDateTime": "",
    "updateDateTime": "",
    "asciiFirstName": "",
    "asciiMiddleName": "",
    "asciiNickName": "",
    "asciiLastName": "",
    "asciiSuffix": "",
    "asciiTitle": "",
    "asciiLanguage": "",
    "closeable": {
      "closeable": false,
      "code": "12001",
      "reason": "Member has negative point balance and cannot be closed or merged."
    },
    "currentTier": {
      "tierCode": "B",
      "description": "Base",
      "rank": 10,
      "typeCode": "Earned",
      "startDate": "2020-01-01",
      "endDate": "2020-12-31",
      "grandFathered": false,
      "earned": true,
      "lifetime": false,
    }
  }
]
```



```
"rolloverNights": 10,
"assignmentReason": "bonus",
"assignmentReasonObjectExternalReference": "",
"assignmentReasonObjectDescription": "Adjustment Bonus 1000 Points",
"createdBy": "",
"sourceCouponUse": {
  "couponCode": "113456816",
  "offerCode": "Q1UP",
  "offerName": "Q1 upgrades",
  "startDate": "2025-01-15",
  "endDate": "2026-01-15"
}
},
"addresses": [
  {
    "memberAddressId": "",
    "address1": "",
    "address2": "",
    "address3": "",
    "city": "",
    "stateCode": "",
    "zip": "",
    "typeCode": "",
    "countryCode": "",
    "companyName": "",
    "asciiAddress1": "",
    "asciiAddress2": "",
    "asciiAddress3": "",
    "asciiCity": "",
    "asciiStateCode": "",
    "asciiZip": "",
    "asciiCountryCode": "",
    "order": "",
    "active": false,
    "validated": false,
    "statusCode": "",
    "updateDateTime": ""
  }
],
"aliases": [
  {
    "memberAliasId": 12345,
    "typeCode": "Email",
    "alias": "A123456",
    "mergeMemberIdentifier": "",
    "startDate": "",
    "endDate": "",
    "partnerProgramCode": "",
    "partnerProgramName": "",
    "statusCode": "",
    "updateDateTime": ""
  }
],
"currencies": [
  {
    "typeCode": "POINTS",
    "typeCodeDescription": "Points",
    "categoryCode": "P",
    "categoryCodeDescription": "Loyalty",
```




```
"expireType": "LimitedTime",
"expireTypeDescription": "Standard Expiration",
"description": "",
"available": 675,
"earned": 1000,
"redeemed": 300,
"expired": 25,
"changeDate": "2023-05-11",
"pointForfeitureDate": "2023-05-11",
"primaryCurrency": false
}
],
"emails": [
{
"memberEmailId": "",
"email": "email@email.com",
"typeCode": "",
"order": 1,
"validated": false,
"statusCode": "",
"updateDateTime": ""
}
],
"phones": [
{
"memberPhoneId": "",
"phoneNumber": "1 414 111-1111",
"phoneExtension": "",
"rawPhoneNumber": "",
"carrier": "",
"typeCode": "Mobile",
"smsCapable": false,
"validated": false,
"statusCode": "A",
"order": 1,
"updateDateTime": ""
}
],
"preferences": [
{
"memberPreferenceId": "",
"preferenceRuleCode": "",
"preferenceRuleDescription": "",
"preferenceRuleDetailCode": "",
"preferenceRuleDetailDescription": "",
"preferenceTypeCode": "",
"preferenceDataType": "",
"value": "",
"description": "",
"updateDateTime": "",
"updateUserId": "",
"updateUserName": ""
}
],
"paymentCardAccounts": [
{
"memberPaymentCardAccountId": "",
"typeCode": "",
"paymentCardAccountNumber": "",
```



```
"enrollmentDate": "",  
"lastOpenDate": "",  
"lastClosedDate": "",  
"firstUseDate": "",  
"accountStatusDate": "",  
"productType": "",  
"sourceCode": ""
```



14.1.2. Enroll Member

POST

https://api.b.preview.loyaltyservices.io/api/v3/members

Try it ▶

Enroll a new member in the program

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member:modify

Authorize

Use Your Own Token

Query Parameters

> migrate boolean

> async boolean



Body Parameters

body MemberEnrollRequest *

Description
Member Enrollment Request Object

> memberIdentifier string	
> accountNumber string	
> accountTypeCode string	MBR
> title string	
> firstName string	
> middleName string	
> lastName string	
> suffix string	
> birthDate string<date>	mm/dd/yyyy
> enrollmentCode string	
> enrollmentDateTime string<date-time>	mm/dd/yyyy, --:-- --
> enrollmentDeviceTypeCode string	
> enrollmentLocation string	
> enrollmentEmployee string	
> enrollmentSourceCode string	
> accountStatusCode string	
> companyName string	
> genderCode string	
> language string	
> onlineActivationDateTime string<date-time>	mm/dd/yyyy, --:-- --
> testAccount boolean	☐
> asciiFirstName string	
> asciiMiddleName string	
> asciiNickName string	
> asciiLastName string	



- > **asciiSuffix** `string`
- > **asciiTitle** `string`
- > **asciiLanguage** `string`
- > **linkedAccounts** `LinkedAccount[]`
- > **addresses** `MemberAddress[]`
- > **emails** `MemberEmail[]`
- > **phones** `MemberPhone[]`
- > **preferences** `MemberPreference[]`
- > **paymentCards** `MemberPaymentCardAccount[]`
- > **aliases** `MemberAlias[]`
- > **lists** `ListMember[]`
- > **referralType** `string`
- > **referralValue** `string`
- > **closedReasonCode** `string`

Responses

▼ ● 200 Success

▼ MemberEnrollResponse

- > **memberIdentifier** `string`
- > **accountNumber** `string`
- > **linkedAccount** `LinkedAccount[]`
- > **addresses** `MemberAddress[]`
- > **emails** `MemberEmail[]`
- > **phones** `MemberPhone[]`
- > **preferences** `MemberPreference[]`
- > **aliases** `MemberAlias[]`
- > **paymentCards** `MemberPaymentCardAccount[]`
- > **currentTier** `MemberTier`
- > **lists** `ListMember[]`

▼ ● 202 Accepted



▼ AsyncTask

- > asyncTaskId integer<int64>
- > taskTypeCode string
- > taskCode string
- > requestDate string<date-time>
- > requestData object
- > statusCode string
- > responseDate string<date-time>
- > responseData object
- > fileName string
- > lineNumber integer
- > fileData string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members?migrate=boolean&async=boolean' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data-raw '{
    "accountTypeCode": "MBR",
    "linkedAccounts": [
      {
        "memberIdentifier": "500000003B",
        "linkedType": "Corp"
      }
    ],
    "addresses": [
      {
        "countryCode": ""
      }
    ],
    "emails": [
      {
        "email": "email@email.com"
      }
    ],
    "phones": [
      {
        "phoneNumber": "1 414 111-1111"
      }
    ],
    "preferences": [
      {
        "preferenceRuleCode": "",
        "preferenceRuleDetailCode": ""
      }
    ],
    "paymentCards": [
      {
        "typeCode": "",
        "paymentCardAccountNumber": ""
      }
    ],
    "aliases": [
      {
        "typeCode": "Email",
        "alias": "A123456",
        "statusCode": "Active"
      }
    ]
  }'
```



Responses

● 200



```
// Success
{
  "memberIdentifier": "",
  "accountNumber": "",
  "linkedAccount": [
    {
      "memberIdentifier": "500000003B",
      "accountNumber": "500000003B",
      "firstName": "",
      "lastName": "",
      "companyName": "",
      "linkedType": "Corp",
      "administrator": true
    }
  ],
  "addresses": [
    {
      "memberAddressId": "",
      "address1": "",
      "address2": "",
      "address3": "",
      "city": "",
      "stateCode": "",
      "zip": "",
      "typeCode": "",
      "countryCode": "",
      "companyName": "",
      "asciiAddress1": "",
      "asciiAddress2": "",
      "asciiAddress3": "",
      "asciiCity": "",
      "asciiStateCode": "",
      "asciiZip": "",
      "asciiCountryCode": "",
      "order": "",
      "active": false,
      "validated": false,
      "statusCode": "",
      "updateDateTime": ""
    }
  ],
  "emails": [
    {
      "memberEmailId": "",
      "email": "email@email.com",
      "typeCode": "",
      "order": 1,
      "validated": false,
      "statusCode": "",
      "updateDateTime": ""
    }
  ],
  "phones": [
    {
      "memberPhoneId": "",
      "phoneNumber": "1 414 111-1111",

```



```
"phoneExtension": "",
"rawPhoneNumber": "",
"carrier": "",
"typeCode": "Mobile",
"smsCapable": false,
"validated": false,
"statusCode": "A",
"order": 1,
"updateDateTime": ""
}
],
"preferences": [
{
"memberPreferenceId": "",
"preferenceRuleCode": "",
"preferenceRuleDescription": "",
"preferenceRuleDetailCode": "",
"preferenceRuleDetailDescription": "",
"preferenceTypeCode": "",
"preferenceDataType": "",
"value": "",
"description": "",
"updateDateTime": "",
"updateUserId": "",
"updateUserName": ""
}
],
"aliases": [
{
"memberAliasId": 12345,
"typeCode": "Email",
"alias": "A123456",
"mergeMemberIdentifier": "",
"startDate": "",
"endDate": "",
"partnerProgramCode": "",
"partnerProgramName": "",
"statusCode": "",
"updateDateTime": ""
}
],
"paymentCards": [
{
"memberPaymentCardAccountId": "",
```



14.1.3. Get Member Profile

GET

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}>

Try it ▶

Retrieve a specific member's profile by their unique account identifier.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> includeAddresses boolean

> includeAliases boolean

> includeCurrencies boolean

> includeEmails boolean

> includePhones boolean

> includePreferences boolean

> includePaymentCardAccounts boolean

> includeAuthorizedMembers boolean

> includeMemberLists boolean

> includeOnlyDisplayableLists boolean

> includeInactive boolean

> includeLinkedAccounts boolean

☐

☐

☐

☐

☐

☐

☐

☐

☐

☐

☐

☐

Responses

▼ ● 200 Success



- ▼ MemberProfile
- > memberIdentifier string
 - > accountNumber string
 - > mergeSurvivorMemberIdentifier string
 - > mergeSurvivorAccountNumber string
 - > accountTypeCode string
 - > title string
 - > firstName string
 - > middleName string
 - > lastName string
 - > suffix string
 - > birthDate string<date>
 - > enrollmentCode string
 - > enrollmentDateTime string<date-time>
 - > enrollmentDeviceTypeCode string
 - > enrollmentLocation string
 - > enrollmentEmployee string
 - > enrollmentSourceCode string
 - > accountStatusCode string
 - > accountStatusCodeDescription string
 - > accountStatusUpdateDateTime string<date-time>
 - > accountStatusUpdateUser string
 - > closedReasonCode string
 - > closedStatusComment string
 - > closedDate string<date>
 - > companyName string
 - > genderCode string
 - > language string
 - > testAccount boolean



- > scrubbed boolean
- > onlineActivationDateTime string<date-time>
- > updateDateTime string<date-time>
- > asciiFirstName string
- > asciiMiddleName string
- > asciiNickName string
- > asciiLastName string
- > asciiSuffix string
- > asciiTitle string
- > asciiLanguage string
- > closeable Closeable
- > currentTier MemberTier
- > addresses MemberAddress[]
- > aliases MemberAlias[]
- > currencies MemberCurrency[]
- > emails MemberEmail[]
- > phones MemberPhone[]
- > preferences MemberPreference[]
- > paymentCardAccounts MemberPaymentCardAccount[]
- > authorizedMembers AuthorizedMember[]
- > lists ListMember[]
- > LinkedAdminAccounts LinkedAccount[]
- > linkedAccounts LinkedAccount[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}?includeOnlyDisplayableLists=boolean' \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "memberIdentifier": "",
  "accountNumber": "",
  "mergeSurvivorMemberIdentifier": "",
  "mergeSurvivorAccountNumber": "",
  "accountTypeCode": "",
  "title": "",
  "firstName": "",
  "middleName": "",
  "lastName": "",
  "suffix": "",
  "birthDate": "",
  "enrollmentCode": "",
  "enrollmentDateTime": "",
  "enrollmentDeviceTypeCode": "",
  "enrollmentLocation": "",
  "enrollmentEmployee": "",
  "enrollmentSourceCode": "",
  "accountStatusCode": "",
  "accountStatusCodeDescription": "",
  "accountStatusUpdateDateTime": "",
  "accountStatusUpdateUser": "",
  "closedReasonCode": "",
  "closedStatusComment": "",
  "closedDate": "",
  "companyName": "",
  "genderCode": "",
  "language": "",
  "testAccount": false,
  "scrubbed": false,
  "onlineActivationDateTime": "",
  "updateDateTime": "",
  "asciiFirstName": "",
  "asciiMiddleName": "",
  "asciiNickName": "",
  "asciiLastName": "",
  "asciiSuffix": "",
  "asciiTitle": "",
  "asciiLanguage": "",
  "closeable": {
    "closeable": false,
    "code": "12001",
    "reason": "Member has negative point balance and cannot be closed or merged."
  },
  "currentTier": {
    "tierCode": "B",
    "description": "Base",
    "rank": 10,
    "typeCode": "Earned",
    "startDate": "2020-01-01",
    "endDate": "2020-12-31",
    "grandFathered": false,
    "earned": true,
    "lifetime": false,
    "rolloverNights": 10,
  }
}
```




```
"assignmentReason": "bonus",
"assignmentReasonObjectExternalReference": "",
"assignmentReasonObjectDescription": "Adjustment Bonus 1000 Points",
"createdBy": "",
"sourceCouponUse": {
  "couponCode": "113456816",
  "offerCode": "Q1UP",
  "offerName": "Q1 upgrades",
  "startDate": "2025-01-15",
  "endDate": "2026-01-15"
}
},
"addresses": [
{
  "memberAddressId": "",
  "address1": "",
  "address2": "",
  "address3": "",
  "city": "",
  "stateCode": "",
  "zip": "",
  "typeCode": "",
  "countryCode": "",
  "companyName": "",
  "asciiAddress1": "",
  "asciiAddress2": "",
  "asciiAddress3": "",
  "asciiCity": "",
  "asciiStateCode": "",
  "asciiZip": "",
  "asciiCountryCode": "",
  "order": "",
  "active": false,
  "validated": false,
  "statusCode": "",
  "updateDateTime": ""
}
],
"aliases": [
{
  "memberAliasId": 12345,
  "typeCode": "Email",
  "alias": "A123456",
  "mergeMemberIdentifier": "",
  "startDate": "",
  "endDate": "",
  "partnerProgramCode": "",
  "partnerProgramName": "",
  "statusCode": "",
  "updateDateTime": ""
}
],
"currencies": [
{
  "typeCode": "POINTS",
  "typeCodeDescription": "Points",
  "categoryCode": "P",
  "categoryCodeDescription": "Loyalty",
  "expireType": "LimitedTime",
```



```
"expireTypeDescription": "Standard Expiration",
"description": "",
"available": 675,
"earned": 1000,
"redeemed": 300,
"expired": 25,
"changeDate": "2023-05-11",
"pointForfeitureDate": "2023-05-11",
"primaryCurrency": false
}
],
"emails": [
{
"memberEmailId": "",
"email": "email@email.com",
"typeCode": "",
"order": 1,
"validated": false,
"statusCode": "",
"updateDateTime": ""
}
],
"phones": [
{
"memberPhoneId": "",
"phoneNumber": "1 414 111-1111",
"phoneExtension": "",
"rawPhoneNumber": "",
"carrier": "",
"typeCode": "Mobile",
"smsCapable": false,
"validated": false,
"statusCode": "A",
"order": 1,
"updateDateTime": ""
}
],
"preferences": [
{
"memberPreferenceId": "",
"preferenceRuleCode": "",
"preferenceRuleDescription": "",
"preferenceRuleDetailCode": "",
"preferenceRuleDetailDescription": "",
"preferenceTypeCode": "",
"preferenceDataType": "",
"value": "",
"description": "",
"updateDateTime": "",
"updateUserId": "",
"updateUserName": ""
}
],
"paymentCardAccounts": [
{
"memberPaymentCardAccountId": "",
"typeCode": "",
"paymentCardAccountNumber": "",
"enrollmentDate": "",
```



```
"lastOpenDate": "",  
"lastClosedDate": "",  
"firstUseDate": "",  
"accountStatusDate": "",  
"productType": ""
```



14.1.4. Update Member Profile

PUT

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}

Try it ▶

Update a member's profile.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> scrub boolean



Body Parameters

▼

body

MemberUpdateRequest

Description

Member Profile Update Request Object

>

title

string

>

firstName

string

>

middleName

string

>

lastName

string

>

suffix

string

>

birthDate

string<date>

>

accountStatusCode

string

>

closedReasonCode

string

>

closedStatusComment

string

>

companyName

string

>

genderCode

string

>

language

string

>

testAccount

boolean

>

asciiFirstName

string

>

asciiMiddleName

string

>

asciiNickName

string

>

asciiLastName

string

>

asciiSuffix

string

>

asciiTitle

string

>

asciiLanguage

string

mm/dd/yyyy

Responses

● 200

Success

▼

● 400

Bad request

▼

ErrorResponse

>

errors

ValidationError[]

● 401

Unauthorized

● 403

Forbidden



● 500 Internal error

Request

Curl

```
curl --request PUT \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}?scrub=true' \  
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.1.5. Partial Update Member Profile

PATCH

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}>

Try it ▶

Partially update a member's profile. Only fields included in the request body are updated. Absent fields are left unchanged. A field explicitly set to null will clear that field on the member record. Requires the "api:member

" scope; calling with "scrub=true" additionally requires the "api:member" scope.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member:modify

Authorize

Use Your Own Token

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 2 ▾

Required Scopes: api:member:modify, api:member:scrub

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> scrub boolean



Body Parameters

▼ **body** MemberPatchRequest *

Description
Member Profile Partial Update Request Object

title	string	nullable	
firstName	string	nullable	
middleName	string	nullable	
lastName	string	nullable	
suffix	string	nullable	
> birthDate	string<date>	nullable	mm/dd/yyyy
accountStatusCode	string	nullable	
closedReasonCode	string	nullable	
closedStatusComment	string	nullable	
companyName	string	nullable	
genderCode	string	nullable	
language	string	nullable	
testAccount	boolean	nullable	☐
asciiFirstName	string	nullable	
asciiMiddleName	string	nullable	
asciiNickName	string	nullable	
asciiLastName	string	nullable	
asciiSuffix	string	nullable	
asciiTitle	string	nullable	
asciiLanguage	string	nullable	

Responses

- 200 Success
- ▼ ● 400 Bad request
 - ▼ ErrorResponse
 - > errors ValidationError[]
- 401 Unauthorized
- 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request PATCH \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}?scrub=boolean' \  
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.1.6. Register for promotion using one-click encrypted tokens.

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/promotion>

Try it ▶

Register for promotion using Token

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:promotion:register

Authorize

Use Your Own Token

Body Parameters

▼ body RegisterForPromotionUsingTokenRequest

Description

Register For Promotion using Token Object

> token string *

Responses

▼ ● 200 Success

▼ RegisterForPromotionResponse

> memberPromotionIds integer[] *

Response Headers

Trace-ID string

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

Response Headers

Trace-ID string

▼ ● 401 Unauthorized

Response Headers

Trace-ID string

▼ ● 403 Forbidden

Response Headers

Trace-ID string



▼ ● 500 Internal error

Response Headers

Trace-ID string

Request

Curl

```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/promotion \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9"  
  }'
```

Responses

● 200

```
// Success  
{  
  "memberPromotionIds": [  
    0  
  ]  
}
```



14.1.7. Update Member's Enrollment Date

PUT

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/member-enrollment-date`

Try it

Update member's enrollment date

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:member:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Body Parameters

body MemberUpdateEnrollmentDateRequest *

> enrollmentDate string<date> *

mm/dd/yyyy



Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/member-enrollment-date \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "enrollmentDate": "2024-05-31"
  }'
```

Responses

● 200

// Success





14.1.8. Evaluate Bonus Eligibility

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/bonuses/evaluate`

Try it ▶

Evaluate whether a member is eligible for a given set of bonuses.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-bonus:read

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** string *

Body Parameters

▾ **body** BonusEligibilityRequest *> **bonuses** string[]

+

Responses

▾ ● 200 Success

▾ BonusEligibilityResponse

> **bonuses** BonusEligibilityItem[]

▾ ● 400 Bad request

▾ ErrorResponse

> **errors** ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/bonuses/evaluate \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "bonuses": [
    {
      "bonusCode": "BONUS26",
      "eligible": true
    }
  ]
}
```



14.2. member-address

14.2.1. Get Member Addresses

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses`

Try it ▶

Retrieve a member's address

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:member-address:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Responses

▼ • 200 Success



▼ MemberAddress[]

- > memberAddressId integer<int64>
- > address1 string
- > address2 string
- > address3 string
- > city string
- > stateCode string
- > zip string
- > typeCode string
- > countryCode string *
- > companyName string
- > asciiAddress1 string
- > asciiAddress2 string
- > asciiAddress3 string
- > asciiCity string
- > asciiStateCode string
- > asciiZip string
- > asciiCountryCode string
- > order integer<int32>
- > active boolean
- > validated boolean
- > statusCode string
- > updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "memberAddressId": "",  
    "address1": "",  
    "address2": "",  
    "address3": "",  
    "city": "",  
    "stateCode": "",  
    "zip": "",  
    "typeCode": "",  
    "countryCode": "",  
    "companyName": "",  
    "asciiAddress1": "",  
    "asciiAddress2": "",  
    "asciiAddress3": "",  
    "asciiCity": "",  
    "asciiStateCode": "",  
    "asciiZip": "",  
    "asciiCountryCode": "",  
    "order": "",  
    "active": false,  
    "validated": false,  
    "statusCode": "",  
    "updateDateTime": ""  
  }  
]
```



14.2.2. Create Member Address

POST

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses

Try it ▶

Create a member address

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-address:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *



Body Parameters

▼

body

MemberAddress

Description

Member Address Object

>

address1

string

>

address2

string

>

address3

string

>

city

string

>

stateCode

string

>

zip

string

>

typeCode

string

>

countryCode

string

*

>

companyName

string

>

asciiAddress1

string

>

asciiAddress2

string

>

asciiAddress3

string

>

asciiCity

string

>

asciiStateCode

string

>

asciiZip

string

>

asciiCountryCode

string

>

order

integer<int32>

Format: int32

>

active

boolean

>

validated

boolean

>

statusCode

string

Responses

▼

● 200

Success



▼ MemberAddress

- > memberAddressId integer<int64>
- > address1 string
- > address2 string
- > address3 string
- > city string
- > stateCode string
- > zip string
- > typeCode string
- > countryCode string *
- > companyName string
- > asciiAddress1 string
- > asciiAddress2 string
- > asciiAddress3 string
- > asciiCity string
- > asciiStateCode string
- > asciiZip string
- > asciiCountryCode string
- > order integer<int32>
- > active boolean
- > validated boolean
- > statusCode string
- > updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "countryCode": ""
  }'
```

Responses

● 200

```
// Success
{
  "memberAddressId": "",
  "address1": "",
  "address2": "",
  "address3": "",
  "city": "",
  "stateCode": "",
  "zip": "",
  "typeCode": "",
  "countryCode": "",
  "companyName": "",
  "asciiAddress1": "",
  "asciiAddress2": "",
  "asciiAddress3": "",
  "asciiCity": "",
  "asciiStateCode": "",
  "asciiZip": "",
  "asciiCountryCode": "",
  "order": "",
  "active": false,
  "validated": false,
  "statusCode": "",
  "updateDateTime": ""
}
```



14.2.3. Get Member Address

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses/{memberAddressId}

Try it ▶

Get a member's address

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-address:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberAddressId integer<int64> *

Format: int64

Responses

▼ ● 200 Success



▼ MemberAddress

> memberAddressId integer<int64>

> address1 string

> address2 string

> address3 string

> city string

> stateCode string

> zip string

> typeCode string

> countryCode string *

> companyName string

> asciiAddress1 string

> asciiAddress2 string

> asciiAddress3 string

> asciiCity string

> asciiStateCode string

> asciiZip string

> asciiCountryCode string

> order integer<int32>

> active boolean

> validated boolean

> statusCode string

> updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses/{memberAddressId} \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "memberAddressId": "",  
  "address1": "",  
  "address2": "",  
  "address3": "",  
  "city": "",  
  "stateCode": "",  
  "zip": "",  
  "typeCode": "",  
  "countryCode": "",  
  "companyName": "",  
  "asciiAddress1": "",  
  "asciiAddress2": "",  
  "asciiAddress3": "",  
  "asciiCity": "",  
  "asciiStateCode": "",  
  "asciiZip": "",  
  "asciiCountryCode": "",  
  "order": "",  
  "active": false,  
  "validated": false,  
  "statusCode": "",  
  "updateDateTime": ""  
}
```



14.2.4. Update Member Address

PUT

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses/{memberAddressId}

Try it ▶

Update a member's address

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-address:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberAddressId integer<int64> *

Format: int64



Body Parameters

body MemberAddress

Description
Member Address Object

- > address1 string
- > address2 string
- > address3 string
- > city string
- > stateCode string
- > zip string
- > typeCode string
- > countryCode string *
- > companyName string
- > asciiAddress1 string
- > asciiAddress2 string
- > asciiAddress3 string
- > asciiCity string
- > asciiStateCode string
- > asciiZip string
- > asciiCountryCode string
- > order integer<int32>
- > active boolean
- > validated boolean
- > statusCode string

Format: int32

Responses

- 200 Success
- ▼ ● 400 Bad request
 - ▼ ErrorResponse
 - > errors ValidationError[]
- 401 Unauthorized
- 403 Forbidden



● 500 Internal error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses/{memberAddressId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "countryCode": ""
  }'
```

Responses

● 200

// Success



14.2.5. Delete Member Address

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses/{memberAddressId}`

Try it ▶

Delete a member's address

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-address:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberAddressId integer<int64> *

Format: int64

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/addresses/{memberAddressId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.3. member-alias

14.3.1. Get Member Aliases

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases

Try it ▶

Retrieve a member's aliases

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-alias:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Responses

▼ ● 200 Success

▼ MemberAlias[]

> memberAliasId integer<int64>

> typeCode string *

> alias string *

> mergeMemberIdentifier string

> startDate string<date>

> endDate string<date>

> partnerProgramCode string

> partnerProgramName string

> statusCode string

> updateDateTime string<date-time>

▼ ● 400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases \  
  --header 'accept: application/json'
```

Responses



```
// Success  
[  
  {  
    "memberAliasId": 12345,  
    "typeCode": "Email",  
    "alias": "A123456",  
    "mergeMemberIdentifier": "",  
    "startDate": "",  
    "endDate": "",  
    "partnerProgramCode": "",  
    "partnerProgramName": "",  
    "statusCode": "",  
    "updateDateTime": ""  
  }  
]
```



14.3.2. Create Member Alias

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases>

Try it ▶

Create a member's alias

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-alias:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Body Parameters

▼ body MemberAlias

Description

Member Alias object

> typeCode string *

> alias string *

> mergeMemberIdentifier string

> startDate string<date>

> endDate string<date>

> partnerProgramCode string

> statusCode string

mm/dd/yyyy



mm/dd/yyyy



Active

Responses

▼ ● 200 Success



▼ MemberAlias

> memberAliasId integer<int64>

> typeCode string *

> alias string *

> mergeMemberIdentifier string

> startDate string<date>

> endDate string<date>

> partnerProgramCode string

> partnerProgramName string

> statusCode string

> updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "typeCode": "Email",
    "alias": "A123456",
    "statusCode": "Active"
  }'
```



Responses

● 200

```
// Success
{
  "memberAliasId": 12345,
  "typeCode": "Email",
  "alias": "A123456",
  "mergeMemberIdentifier": "",
  "startDate": "",
  "endDate": "",
  "partnerProgramCode": "",
  "partnerProgramName": "",
  "statusCode": "",
  "updateDateTime": ""
}
```



14.3.3. Get Member Alias

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases/{memberAliasId}`

Try it ▶

Get a member's alias

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-alias:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberAliasId integer<int64> *

Format: int64

Responses

▼ ● 200 Success

▼ MemberAlias

> memberAliasId integer<int64>

> typeCode string *

> alias string *

> mergeMemberIdentifier string

> startDate string<date>

> endDate string<date>

> partnerProgramCode string

> partnerProgramName string

> statusCode string

> updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases/{memberAliasId} \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "memberAliasId": 12345,  
  "typeCode": "Email",  
  "alias": "A123456",  
  "mergeMemberIdentifier": "",  
  "startDate": "",  
  "endDate": "",  
  "partnerProgramCode": "",  
  "partnerProgramName": "",  
  "statusCode": "",  
  "updateDateTime": ""  
}
```



14.3.4. Update Member Alias

PUT

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases/{memberAliasId}

Try it ▶

Update a member's Alias

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-alias:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberAliasId integer<int64> *

Format: int64

Body Parameters

body MemberAlias

Description
Member Alias Object

> typeCode string *

> alias string *

> mergeMemberIdentifier string

> startDate string<date>

mm/dd/yyyy 📅

> endDate string<date>

mm/dd/yyyy 📅

> partnerProgramCode string

> statusCode string

Active

Responses

- 200 Success
- 400 Bad request
 - ErrorResponse
 - errors ValidationError[]
- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases/{memberAliasId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "typeCode": "Email",
    "alias": "A123456",
    "statusCode": "Active"
  }'
```

Responses

200

```
// Success
```



14.3.5. Delete Member Alias

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases/{memberAliasId}`

Try it ▶

Delete a member's alias

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-alias:modify

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** *string* *

> **memberAliasId** *integer<int64>* *

Format: int64

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** *ValidationError[]*

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/aliases/{memberAliasId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.4. member-coupon

14.4.1. Get Member Coupons

GET

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons>

Try it ▶

Retrieve a member's coupons.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-coupon:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> aggregateStatusCode string

> offerType string

> includeAttributes boolean

> includeUses boolean

> includeOfferAttributes boolean

> includeTransferData boolean

Responses

▼ ● 200 Success



▼ MemberCoupon[]

- > memberCouponId integer<int64>
- > memberIdentifier string
- > coupon Coupon *
- > sourceTypeCode string *
- > sourceReferenceRuleCode string
- > sourceReferenceId integer<int64> *
- > aggregateStatusCode string
- > statusCode string *
- > transferable boolean
- > previouslyTransferred boolean

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "memberCouponId": 147258369,
    "memberIdentifier": "",
    "coupon": {
      "couponId": 147258369,
      "couponRequestId": 147258370,
      "offer": {
        "offerId": 147258369,
        "offerTypeCode": "Discount",
        "code": "ALON2020",
        "name": "Free night",
        "description": "Award double points",
        "featureVersion": 2,
        "statusCode": "Enabled",
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
          {
            "offerAttributeId": 147258369,
            "attribute": {
              "attributeId": 147258369,
              "typeCode": "Validation",
              "statusCode": "Enabled",
              "name": "AccountNumber",
              "description": "Account Number",
              "dataTypeCode": "open"
            },
            "defaultValue": "Standard",
            "requireOnUse": true,
            "requireOnHold": true,
            "includeOnResponse": false,
            "attributeTypeCode": "Validation",
            "operatorCode": "EQ",
            "operand": "GI",
            "smartGroupId": 147258369
          }
        ]
      }
    }
  }
]
```



14.4.2. Get Member Coupon By Coupon Code

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}`

Try it ▶

Retrieve a member's coupon by coupon code.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-coupon:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> couponCode string *

Query Parameters

> includeAttributes boolean

> includeUses boolean

> includeOfferAttributes boolean

> includeTransferData boolean

Responses

✓ ● 200 Success



▼ MemberCoupon

- > memberCouponId integer<int64>
- > memberIdentifier string
- > coupon Coupon *
- > sourceTypeCode string *
- > sourceReferenceRuleCode string
- > sourceReferenceId integer<int64> *
- > aggregateStatusCode string
- > statusCode string *
- > transferable boolean
- > previouslyTransferred boolean

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode} \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "memberCouponId": 147258369,
  "memberIdentifier": "",
  "coupon": {
    "couponId": 147258369,
    "couponRequestId": 147258370,
    "offer": {
      "offerId": 147258369,
      "offerTypeCode": "Discount",
      "code": "ALON2020",
      "name": "Free night",
      "description": "Award double points",
      "featureVersion": 2,
      "statusCode": "Enabled",
      "startDateTime": "",
      "endDateTime": "",
      "offerAttributes": [
        {
          "offerAttributeId": 147258369,
          "attribute": {
            "attributeId": 147258369,
            "typeCode": "Validation",
            "statusCode": "Enabled",
            "name": "AccountNumber",
            "description": "Account Number",
            "dataTypeCode": "open"
          },
          "defaultValue": "Standard",
          "requireOnUse": true,
          "requireOnHold": true,
          "includeOnResponse": false,
          "attributeTypeCode": "Validation",
          "operatorCode": "EQ",
          "operand": "GI",
          "smartGroupId": 147258369
        }
      ]
    }
  }
}
```



14.4.3. Cancel member coupon

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}`

Try it ▶

Cancel a member coupon and its uses

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-coupon-use:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> couponCode string *

Responses

▼ ● 200 Success

▼ MemberCoupon

- > memberCouponId integer<int64>
- > memberIdentifier string
- > coupon Coupon *
- > sourceTypeCode string *
- > sourceReferenceRuleCode string
- > sourceReferenceId integer<int64> *
- > aggregateStatusCode string
- > statusCode string *
- > transferable boolean
- > previouslyTransferred boolean

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request



```
curl --request DELETE \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode} \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "memberCouponId": 147258369,
  "memberIdentifier": "",
  "coupon": {
    "couponId": 147258369,
    "couponRequestId": 147258370,
    "offer": {
      "offerId": 147258369,
      "offerTypeCode": "Discount",
      "code": "ALON2020",
      "name": "Free night",
      "description": "Award double points",
      "featureVersion": 2,
      "statusCode": "Enabled",
      "startDateTime": "",
      "endDateTime": "",
      "offerAttributes": [
        {
          "offerAttributeId": 147258369,
          "attribute": {
            "attributeId": 147258369,
            "typeCode": "Validation",
            "statusCode": "Enabled",
            "name": "AccountNumber",
            "description": "Account Number",
            "dataTypeCode": "open"
          },
          "defaultValue": "Standard",
          "requireOnUse": true,
          "requireOnHold": true,
          "includeOnResponse": false,
          "attributeTypeCode": "Validation",
          "operatorCode": "EQ",
          "operand": "GI",
          "smartGroupId": 147258369
        }
      ]
    }
  }
}
```



14.4.4. Transfer Coupon

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/transfer>

Try it ▶

Transfer a coupon to another member

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-coupon:transfer

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> couponCode string *

Body Parameters

▼ body CouponTransferRequest

Description

Use Coupon Request Object

> recipientAccountNumber string

> recipientFirstName string

> recipientLastName string

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

📄 Curl

```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/transfer \
  --header 'accept: application/json'
```



Responses

● 200

// Success



14.4.5. Post Coupon Use

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/uses>

Try it ▶

Post a coupon use or hold

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-coupon-use:update

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> couponCode string *

Query Parameters

> holdCoupon boolean

☐

> validateOnly boolean

☐

> validateCouponValidToday boolean

☐

> includeCouponReward boolean

☐

Body Parameters

▼ body CouponUseRequest

Description

Use Coupon Request Object

> useDateTime string<date-time>

mm/dd/yyyy, --:-- -- ☒

> alias string

> useCount integer<int32>

Format: int32

> messageFromGiftingMember string

> validationAttributes map<string, string>

> reportingAttributes map<string, string>

> attributes map<string, string>

Responses

▼ ● 200 Success



▼ CouponUseResponse

- > couponUseld integer<int64>
- > countRemaining integer<int64>
- > statusCode string
- > overUse boolean
- > overUseErrorCode string
- > overUserErrorDescription string
- > couponReward CouponUseReward

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

[Curl](#)

```
curl --request POST \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/uses?holdCoupon=true&val  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "couponUseId": "",
  "countRemaining": "",
  "statusCode": "",
  "overUse": false,
  "overUseErrorCode": "",
  "overUserErrorDescription": "",
  "couponReward": {
    "bonuses": [
      {
        "transactionId": "",
        "transactionGroupId": "",
        "transactionGroupReferenceId": "",
        "parentTransactionId": "",
        "linkedTransactionId": "",
        "referenceId": "",
        "memberIdentifier": "",
        "accountNumber": "",
        "memberRedemptionId": "1",
        "transactionDateTime": "",
        "partnerName": "",
        "partnerProgramCode": "",
        "description": "",
        "typeCode": "",
        "typeDescription": "",
        "subTypeCode": "",
        "postDate": "",
        "originationDate": "",
        "retro": false,
        "edited": false,
        "sourceCode": "",
        "statusCode": "",
        "updateDateTime": "",
        "earningRuleDescription": "",
        "customData": {
          "ace01": "",
          "ace01Description": "",
          "ace02": "",
          "ace02Description": "",
          "ace03": "",
          "ace03Description": "",
          "ace04": "",
          "ace04Description": "",
          "ace05": "",
          "ace05Description": "",
          "ace06": "",
          "ace06Description": "",
          "ace07": "",
          "ace07Description": "",
          "ace08": "",
          "ace08Description": "",
          "ace09": "",
          "ace09Description": "",
          "ace10": "",
          "ace10Description": "",
        }
      }
    ]
  }
}
```



```
"ace11": "",
"ace11Description": "",
"ace12": "",
"ace12Description": "",
"ace13": "",
"ace13Description": "",
"ace14": "",
"ace14Description": "",
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
```



```
"checkOutDate": "",
"eligibleStays": "",
"eligibleCappedNights": "",
"eligibleNights": "",
"roomNumber": "",
"roomType": "",
"roomRate": 0,
"roomRateIntl": 0,
"totalRevenueAmount": 0,
"totalRevenueAmountIntl": 0,
"loyaltyEligibleAmount": 0,
"loyaltyEligibleAmountIntl": 0,
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountIntl": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueIntl": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueIntl": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
```



14.4.6. Cancel coupon use(s)

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/uses`

Try it ▶

Cancel coupon uses associated with a specific member coupon

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-coupon-use:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> couponCode string *

Responses

▼ ● 200 Success

▼ MemberCoupon

- > memberCouponId integer<int64>
- > memberIdentifier string
- > coupon Coupon *
- > sourceTypeCode string *
- > sourceReferenceRuleCode string
- > sourceReferenceId integer<int64> *
- > aggregateStatusCode string
- > statusCode string *
- > transferable boolean
- > previouslyTransferred boolean

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request



```
curl --request DELETE \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/uses \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "memberCouponId": 147258369,
  "memberIdentifier": "",
  "coupon": {
    "couponId": 147258369,
    "couponRequestId": 147258370,
    "offer": {
      "offerId": 147258369,
      "offerTypeCode": "Discount",
      "code": "ALON2020",
      "name": "Free night",
      "description": "Award double points",
      "featureVersion": 2,
      "statusCode": "Enabled",
      "startDateTime": "",
      "endDateTime": "",
      "offerAttributes": [
        {
          "offerAttributeId": 147258369,
          "attribute": {
            "attributeId": 147258369,
            "typeCode": "Validation",
            "statusCode": "Enabled",
            "name": "AccountNumber",
            "description": "Account Number",
            "dataTypeCode": "open"
          },
          "defaultValue": "Standard",
          "requireOnUse": true,
          "requireOnHold": true,
          "includeOnResponse": false,
          "attributeTypeCode": "Validation",
          "operatorCode": "EQ",
          "operand": "GI",
          "smartGroupId": 147258369
        }
      ]
    }
  }
}
```



14.4.7. Edit Coupon Use

PUT

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/uses/{couponU

Try it ▶

Update a pre-existing coupon use

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-coupon-use:update

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> couponCode string *

> couponUseld integer<int64> *

Format: int64

Query Parameters

> holdCoupon boolean

> validateCouponValidToday boolean

Body Parameters

body CouponUseRequest

Description

Use Coupon Request Object

> useDateTime string<date-time>

> alias string

> useCount integer<int32>

> messageFromGiftingMember string

> validationAttributes map<string, string>

> reportingAttributes map<string, string>

> attributes map<string, string>

mm/dd/yyyy, --:-- --📅

Format: int32

Responses

200 Success



▼ CouponUseResponse

- > couponUseId integer<int64>
- > countRemaining integer<int64>
- > statusCode string
- > overUse boolean
- > overUseErrorCode string
- > overUserErrorDescription string
- > couponReward CouponUseReward

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/uses/{couponUseId} \
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "couponUseId": "",
  "countRemaining": "",
  "statusCode": "",
  "overUse": false,
  "overUseErrorCode": "",
  "overUserErrorDescription": "",
  "couponReward": {
    "bonuses": [
      {
        "transactionId": "",
        "transactionGroupId": "",
        "transactionGroupReferenceId": "",
        "parentTransactionId": "",
        "linkedTransactionId": "",
        "referenceId": "",
        "memberIdentifier": "",
        "accountNumber": "",
        "memberRedemptionId": "1",
        "transactionDateTime": "",
        "partnerName": "",
        "partnerProgramCode": "",
        "description": "",
        "typeCode": "",
        "typeDescription": "",
        "subTypeCode": "",
        "postDate": "",
        "originationDate": "",
        "retro": false,
        "edited": false,
        "sourceCode": "",
        "statusCode": "",
        "updateDateTime": "",
        "earningRuleDescription": "",
        "customData": {
          "ace01": "",
          "ace01Description": "",
          "ace02": "",
          "ace02Description": "",
          "ace03": "",
          "ace03Description": "",
          "ace04": "",
          "ace04Description": "",
          "ace05": "",
          "ace05Description": "",
          "ace06": "",
          "ace06Description": "",
          "ace07": "",
          "ace07Description": "",
          "ace08": "",
          "ace08Description": "",
          "ace09": "",
          "ace09Description": "",
          "ace10": "",
          "ace10Description": "",
        }
      }
    ]
  }
}
```



```
"ace11": "",
"ace11Description": "",
"ace12": "",
"ace12Description": "",
"ace13": "",
"ace13Description": "",
"ace14": "",
"ace14Description": "",
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
```



```
"checkOutDate": "",
"eligibleStays": "",
"eligibleCappedNights": "",
"eligibleNights": "",
"roomNumber": "",
"roomType": "",
"roomRate": 0,
"roomRateIntl": 0,
"totalRevenueAmount": 0,
"totalRevenueAmountIntl": 0,
"loyaltyEligibleAmount": 0,
"loyaltyEligibleAmountIntl": 0,
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountIntl": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueIntl": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueIntl": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
```



14.4.8. Return Coupon Use

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/uses/{couponU`

Try it ▶

Return an existing coupon use

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-coupon-use:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> couponCode string *

> couponUseId integer<int64> *

Format: int64

Body Parameters

▼ body CouponUseReturnRequest

Description

Coupon Use Return Request

> reportingAttributes map<string, string>

> comments string

Responses

▼ ● 200 Success

▼ CouponUseResponse

> couponUseId integer<int64>

> countRemaining integer<int64>

> statusCode string

> overUse boolean

> overUseErrorCode string

> overUserErrorDescription string

> couponReward CouponUseReward

▼ ● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/coupons/{couponCode}/uses/{couponUseId} \
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "couponUseId": "",
  "countRemaining": "",
  "statusCode": "",
  "overUse": false,
  "overUseErrorCode": "",
  "overUserErrorDescription": "",
  "couponReward": {
    "bonuses": [
      {
        "transactionId": "",
        "transactionGroupId": "",
        "transactionGroupReferenceId": "",
        "parentTransactionId": "",
        "linkedTransactionId": "",
        "referenceId": "",
        "memberIdentifier": "",
        "accountNumber": "",
        "memberRedemptionId": "1",
        "transactionDateTime": "",
        "partnerName": "",
        "partnerProgramCode": "",
        "description": "",
        "typeCode": "",
        "typeDescription": "",
        "subTypeCode": "",
        "postDate": "",
        "originationDate": "",
        "retro": false,
        "edited": false,
        "sourceCode": "",
        "statusCode": "",
        "updateDateTime": "",
        "earningRuleDescription": "",
        "customData": {
          "ace01": "",
          "ace01Description": "",
          "ace02": "",
          "ace02Description": "",
          "ace03": "",
          "ace03Description": "",
          "ace04": "",
          "ace04Description": "",
          "ace05": "",
          "ace05Description": "",
          "ace06": "",
          "ace06Description": "",
          "ace07": "",
          "ace07Description": "",
          "ace08": "",
          "ace08Description": "",
          "ace09": "",
          "ace09Description": "",
          "ace10": "",
          "ace10Description": ""
        }
      }
    ]
  }
}
```



```
"ace11": "",
"ace11Description": "",
"ace12": "",
"ace12Description": "",
"ace13": "",
"ace13Description": "",
"ace14": "",
"ace14Description": "",
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
```



```
"checkOutDate": "",
"eligibleStays": "",
"eligibleCappedNights": "",
"eligibleNights": "",
"roomNumber": "",
"roomType": "",
"roomRate": 0,
"roomRateIntl": 0,
"totalRevenueAmount": 0,
"totalRevenueAmountIntl": 0,
"loyaltyEligibleAmount": 0,
"loyaltyEligibleAmountIntl": 0,
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountIntl": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueIntl": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueIntl": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
```



14.5. member-currency

14.5.1. Get Member Currencies

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/currencies`

Try it ▶

Retrieve a member's currencies

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:member-currency:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Responses

▼ • 200 Success



▼ MemberCurrency[]

- > typeCode string *
- > typeCodeDescription string
- > categoryCode string *
- > categoryCodeDescription string
- > expireType string
- > expireTypeDescription string
- > description string
- > available number *
- > earned number *
- > redeemed number *
- > expired number *
- > changeDate string<date>
- > pointForfeitureDate string<date>
- > primaryCurrency boolean

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/currencies \  
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "typeCode": "POINTS",
    "typeCodeDescription": "Points",
    "categoryCode": "P",
    "categoryCodeDescription": "Loyalty",
    "expireType": "LimitedTime",
    "expireTypeDescription": "Standard Expiration",
    "description": "",
    "available": 675,
    "earned": 1000,
    "redeemed": 300,
    "expired": 25,
    "changeDate": "2023-05-11",
    "pointForfeitureDate": "2023-05-11",
    "primaryCurrency": false
  }
]
```



14.5.2. Adjust member currency

POST

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/currencies

Try it ▶

Add a currency adjustment to the member

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-currency:adjust

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> migrate boolean

> migrateBalance boolean

> async boolean



Body Parameters

▼ **body** AdjustCurrencyRequest

Description
Adjust Currency Request Object

- > **adjustmentRuleCode** string *
- > **amount** number *
- > **comment** string *
- > **externalReference** string
- > **sharedExternalReference** string
- > **reasonCode** string
- > **subReasonCode** string
- > **validateUser** boolean
- > **amountConfirmed** boolean
- > **transactionDate** string<date>
- > **customData** CustomData

mm/dd/yyyy

Responses

▼ ● 200 Success



▼ Transaction

- > transactionId string
- > transactionGroupId string
- > transactionGroupReferenceId string
- > parentTransactionId string
- > linkedTransactionId string
- > referenceId string
- > memberIdentifier string
- > accountNumber string
- > memberRedemptionId string
- > transactionDateTime string<date-time>
- > partnerName string
- > partnerProgramCode string
- > description string
- > typeCode string
- > typeDescription string
- > subTypeCode string
- > postDate string<date>
- > originationDate string<date>
- > retro boolean
- > edited boolean
- > sourceCode string
- > statusCode string
- > updateDateTime string<date-time>
- > earningRuleDescription string
- > customData CustomData
- > spend TransactionSpend
- > earn TransactionEarn
- > redemption TransactionRedemption



- > expiredCurrency TransactionExpiredCurrency
- > hotelTransaction HotelTransaction
- > travelTransaction TravelTransaction
- > transactionReference map<string, object>
- > cancelable boolean
- > editable boolean
- > createUserId integer<int64>
- > createUserName string
- > updateUserId integer<int64>
- > updateUserName string
- > memberCommentId integer<int64>
- > cancellationSubReason string
- > businessEligible boolean
- > businessAccountNumber string
- > awardNumber string
- > baseAccountNumber string
- > transactionAdjustment TransactionAdjustment
- > transactionBonus TransactionBonus
- > transactionPartner TransactionPartner
- > transactionPromotion TransactionPromotion
- > transactionTransfer TransactionTransfer
- > mergeDetails MergeDetail
- > transactionRelationships TransactionRelationship[]

▼ ● 202 Accepted



▼ AsyncTask

```
> asyncTaskId integer<int64>
> taskTypeCode string
> taskCode string
> requestDate string<date-time>
> requestData object
> statusCode string
> responseDate string<date-time>
> responseData object
> fileName string
> lineNumber integer
> fileData string
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/currencies?migrate=boolean&migrateBalance=boo
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "adjustmentRuleCode": "",
    "amount": "",
    "comment": ""
  }'
```



Responses

● 200



```
// Success
{
  "transactionId": "",
  "transactionGroupId": "",
  "transactionGroupReferenceId": "",
  "parentTransactionId": "",
  "linkedTransactionId": "",
  "referenceId": "",
  "memberIdentifier": "",
  "accountNumber": "",
  "memberRedemptionId": "1",
  "transactionDateTime": "",
  "partnerName": "",
  "partnerProgramCode": "",
  "description": "",
  "typeCode": "",
  "typeDescription": "",
  "subTypeCode": "",
  "postDate": "",
  "originationDate": "",
  "retro": false,
  "edited": false,
  "sourceCode": "",
  "statusCode": "",
  "updateDateTime": "",
  "earningRuleDescription": "",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
```



```
    "ace15Description": ""
  },
  "spend": {
    "perItem": 0,
    "spendAmount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "isoCurrencyCode": "",
    "currencyConversionRatio": "",
    "quantity": ""
  },
  "earn": {
    "amount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "earningsLimited": false,
    "tierQualifying": false,
    "transactionCounters": [
      {
        "counterTypeCode": "",
        "counterValue": 0
      }
    ]
  },
  "redemption": {
    "redemptionItemCode": "",
    "currencyRequired": 0,
    "minimumQuantity": "",
    "maximumQuantity": "",
    "redeemIncrement": "",
    "faceValue": 0,
    "amountProvided": false,
    "poolName": "PL-16",
    "lineItems": [
      {
        "description": "",
        "itemCode": "",
        "quantity": "",
        "amount": 0
      }
    ]
  },
  "expiredCurrency": {
    "expiredAmount": 0,
    "expiredDate": "",
    "currencyTypeCode": "",
    "currencyCategoryCode": ""
  },
  "hotelTransaction": {
    "checkInDate": "",
    "checkOutDate": "",
    "eligibleStays": "",
    "eligibleCappedNights": "",
    "eligibleNights": "",
    "roomNumber": "",
    "roomType": "",
    "roomRate": 0,
    "roomRateIntl": 0,
    "totalRevenueAmount": 0,
```



```
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",
```



14.5.3. Get member currency buckets

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/currencies/{currencyTypeCode}`

Try it

Retrieve detailed member currency buckets based on currency type

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:member-currency:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> currencyTypeCode string *

Query Parameters

> filterOnlyAvailable boolean

Responses

✓ 200 Success

MemberCurrencyBucket[]

- > typeCode string
- > categoryCode string
- > available number
- > earned number
- > redeemed number
- > expired number
- > expirationDate string<date>
- > expirationRuleDescription string
- > earnedDate string<date>
- > changeDate string<date>
- > primaryCurrency boolean

✓ 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/currencies/{currencyTypeCode} \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "typeCode": "POINTS",
    "categoryCode": "P",
    "available": 1608,
    "earned": 1608,
    "redeemed": 1000,
    "expired": 0,
    "expirationDate": "2025-01-01",
    "expirationRuleDescription": "Points will expire 4 years after they are earned",
    "earnedDate": "2022-01-01",
    "changeDate": "2023-01-01",
    "primaryCurrency": false
  }
]
```



14.6. member-email

14.6.1. Get Member Emails

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails`

Try it ▶

Retrieve a member's emails

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:member-email:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Responses

✓ ● 200 Success

▾ `MemberEmail[]`

> `memberEmailId` `integer<int64>`

> `email` `string` *

> `typeCode` `string`

> `order` `integer<int32>`

> `validated` `boolean`

> `statusCode` `string`

> `updateDateTime` `string<date-time>`

✓ ● 400 Bad request

▾ `ErrorResponse`

> `errors` `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "memberEmailId": "",  
    "email": "email@email.com",  
    "typeCode": "",  
    "order": 1,  
    "validated": false,  
    "statusCode": "",  
    "updateDateTime": ""  
  }  
]
```



14.6.2. Create Member Email

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails>

Try it

Create a member's email

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:member-email:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Body Parameters

body MemberEmail

Description

Member Email Object

> email string *

> typeCode string

> order integer<int32>

> validated boolean

> statusCode string

Responses

200 Success

MemberEmail

> memberEmailId integer<int64>

> email string *

> typeCode string

> order integer<int32>

> validated boolean

> statusCode string

> updateDateTime string<date-time>



▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data-raw '{  
    "email": "email@email.com"  
  }'
```

Responses

● 200

```
// Success  
{  
  "memberEmailId": "",  
  "email": "email@email.com",  
  "typeCode": "",  
  "order": 1,  
  "validated": false,  
  "statusCode": "",  
  "updateDateTime": ""  
}
```



14.6.3. Get Member Email

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails/{memberEmailId}`

Try it ▶

Get a member's email

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-email:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberEmailId integer<int64> *

Format: int64

Responses

▼ ● 200 Success

▼ MemberEmail

> memberEmailId integer<int64>

> email string *

> typeCode string

> order integer<int32>

> validated boolean

> statusCode string

> updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request

Curl

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails/{memberEmailId} \  
  --header 'accept: application/json'
```

Responses

200

```
// Success  
{  
  "memberEmailId": "",  
  "email": "email@email.com",  
  "typeCode": "",  
  "order": 1,  
  "validated": false,  
  "statusCode": "",  
  "updateDateTime": ""  
}
```



14.6.4. Update Member Email

PUT

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails/{memberEmailId}

Try it ▶

Update a member's email

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-email:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberEmailId integer<int64> *

Format: int64

Body Parameters

body MemberEmail

Description
Member Email Object

> email string *

> typeCode string

> order integer<int32>

> validated boolean

> statusCode string

Format: int32

Responses

- 200 Success
- 400 Bad request
 - ErrorResponse
 - errors ValidationError[]
- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request

Curl

```
curl --request PUT \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails/{memberEmailId} \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data-raw '{  
    "email": "email@email.com"  
  }'
```

Responses

200

```
// Success
```



14.6.5. Delete Member Email

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails/{memberEmailId}`

Try it ▶

Delete a member's email

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-email:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberEmailId integer<int64> *

Format: int64

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/emails/{memberEmailId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.7. member-event

14.7.1. Post a member event

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/events/{eventType}>

Try it ▶

Post a member event, possibly triggering qualifying event rules.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-event:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> eventType string *

Query Parameters

> simulate boolean

Body Parameters

body Event

> eventDateTime string<date-time>

> parameters object

mm/dd/yyyy, --:-- --📅

Responses

- 200 Success

EventResults

> results EventRuleResults[]
- 400 Bad request

ErrorResponse

> errors ValidationError[]
- 401 Unauthorized



- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/events/{eventType}?simulate=boolean' \
  --header 'accept: application/json'
```

Responses



```
// Success
{
  "results": [
    {
      "eventRuleId": "",
      "eventRuleCode": "",
      "results": [
        {
          "resultTypeCode": "Bonus",
          "parameters": {}
        }
      ]
    }
  ]
}
```



14.7.2. Get event types

GET

`https://api.b.preview.loyaltyservices.io/api/v3/event-types`

Try it ▶

Get event types and their capabilities

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:member-event:read`

Authorize

Use Your Own Token

Responses

▼ ● 200 Success

▼ EventType[]

- > eventType string
- > eligibility EventTypeEligibility
- > results string[]
- > contextFields string[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/event-types \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "eventType": "CheckIn",
    "eligibility": {
      "event": false,
      "member": false,
      "accrual": false,
      "context": false
    },
    "results": [
      ""
    ],
    "contextFields": [
      ""
    ]
  }
]
```



14.8. member-list

14.8.1. Get member's lists

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/lists`

Try it ▶

Retrieve the member lists that a member belongs to

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:member-list:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Query Parameters

> `includeInactive` `boolean`

> `includeOnlyDisplayableLists` `boolean`

Responses

✓ ● 200 Success

▾ `ListMember[]`

Description

Represents a member in a member list

> `memberIdentifier` `string`

> `startDate` `string<date>`

> `endDate` `string<date>`

> `memberList` `MemberList`

✓ ● 400 Bad request

▾ `ErrorResponse`

> `errors` `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/lists?includeInactive=boolean&includeOnlyDisp  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "memberIdentifier": "",  
    "startDate": "2020-04-06",  
    "endDate": "2026-04-02",  
    "memberList": {  
      "memberListId": "",  
      "code": "",  
      "name": "",  
      "description": "",  
      "statusCode": "",  
      "typeCode": ""  
    }  
  }  
]
```




14.8.2. Add to member list

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/lists/{memberListId}`

Try it

Include a member in a member list

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-list:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberListId integer<int64> *

Format: int64

Query Parameters

> async boolean

Responses

● 200 Success

▾ ● 202 Accepted

▾ AsyncTask

> asyncTaskId integer<int64>

> taskTypeCode string

> taskCode string

> requestDate string<date-time>

> requestData object

> statusCode string

> responseDate string<date-time>

> responseData object

> fileName string

> lineNumber integer

> fileData string

▾ ● 400 Bad request



▼

 ErrorResponse

>

 errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/lists/{memberListId}?async=boolean' \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
```



14.8.3. Remove from member list

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/lists/{memberListId}`

Try it ▶

Exclude a member from a member list

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-list:modify

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** *string* *> **memberListId** *integer<int64>* *

Format: int64

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** *ValidationError[]*

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/lists/{memberListId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.8.4. Add to member list

POST

<https://api.b.preview.loyaltyservices.io/api/v4/members/{memberIdentifier}/lists/{memberListCode}>

Try it

Include a member in a member list

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:member-list:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberListCode string *

Query Parameters

> async boolean

Body Parameters

body ListMemberRequest

Description

Optional parameters for adding member to list

> startDate string<date>

mm/dd/yyyy



> endDate string<date>

mm/dd/yyyy



Responses

200 Success

ListMember

Description

Represents a member in a member list

> memberIdentifier string

> startDate string<date>

> endDate string<date>

> memberList MemberList

202 Accepted



▼ AsyncTask

> asyncTaskId integer<int64>

> taskTypeCode string

> taskCode string

> requestDate string<date-time>

> requestData object

> statusCode string

> responseDate string<date-time>

> responseData object

> fileName string

> lineNumber integer

> fileData string

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request POST \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v4/members/{memberIdentifier}/lists/{memberListCode}?async=boolean' \  
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "memberIdentifier": "",
  "startDate": "2020-04-06",
  "endDate": "2026-04-02",
  "memberList": {
    "memberListId": "",
    "code": "",
    "name": "",
    "description": "",
    "statusCode": "",
    "typeCode": ""
  }
}
```



14.8.5. Remove from member list

DELETE`https://api.b.preview.loyaltyservices.io/api/v4/members/{memberIdentifier}/lists/{memberListCode}`

Try it

Exclude a member from a member list

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-list:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberListCode string *

Query Parameters

> async boolean

Responses

● 200 Success

▾ ● 202 Accepted

▾ AsyncTask

> asyncTaskId integer<int64>

> taskTypeCode string

> taskCode string

> requestDate string<date-time>

> requestData object

> statusCode string

> responseDate string<date-time>

> responseData object

> fileName string

> lineNumber integer

> fileData string

▾ ● 400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request DELETE \
  --url 'https://api.b.preview.loyaltyservices.io/api/v4/members/{memberIdentifier}/lists/{memberListCode}?async=boolean' \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.8.6. Find member lists

GET

https://api.b.preview.loyaltyservices.io/api/v3/member-lists

Try it ▶

Warning: Deprecated

Retrieve available member lists. DEPRECATED: use findSegments instead.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-list:read

Authorize

Use Your Own Token

Query Parameters

> segmentName string

> segmentCode string

> excludeTypeCode string[]

> statusCode string

> startDate string<date>

> endDate string<date>

> usedInType string

> usedInName string

+

mm/dd/yyyy 

mm/dd/yyyy 

Responses

▼ ● 200 Success



▼ MemberList[]

Description

Defines a list of members

- > **memberListId** integer<int64>
- > **code** string
- > **name** string
- > **description** string
- > **statusCode** string
- > **typeCode** string

▼ ● 400 Bad request

▼ ErrorResponse

- > **errors** ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/member-lists \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "memberListId": "",
    "code": "",
    "name": "",
    "description": "",
    "statusCode": "",
    "typeCode": ""
  }
]
```



14.9. member-payment-card-account

14.9.1. Get Member Payment Card Accounts

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts

Try it ▶

Retrieve a member's payment (credit) card accounts

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-payment-card:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Responses

▼ ● 200 Success



▼ MemberPaymentCardAccount[]

Description

Representation of a member's payment (credit) card. DO NOT ENTER PCI INFORMATION.

> memberPaymentCardAccountId integer<int64>

> typeCode string *

> paymentCardAccountNumber string *

> enrollmentDate string<date>

> lastOpenDate string<date>

> lastClosedDate string<date>

> firstUseDate string<date>

> accountStatusDate string<date>

> productType string

> sourceCode string

> merchantCode string

> channelCode string

> firstName string

> lastName string

> birthDate string<date>

> updateDateTime string<date-time>

> active boolean

> cardComment string

additionalProperties object

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "memberPaymentCardAccountId": "",  
    "typeCode": "",  
    "paymentCardAccountNumber": "",  
    "enrollmentDate": "",  
    "lastOpenDate": "",  
    "lastClosedDate": "",  
    "firstUseDate": "",  
    "accountStatusDate": "",  
    "productType": "",  
    "sourceCode": "",  
    "merchantCode": "",  
    "channelCode": "",  
    "firstName": "",  
    "lastName": "",  
    "birthDate": "",  
    "updateDateTime": "",  
    "active": false,  
    "cardComment": "",  
    "additionalProperties": {}  
  }  
]
```



14.9.2. Create Member Payment Card Account

POST

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts

Try it ▶

Create a member payment card account

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-payment-card:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *



Body Parameters

▼

body

MemberPaymentCardAccount

Description

Member Payment Card Account Object

>

typeCode

string

*

>

paymentCardAccountNumber

string

*

>

enrollmentDate

string<date>

📅

>

lastOpenDate

string<date>

📅

>

lastClosedDate

string<date>

📅

>

firstUseDate

string<date>

📅

>

accountStatusDate

string<date>

📅

>

productType

string

>

sourceCode

string

>

merchantCode

string

>

channelCode

string

>

firstName

string

>

lastName

string

>

birthDate

string<date>

📅

>

cardComment

string

additionalProperties

object

Responses

▼

● 200

Success



▼ MemberPaymentCardAccount

Description

Representation of a member's payment (credit) card. DO NOT ENTER PCI INFORMATION.

```
> memberPaymentCardAccountId integer<int64>
> typeCode string *
> paymentCardAccountNumber string *
> enrollmentDate string<date>
> lastOpenDate string<date>
> lastClosedDate string<date>
> firstUseDate string<date>
> accountStatusDate string<date>
> productType string
> sourceCode string
> merchantCode string
> channelCode string
> firstName string
> lastName string
> birthDate string<date>
> updateDateTime string<date-time>
> active boolean
> cardComment string
additionalProperties object
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "typeCode": "",
    "paymentCardAccountNumber": ""
  }'
```

Responses

● 200

```
// Success
{
  "memberPaymentCardAccountId": "",
  "typeCode": "",
  "paymentCardAccountNumber": "",
  "enrollmentDate": "",
  "lastOpenDate": "",
  "lastClosedDate": "",
  "firstUseDate": "",
  "accountStatusDate": "",
  "productType": "",
  "sourceCode": "",
  "merchantCode": "",
  "channelCode": "",
  "firstName": "",
  "lastName": "",
  "birthDate": "",
  "updateDateTime": "",
  "active": false,
  "cardComment": "",
  "additionalProperties": {}
}
```



14.9.3. Get Member Payment Card Account

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts/{memberPaymentCardAccountId}

Try it ▶

Get a member's payment card account

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-payment-card:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberPaymentCardAccountId integer<int64> *

Format: int64

Responses

▼ ● 200 Success



▼ MemberPaymentCardAccount

Description

Representation of a member's payment (credit) card. DO NOT ENTER PCI INFORMATION.

> memberPaymentCardAccountId integer<int64>

> typeCode string *

> paymentCardAccountNumber string *

> enrollmentDate string<date>

> lastOpenDate string<date>

> lastClosedDate string<date>

> firstUseDate string<date>

> accountStatusDate string<date>

> productType string

> sourceCode string

> merchantCode string

> channelCode string

> firstName string

> lastName string

> birthDate string<date>

> updateDateTime string<date-time>

> active boolean

> cardComment string

additionalProperties object

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts/{memberPaymentCardAccountIdentifier} \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "memberPaymentCardAccountId": "",  
  "typeCode": "",  
  "paymentCardAccountNumber": "",  
  "enrollmentDate": "",  
  "lastOpenDate": "",  
  "lastClosedDate": "",  
  "firstUseDate": "",  
  "accountStatusDate": "",  
  "productType": "",  
  "sourceCode": "",  
  "merchantCode": "",  
  "channelCode": "",  
  "firstName": "",  
  "lastName": "",  
  "birthDate": "",  
  "updateDateTime": "",  
  "active": false,  
  "cardComment": "",  
  "additionalProperties": {}  
}
```



14.9.4. Update Member Payment Card Account

PUT`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts/{memberPaymentCardAccountId}`**Try it** ▶

Update a member's payment card account

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: **Client Credentials**

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-payment-card:modify

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** `string` *

> **memberPaymentCardAccountId** `integer<int64>` *

Format: int64



Body Parameters

▼

body

MemberPaymentCardAccount

Description

Member Payment Card Account Object

>

typeCode

string

*

>

paymentCardAccountNumber

string

*

>

enrollmentDate

string<date>

📅

>

lastOpenDate

string<date>

📅

>

lastClosedDate

string<date>

📅

>

firstUseDate

string<date>

📅

>

accountStatusDate

string<date>

📅

>

productType

string

>

sourceCode

string

>

merchantCode

string

>

channelCode

string

>

firstName

string

>

lastName

string

>

birthDate

string<date>

📅

>

cardComment

string

additionalProperties

object

Responses

- 200

Success
- ▼

● 400

Bad request

▼

ErrorResponse

>

errors

ValidationError[]
- 401

Unauthorized
- 403

Forbidden
- 500

Internal error



Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts/{memberPaymentCardAccountIdentifier} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "typeCode": "",
    "paymentCardAccountNumber": ""
  }'
```

Responses

200

```
// Success
```



14.9.5. Delete Member Payment Card Account

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts/{memberPaymentCardAccount}`

Try it ▶

Delete a member's payment card account

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-payment-card:modify

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** *string* *

> **memberPaymentCardAccountId** *integer<int64>* *

Format: int64

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** *ValidationError[]*

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/payment-card-accounts/{memberPaymentCardAccount} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.10. member-phone

14.10.1. Get Member Phone Numbers

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones`

Try it ▶

Retrieve a member's phone numbers

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:member-phone:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Responses

▼ ● 200 Success

▼ `MemberPhone[]`

- > `memberPhoneId` `integer<int64>`
- > `phoneNumber` `string` *
- > `phoneExtension` `string`
- > `rawPhoneNumber` `string`
- > `carrier` `string`
- > `typeCode` `string`
- > `smsCapable` `boolean`
- > `validated` `boolean`
- > `statusCode` `string`
- > `order` `integer`
- > `updateDateTime` `string<date-time>`

▼ ● 400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "memberPhoneId": "",  
    "phoneNumber": "1 414 111-1111",  
    "phoneExtension": "",  
    "rawPhoneNumber": "",  
    "carrier": "",  
    "typeCode": "Mobile",  
    "smsCapable": false,  
    "validated": false,  
    "statusCode": "A",  
    "order": 1,  
    "updateDateTime": ""  
  }  
]
```



14.10.2. Create Member Phone Number

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones>

Try it ▶

Create a member's phone number

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-phone:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Body Parameters

▼ body MemberPhone

Description

Member Phone Object

> phoneNumber string *

> phoneExtension string

> carrier string

> typeCode string

> smsCapable boolean

> validated boolean

> statusCode string

> order integer

Responses

▼ ● 200 Success



▼ MemberPhone

```
> memberPhoneId integer<int64>
> phoneNumber string *
> phoneExtension string
> rawPhoneNumber string
> carrier string
> typeCode string
> smsCapable boolean
> validated boolean
> statusCode string
> order integer
> updateDateTime string<date-time>
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "phoneNumber": "1 414 111-1111"
  }'
```



Responses

● 200

```
// Success
{
  "memberPhoneId": "",
  "phoneNumber": "1 414 111-1111",
  "phoneExtension": "",
  "rawPhoneNumber": "",
  "carrier": "",
  "typeCode": "Mobile",
  "smsCapable": false,
  "validated": false,
  "statusCode": "A",
  "order": 1,
  "updateDateTime": ""
}
```



14.10.3. Get Member Phone

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones/{memberPhoneId}`

Try it

Get a member's phone number

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-phone:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberPhoneId integer<int64> *

Format: int64

Responses

✓ ● 200 Success

▼ MemberPhone

> memberPhoneId integer<int64>

> phoneNumber string *

> phoneExtension string

> rawPhoneNumber string

> carrier string

> typeCode string

> smsCapable boolean

> validated boolean

> statusCode string

> order integer

> updateDateTime string<date-time>

✓ ● 400 Bad request

▼ ErrorResponse

> errors ValidationErrors[]



- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

Curl

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones/{memberPhoneId} \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "memberPhoneId": "",  
  "phoneNumber": "1 414 111-1111",  
  "phoneExtension": "",  
  "rawPhoneNumber": "",  
  "carrier": "",  
  "typeCode": "Mobile",  
  "smsCapable": false,  
  "validated": false,  
  "statusCode": "A",  
  "order": 1,  
  "updateDateTime": ""  
}
```



14.10.4. Update Member Phone

PUT

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones/{memberPhoneId}`

Try it ▶

Update a member's phone number

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-phone:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberPhoneId integer<int64> *

Format: int64

Body Parameters

▼ body MemberPhone

Description

Member Phone Object

> phoneNumber string *

> phoneExtension string

> carrier string

> typeCode string

> smsCapable boolean

> validated boolean

> statusCode string

> order integer

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones/{memberPhoneId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "phoneNumber": "1 414 111-1111"
  }'
```

Responses

● 200

// Success



14.10.5. Delete Member Phone

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones/{memberPhoneId}`

Try it

Delete a member's phone number

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:member-phone:modify

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** *string* *

> **memberPhoneId** *integer<int64>* *

Format: int64

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** *ValidationError[]*

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/phones/{memberPhoneId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.11. member-merge

14.11.1. Merge Members

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/merges>

Try it ▶

Merge two member accounts

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member:merge

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Body Parameters

▾ body MemberMergeRequest *

> toMemberIdentifier string

Responses

▾ ● 200 Success

▾ MemberMerge

- > memberMergeId integer
- > fromMemberIdentifier string
- > toMemberIdentifier string
- > statusCode string

▾ ● 400 Bad request

▾ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/merges \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "memberMergeId": 0,  
  "fromMemberIdentifier": "",  
  "toMemberIdentifier": "",  
  "statusCode": ""  
}
```



14.11.2. Query Member Merge Status

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/merges/{memberMergeId}`

Try it ▶

Check progress status of a member merge

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member:merge

Authorize

Use Your Own Token

Path Parameters

> memberMergeId integer *

Responses

▼ ● 200 Success

▼ MemberMerge

- > memberMergeId integer
- > fromMemberIdentifier string
- > toMemberIdentifier string
- > statusCode string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/merges/{memberMergeId} \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "memberMergeId": 0,
  "fromMemberIdentifier": "",
  "toMemberIdentifier": "",
  "statusCode": ""
}
```



14.12. member-preference

14.12.1. Get Member Preferences

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences`

Try it ▶

Retrieve a member's preferences

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-preference:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Responses

▼ ● 200 Success

▼ MemberPreference[]

- > memberPreferenceId integer<int64>
- > preferenceRuleCode string *
- > preferenceRuleDescription string
- > preferenceRuleDetailCode string *
- > preferenceRuleDetailDescription string
- > preferenceTypeCode string
- > preferenceDataType string
- > value string
- > description string
- > updateDateTime string<date-time>
- > updateUserId integer<int64>
- > updateUserName string



▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "memberPreferenceId": "",
    "preferenceRuleCode": "",
    "preferenceRuleDescription": "",
    "preferenceRuleDetailCode": "",
    "preferenceRuleDetailDescription": "",
    "preferenceTypeCode": "",
    "preferenceDataType": "",
    "value": "",
    "description": "",
    "updateDateTime": "",
    "updateUserId": "",
    "updateUserName": ""
  }
]
```




14.12.2. Create Member Preference

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences>

Try it

Create a member's preference

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:member-preference:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Body Parameters

body MemberPreference

Description

Member Preference Object

> preferenceRuleCode string *

> preferenceRuleDetailCode string *

> value string

> description string

Responses

200 Success



▼ MemberPreference

```
> memberPreferenceId integer<int64>

> preferenceRuleCode string *

> preferenceRuleDescription string

> preferenceRuleDetailCode string *

> preferenceRuleDetailDescription string

> preferenceTypeCode string

> preferenceDataType string

> value string

> description string

> updateDateTime string<date-time>

> updateUserId integer<int64>

> updateUserName string
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "preferenceRuleCode": "",
    "preferenceRuleDetailCode": ""
  }'
```



Responses

● 200

```
// Success
{
  "memberPreferenceId": "",
  "preferenceRuleCode": "",
  "preferenceRuleDescription": "",
  "preferenceRuleDetailCode": "",
  "preferenceRuleDetailDescription": "",
  "preferenceTypeCode": "",
  "preferenceDataType": "",
  "value": "",
  "description": "",
  "updateDateTime": "",
  "updateUserId": "",
  "updateUserName": ""
}
```



14.12.3. Get Member Preference

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences/{memberPreferenceId}`

Try it ▶

Get a member's preference

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-preference:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberPreferenceId integer<int64> *

Format: int64

Responses

▼ ● 200 Success

▼ MemberPreference

> memberPreferenceId integer<int64>

> preferenceRuleCode string *

> preferenceRuleDescription string

> preferenceRuleDetailCode string *

> preferenceRuleDetailDescription string

> preferenceTypeCode string

> preferenceDataType string

> value string

> description string

> updateDateTime string<date-time>

> updateUserId integer<int64>

> updateUserName string

▼ ● 400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences/{memberPreferenceId} \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
{
  "memberPreferenceId": "",
  "preferenceRuleCode": "",
  "preferenceRuleDescription": "",
  "preferenceRuleDetailCode": "",
  "preferenceRuleDetailDescription": "",
  "preferenceTypeCode": "",
  "preferenceDataType": "",
  "value": "",
  "description": "",
  "updateDateTime": "",
  "updateUserId": "",
  "updateUserName": ""
}
```



14.12.4. Update Member Preference

PUT`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences/{memberPreferenceId}`

Try it

Update a member's preference

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:member-preference:modify

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** `string` *

> **memberPreferenceId** `integer<int64>` *

Format: int64

Body Parameters

body `MemberPreference`

Description

Member Preference Object

> **preferenceRuleCode** `string` *

> **preferenceRuleDetailCode** `string` *

> **value** `string`

> **description** `string`

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences/{memberPreferenceId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "preferenceRuleCode": "",
    "preferenceRuleDetailCode": ""
  }'
```

Responses

● 200

// Success



14.12.5. Delete Member Preference

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences/{memberPreferenceId}`

Try it

Delete a member's preference

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:member-preference:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberPreferenceId integer<int64> *

Format: int64

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/preferences/{memberPreferenceId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.13. member-promotion

14.13.1. Get Member Promotions

GET

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/promotions>

Try it ▶

Retrieve a member's registered promotions, including the associated promotion details, registration information, promotion counters, a member's progress toward completion, and any resulting promotion awards.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-promotion:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> cycle integer<int64>

> includeCouponDetails boolean

Format: int64

Responses

▼ ● 200 Success



▼ MemberPromotion[]

- > memberPromotionId integer<int64>
- > qualifyDateTime string<date-time>
- > lastUpdatedBy string
- > registeredBy string
- > promotion Promotion
- > registered boolean
- > registerDateTime string<date-time>
- > registerChannelCode string
- > createDateTime string<date-time>
- > endDateTime string<date-time>
- > timesAwarded integer<int64>
- > nextAwardThreshold number
- > promotionRegistrationSource string
- > preQualificationCounters MemberPromotionCounter[]
- > counters MemberPromotionCounter[]
- > details MemberPromotionDetail[]
- > results MemberPromotionResult[]
- > relatedPromotions MemberPromotion[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/promotions?includeCouponDetails=boolean' \
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "memberPromotionId": "",
    "qualifyDateTime": "",
    "lastUpdatedBy": "",
    "registeredBy": "",
    "promotion": {
      "promotionId": "",
      "promotionCode": "",
      "promotionName": "",
      "promotionDescription": "",
      "promotionTypeCode": "",
      "promotionSubTypeCode": "",
      "registerStartDateTime": "",
      "registerEndDateTime": "",
      "registrationChannelCodes": [
        ""
      ],
    },
    "autoRegister": false,
    "startDate": "",
    "endDate": "",
    "accrualTypeCode": "",
    "callCenterOnly": false,
    "displayOnline": false,
    "displayOnlineUntilDateTime": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": "",
      "ace15": "",
    },
  }
]
```



```
"ace15Description": "",
"ace01SmartGroupId": "",
"ace02SmartGroupId": "",
"ace03SmartGroupId": "",
"ace04SmartGroupId": "",
"ace05SmartGroupId": "",
"ace06SmartGroupId": "",
"ace07SmartGroupId": "",
"ace08SmartGroupId": "",
"ace09SmartGroupId": "",
"ace10SmartGroupId": "",
"ace11SmartGroupId": "",
"ace12SmartGroupId": "",
"ace13SmartGroupId": "",
"ace14SmartGroupId": "",
"ace15SmartGroupId": ""
},
"genericItem": {
  "contentCode": "Bonus100",
  "contentTypeCode": "HEADER",
  "name": "Bonus 100 points",
  "shortDescription": "Award 100 points",
  "description": "Award 100 points on 100th stay",
  "alternateDescription": "Award 100 points on 100th stay",
  "termsAndConditions": "Terms and conditions for this promotion",
  "title": "Award 100 points",
  "label": "100 points awarded",
  "logoKey": "100-points.png",
  "displayTypeKey": "100 points",
  "startDate": "",
  "endDate": "",
  "url": "/loyaltyprogram/bonus/100-points",
  "tierCode": "B",
  "displayForAll": true,
  "descriptorCode": "LIMITED_TIME"
},
"graceLimit": "",
"hasMinUserRoleToRegister": false,
"minimumAccrualAmount": 0,
"occurrenceLimit": "",
"partnerProgramCode": "",
"currencyCategoryCode": "",
"processRetroAccruals": false,
"restrict": false,
"memberListId": "",
"slidingQualificationPeriod": "",
"tierCode": "",
"trackRegistrationSource": false,
"preQualificationCounters": [
  {
    "counterTypeCode": "",
    "target": 0,
    "result": [
      {
        "type": "",
        "value": "",
        "valueType": "",
        "description": "",
        "resultAwardId": ""
      }
    ]
  }
]
```



```
        "resultAwardCode": "",
        "bonusByActivity": false,
        "displayName": ""
    }
]
},
"counters": [
{
    "counterTypeCode": "",
    "target": 0,
    "result": [
        {
            "type": "",
            "value": "",
            "valueType": "",
            "description": "",
            "resultAwardId": "",
            "resultAwardCode": "",
            "bonusByActivity": false,
            "displayName": ""
        }
    ]
}
],
"resultCommunication": false,
"resultCommunicationId": "",
"resultCurrency": false,
"resultCurrencyAmount": 0,
"resultCurrencyCategoryCode": "",
"resultCurrencyTierQualifier": false,
"resultExternal": false,
"resultExternalCode": "",
"resultPromotion": false,
"resultPromotionId": "",
"resultTier": false,
"resultTierCode": "",
"resultTierExpireDateTime": "",
"rewardThresholds": false,
"results": [
{
    "type": "",
    "value": "",
    "valueType": "",
    "description": "",
    "resultAwardId": "",
    "resultAwardCode": "",
    "bonusByActivity": false,
    "displayName": ""
}
],
"statusCode": "",
"renewableCounters": false
},
"registered": false,
"registerDateTime": "",
"registerChannelCode": "",
"createDateTime": "",
"endDateTime": "",
```



```
"timesAwarded": "",
"nextAwardThreshold": 0,
"promotionRegistrationSource": "",
"preQualificationCounters": [
  {
    "counterTypeCode": "",
    "target": 0,
    "progress": 0,
    "percentage": 0,
    "result": [
      {
        "type": "",
        "value": "",
        "valueType": "",
        "description": "",
        "resultAwardId": "",
        "resultAwardCode": "",
        "bonusByActivity": false,
        "displayName": "",
        "memberCoupons": [
          {
            "memberCouponId": 147258369,
            "memberIdentifier": "",
            "coupon": {
              "couponId": 147258369,
              "couponRequestId": 147258370,
              "offer": {
                "offerId": 147258369,
                "offerTypeCode": "Discount",
                "code": "ALON2020",
                "name": "Free night",
                "description": "Award double points",
                "featureVersion": 2,
                "statusCode": "Enabled",
                "startDateTime": "",
                "endDateTime": "",
                "offerAttributes": [
                  {
                    "": ""
                  }
                ]
              },
              "code": "ZPVWTAROW20Q",
              "statusCode": "Activated",
              "startDate": "",
              "endDate": "",
              "useTypeCode": "Single",
              "useLimit": 1,
              "currentUse": 1,
              "couponUses": [
                {
                  "useDateTime": "",
                  "statusCode": "Used",
                  "channelCode": "System",
                  "alias": "",
                  "useCount": 1,
                  "couponUseAttributes": [
                    {
                      "": ""
                    }
                  ]
                }
              ]
            }
          }
        ]
      }
    ]
  }
]
```



```
    }
  ]
}
],
"couponAttributes": [
  {
    "couponAttributeId": 147258369,
    "attribute": {
      "": ""
    },
    "value": "2021-12-31T12:00:00",
    "requireOnUse": true,
    "requireOnHold": false
  }
],
"sourceTypeCode": "Promotion",
"sourceReferenceRuleCode": "StayBonus",
"sourceReferenceId": 147258369,
"aggregateStatusCode": "Active",
"statusCode": "Active",
"transferable": false,
"previouslyTransferred": false
}
]
}
]
}
],
"counters": [
  {
    "counterTypeCode": "",
    "target": 0,
    "progress": 0,
    "percentage": 0,
    "result": [
      {
        "type": "",
        "value": "",
        "valueType": "",
        "description": "",
        "resultAwardId": "",
        "resultAwardCode": "",
        "bonusByActivity": false,
        "displayName": "",
        "memberCoupons": [
          {
            "memberCouponId": 147258369,
            "memberIdentifier": "",
            "coupon": {
              "couponId": 147258369,
              "couponRequestId": 147258370,
              "offer": {
                "offerId": 147258369,
                "offerTypeCode": "Discount",
                "code": "ALON2020",
                "name": "Free night",
                "description": "Award double points",
                "featureVersion": 2,
```




```
        "statusCode": "Enabled",
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
            {
                "": ""
            }
        ],
        "code": "ZPVWTAROW20Q",
        "statusCode": "Activated",
        "startDate": "",
        "endDate": "",
        "useTypeCode": "Single",
        "useLimit": 1,
        "currentUse": 1,
        "couponUses": [
            {
                "useDateTime": "",
                "statusCode": "Used",
                "channelCode": "System",
                "alias": "",
                "useCount": 1,
                "couponUseAttributes": [
                    {
                        "": ""
                    }
                ]
            }
        ],
        "couponAttributes": [
            {
                "couponAttributeId": 147258369,
                "attribute": {
                    "": ""
                },
                "value": "2021-12-31T12:00:00",
                "requireOnUse": true,
                "requireOnHold": false
            }
        ],
        "sourceTypeCode": "Promotion",
        "sourceReferenceRuleCode": "StayBonus",
        "sourceReferenceId": 147258369,
        "aggregateStatusCode": "Active",
        "statusCode": "Active",
        "transferable": false,
        "previouslyTransferred": false
    }
}
]
}
]
}
],
"details": [
    {
        "referenceId": "",
        "transactionId": "",
    }
]
```



```
"accrualDateTime": "",
"description": "",
"customData": {
  "ace01": "",
  "ace01Description": "",
  "ace02": "",
  "ace02Description": "",
  "ace03": "",
  "ace03Description": "",
  "ace04": "",
  "ace04Description": "",
  "ace05": "",
  "ace05Description": "",
  "ace06": "",
  "ace06Description": "",
  "ace07": "",
  "ace07Description": "",
  "ace08": "",
  "ace08Description": "",
  "ace09": "",
  "ace09Description": "",
  "ace10": "",
  "ace10Description": "",
  "ace11": "",
  "ace11Description": "",
  "ace12": "",
  "ace12Description": "",
  "ace13": "",
  "ace13Description": "",
  "ace14": "",
  "ace14Description": "",
  "ace15": "",
  "ace15Description": ""
},
"cycle": "",
"contributedToPreQualification": false,
"slidingQualificationAwarded": false,
"checkInDate": "",
"checkOutDate": "",
"reservationNumber": "",
"counters": [
  {
    "counterTypeCode": "",
    "counterValue": 0
  }
]
},
"results": [
  {
    "type": "",
    "value": "",
    "valueType": "",
    "description": "",
    "resultAwardId": "",
    "resultAwardCode": "",
    "bonusByActivity": false,
    "displayName": "",
    "memberCoupons": [
```



```
{
  "memberCouponId": 147258369,
  "memberIdentifier": "",
  "coupon": {
    "couponId": 147258369,
    "couponRequestId": 147258370,
    "offer": {
      "offerId": 147258369,
      "offerTypeCode": "Discount",
      "code": "ALON2020",
      "name": "Free night",
      "description": "Award double points",
      "featureVersion": 2,
      "statusCode": "Enabled",
      "startDateTime": "",
      "endDateTime": "",
      "offerAttributes": [
        {
          "offerAttributeId": 147258369,
          "attribute": {
            "": ""
          },
          "defaultValue": "Standard",
          "requireOnUse": true,
          "requireOnHold": true,
          "includeOnResponse": false,
          "attributeTypeCode": "Validation",
          "operatorCode": "EQ",
          "operand": "GI",
          "smartGroupId": 147258369
        }
      ]
    }
  }
}
```



14.13.2. Register member for promotion

POST

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/promotions

Try it ▶

Register member for promotion

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-promotion:register

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Body Parameters

body RegisterForPromotionRequest

Description

Register For Promotion Object

> promotionId integer<int64>

Format: int64

> promotionCode string

> registrationSource string

Responses

- 200 Success

RegisterForPromotionResponse

> memberPromotionIds integer[] *
- 400 Bad request

ErrorResponse

> errors ValidationError[]

401 Unauthorized

403 Forbidden

500 Internal error
- Request

Curl

```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/promotions \
  --header 'accept: application/json'
```
- 464 / 1062
- Created at Oct 02, 2026



Responses

● 200

```
// Success
{
  "memberPromotionIds": [
    0
  ]
}
```



14.13.3. Get Member Promotion By Member Promotion Id

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/promotions/{memberPromotionId}

Try it ▶

Retrieve a member's registered promotion by member promotion id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-promotion:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberPromotionId integer<int64> *

Format: int64

Query Parameters

> cycle integer<int64>

> includeCouponDetails boolean

Format: int64

Responses

▼ ● 200 Success



▼ MemberPromotion

- > memberPromotionId integer<int64>
- > qualifyDateTime string<date-time>
- > lastUpdatedBy string
- > registeredBy string
- > promotion Promotion
- > registered boolean
- > registerDateTime string<date-time>
- > registerChannelCode string
- > createDateTime string<date-time>
- > endDateTime string<date-time>
- > timesAwarded integer<int64>
- > nextAwardThreshold number
- > promotionRegistrationSource string
- > preQualificationCounters MemberPromotionCounter[]
- > counters MemberPromotionCounter[]
- > details MemberPromotionDetail[]
- > results MemberPromotionResult[]
- > relatedPromotions MemberPromotion[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized
● 403 Forbidden
● 500 Internal error

Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/promotions/{memberPromotionId}?includeCouponD  
  --header 'accept: application/json'
```



Responses

200



```
// Success
{
  "memberPromotionId": "",
  "qualifyDateTime": "",
  "lastUpdatedBy": "",
  "registeredBy": "",
  "promotion": {
    "promotionId": "",
    "promotionCode": "",
    "promotionName": "",
    "promotionDescription": "",
    "promotionTypeCode": "",
    "promotionSubTypeCode": "",
    "registerStartDateTime": "",
    "registerEndDateTime": "",
    "registrationChannelCodes": [
      ""
    ],
    "autoRegister": false,
    "startDate": "",
    "endDate": "",
    "accrualTypeCode": "",
    "callCenterOnly": false,
    "displayOnline": false,
    "displayOnlineUntilDateTime": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": "",
      "ace15": "",
      "ace15Description": ""
    }
  }
}
```



```
"ace01SmartGroupId": "",
"ace02SmartGroupId": "",
"ace03SmartGroupId": "",
"ace04SmartGroupId": "",
"ace05SmartGroupId": "",
"ace06SmartGroupId": "",
"ace07SmartGroupId": "",
"ace08SmartGroupId": "",
"ace09SmartGroupId": "",
"ace10SmartGroupId": "",
"ace11SmartGroupId": "",
"ace12SmartGroupId": "",
"ace13SmartGroupId": "",
"ace14SmartGroupId": "",
"ace15SmartGroupId": ""
},
"genericItem": {
  "contentCode": "Bonus100",
  "contentTypeCode": "HEADER",
  "name": "Bonus 100 points",
  "shortDescription": "Award 100 points",
  "description": "Award 100 points on 100th stay",
  "alternateDescription": "Award 100 points on 100th stay",
  "termsAndConditions": "Terms and conditions for this promotion",
  "title": "Award 100 points",
  "label": "100 points awarded",
  "logoKey": "100-points.png",
  "displayTypeKey": "100 points",
  "startDate": "",
  "endDate": "",
  "url": "/loyaltyprogram/bonus/100-points",
  "tierCode": "B",
  "displayForAll": true,
  "descriptorCode": "LIMITED_TIME"
},
"graceLimit": "",
"hasMinUserRoleToRegister": false,
"minimumAccrualAmount": 0,
"occurrenceLimit": "",
"partnerProgramCode": "",
"currencyCategoryCode": "",
"processRetroAccruals": false,
"restrict": false,
"memberListId": "",
"slidingQualificationPeriod": "",
"tierCode": "",
"trackRegistrationSource": false,
"preQualificationCounters": [
  {
    "counterTypeCode": "",
    "target": 0,
    "result": [
      {
        "type": "",
        "value": "",
        "valueType": "",
        "description": "",
        "resultAwardId": "",
        "resultAwardCode": ""
      }
    ]
  }
]
```



```
        "bonusByActivity": false,
        "displayName": ""
    }
}
],
"counters": [
{
    "counterTypeCode": "",
    "target": 0,
    "result": [
        {
            "type": "",
            "value": "",
            "valueType": "",
            "description": "",
            "resultAwardId": "",
            "resultAwardCode": "",
            "bonusByActivity": false,
            "displayName": ""
        }
    ]
}
],
"resultCommunication": false,
"resultCommunicationId": "",
"resultCurrency": false,
"resultCurrencyAmount": 0,
"resultCurrencyCategoryCode": "",
"resultCurrencyTierQualifier": false,
"resultExternal": false,
"resultExternalCode": "",
"resultPromotion": false,
"resultPromotionId": "",
"resultTier": false,
"resultTierCode": "",
"resultTierExpireDateTime": "",
"rewardThresholds": false,
"results": [
{
    "type": "",
    "value": "",
    "valueType": "",
    "description": "",
    "resultAwardId": "",
    "resultAwardCode": "",
    "bonusByActivity": false,
    "displayName": ""
}
],
"statusCode": "",
"renewableCounters": false
},
"registered": false,
"registerDateTime": "",
"registerChannelCode": "",
"createDateTime": "",
"endDateTime": "",
"timesAwarded": "",
```



```
"nextAwardThreshold": 0,
"promotionRegistrationSource": "",
"preQualificationCounters": [
  {
    "counterTypeCode": "",
    "target": 0,
    "progress": 0,
    "percentage": 0,
    "result": [
      {
        "type": "",
        "value": "",
        "valueType": "",
        "description": "",
        "resultAwardId": "",
        "resultAwardCode": "",
        "bonusByActivity": false,
        "displayName": "",
        "memberCoupons": [
          {
            "memberCouponId": 147258369,
            "memberIdentifier": "",
            "coupon": {
              "couponId": 147258369,
              "couponRequestId": 147258370,
              "offer": {
                "offerId": 147258369,
                "offerTypeCode": "Discount",
                "code": "ALON2020",
                "name": "Free night",
                "description": "Award double points",
                "featureVersion": 2,
                "statusCode": "Enabled",
                "startDateTime": "",
                "endDateTime": "",
                "offerAttributes": [
                  {
                    "": ""
                  }
                ]
              },
              "code": "ZPVWTAROW20Q",
              "statusCode": "Activated",
              "startDate": "",
              "endDate": "",
              "useTypeCode": "Single",
              "useLimit": 1,
              "currentUse": 1,
              "couponUses": [
                {
                  "useDateTime": "",
                  "statusCode": "Used",
                  "channelCode": "System",
                  "alias": "",
                  "useCount": 1,
                  "couponUseAttributes": [
                    {
                      "": ""
                    }
                  ]
                }
              ]
            }
          }
        ]
      }
    ]
  }
]
```



```
    ]
  }
],
"couponAttributes": [
  {
    "couponAttributeId": 147258369,
    "attribute": {
      "": ""
    },
    "value": "2021-12-31T12:00:00",
    "requireOnUse": true,
    "requireOnHold": false
  }
]
},
"sourceTypeCode": "Promotion",
"sourceReferenceRuleCode": "StayBonus",
"sourceReferenceId": 147258369,
"aggregateStatusCode": "Active",
"statusCode": "Active",
"transferable": false,
"previouslyTransferred": false
}
]
}
]
},
"counters": [
  {
    "counterTypeCode": "",
    "target": 0,
    "progress": 0,
    "percentage": 0,
    "result": [
      {
        "type": "",
        "value": "",
        "valueType": "",
        "description": "",
        "resultAwardId": "",
        "resultAwardCode": "",
        "bonusByActivity": false,
        "displayName": "",
        "memberCoupons": [
          {
            "memberCouponId": 147258369,
            "memberIdentifier": "",
            "coupon": {
              "couponId": 147258369,
              "couponRequestId": 147258370,
              "offer": {
                "offerId": 147258369,
                "offerTypeCode": "Discount",
                "code": "ALON2020",
                "name": "Free night",
                "description": "Award double points",
                "featureVersion": 2,
                "statusCode": "Enabled",
```



```
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
            {
                "": ""
            }
        ]
    },
    "code": "ZPVWTAROW20Q",
    "statusCode": "Activated",
    "startDate": "",
    "endDate": "",
    "useTypeCode": "Single",
    "useLimit": 1,
    "currentUse": 1,
    "couponUses": [
        {
            "useDateTime": "",
            "statusCode": "Used",
            "channelCode": "System",
            "alias": "",
            "useCount": 1,
            "couponUseAttributes": [
                {
                    "": ""
                }
            ]
        }
    ],
    "couponAttributes": [
        {
            "couponAttributeId": 147258369,
            "attribute": {
                "": ""
            },
            "value": "2021-12-31T12:00:00",
            "requireOnUse": true,
            "requireOnHold": false
        }
    ]
},
"sourceTypeCode": "Promotion",
"sourceReferenceRuleCode": "StayBonus",
"sourceReferenceId": 147258369,
"aggregateStatusCode": "Active",
"statusCode": "Active",
"transferable": false,
"previouslyTransferred": false
}
]
}
]
},
"details": [
    {
        "referenceId": "",
        "transactionId": "",
        "accrualDateTime": "",
```



```
"description": "",
"customData": {
  "ace01": "",
  "ace01Description": "",
  "ace02": "",
  "ace02Description": "",
  "ace03": "",
  "ace03Description": "",
  "ace04": "",
  "ace04Description": "",
  "ace05": "",
  "ace05Description": "",
  "ace06": "",
  "ace06Description": "",
  "ace07": "",
  "ace07Description": "",
  "ace08": "",
  "ace08Description": "",
  "ace09": "",
  "ace09Description": "",
  "ace10": "",
  "ace10Description": "",
  "ace11": "",
  "ace11Description": "",
  "ace12": "",
  "ace12Description": "",
  "ace13": "",
  "ace13Description": "",
  "ace14": "",
  "ace14Description": "",
  "ace15": "",
  "ace15Description": ""
},
"cycle": "",
"contributedToPreQualification": false,
"slidingQualificationAwarded": false,
"checkInDate": "",
"checkOutDate": "",
"reservationNumber": "",
"counters": [
  {
    "counterTypeCode": "",
    "counterValue": 0
  }
]
},
"results": [
  {
    "type": "",
    "value": "",
    "valueType": "",
    "description": "",
    "resultAwardId": "",
    "resultAwardCode": "",
    "bonusByActivity": false,
    "displayName": "",
    "memberCoupons": [
      {
```



```
"memberCouponId": 147258369,
"memberIdentifier": "",
"coupon": {
  "couponId": 147258369,
  "couponRequestId": 147258370,
  "offer": {
    "offerId": 147258369,
    "offerTypeCode": "Discount",
    "code": "ALON2020",
    "name": "Free night",
    "description": "Award double points",
    "featureVersion": 2,
    "statusCode": "Enabled",
    "startDateTime": "",
    "endDateTime": "",
    "offerAttributes": [
      {
        "offerAttributeId": 147258369,
        "attribute": {
          "": ""
        },
        "defaultValue": "Standard",
        "requireOnUse": true,
        "requireOnHold": true,
        "includeOnResponse": false,
        "attributeTypeCode": "Validation",
        "operatorCode": "EQ",
        "operand": "GI",
```




14.14. member-redemption

14.14.1. Get Member Redemptions

GET

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions>

Try it ▶

Retrieve a member's redemptions

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-redemption:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Responses

▼ ● 200 Success



- ▼ MemberRedemption[]
 - > memberRedemptionId object
 - > memberIdentifier string
 - > redemptionItemCode string
 - > redemptionItemDescription string
 - > referenceId string
 - > redemptionDate string<date>
 - > customData CustomData
 - > quantity integer<int32>
 - > currencyValue number
 - > partnerCode string
 - > expectedShipDate string<date>
 - > orderAddress MemberRedemptionOrderAddress
 - > bonuses Transaction[]
 - > coupons MemberCoupon[]
 - > relatedRedemptionId integer<int64>
 - > fraudRiskScore number
 - > fraudConfirmation boolean
 - > status string
 - > redemptionRuleCode string
 - > transactionAmount number
 - > invoiceNumber string
 - > lineItems MemberRedemptionLineItem[]
 - > poolFunding RedemptionPoolContribution[]
 - > poolCancellations RedemptionPoolContribution[]

▼ ● 400 Bad request

- ▼ ErrorResponse
 - > errors ValidationError[]

● 401 Unauthorized



- **403** Forbidden
- **500** Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "memberRedemptionId": {},
    "memberIdentifier": "",
    "redemptionItemCode": "GIFT-CARD-50",
    "redemptionItemDescription": "50$ Gift Card",
    "referenceId": 12345,
    "redemptionDate": "2020-01-01",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": "",
      "ace15": "",
      "ace15Description": ""
    },
    "quantity": 10,
    "currencyValue": 100,
    "partnerCode": "PARTNER123",
    "expectedShipDate": "2020-01-01",
    "orderAddress": {
      "name": "",
      "address1": "",
      "address2": "",
      "address3": "",
      "city": "",
      "stateCode": "",
      "zip": "",
      "countryCode": "",
      "emailAddress": "",
      "phoneNumber": ""
    }
  }
]
```



```
},
"bonuses": [
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": "",
      "ace15": "",
      "ace15Description": ""
    }
  },
],
```



```
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
  "totalRevenueAmount": 0,
  "totalRevenueAmountIntl": 0,
  "loyaltyEligibleAmount": 0,
```



```
"loyaltyEligibleAmountInt1": 0,
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountInt1": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueInt1": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueInt1": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
"transactionAdjustment": {
  "comments": "",
  "referenceId": "",
  "reasonCode": ""
},
"transactionBonus": {
  "bonusCode": "",
  "bonusType": ""
},
"transactionPartner": {
  "partnerName": "",
  "partnerCurrencyCategoryCode": ""
},
"transactionPromotion": {
  "promotionCode": ""
},
"transactionTransfer": {
  "fromMemberAccountNumber": "A234567890",
  "fromMemberName": "Marcel Rufus",
```




```
    "toMemberAccountNumber": "B987654321",
    "toMemberName": "Janel Liesje",
    "transferredBy": "A0123456789"
  },
  "mergeDetails": {
    "fromMemberAccountNumber": "A234567890"
  },
  "transactionRelationships": [
    {
      "type": "B2B_SUCCESSOR",
      "key": "EB-1234",
      "accountNumber": ""
    }
  ]
},
"coupons": [
  {
    "memberCouponId": 147258369,
    "memberIdentifier": "",
    "coupon": {
      "couponId": 147258369,
      "couponRequestId": 147258370,
      "offer": {
        "offerId": 147258369,
        "offerTypeCode": "Discount",
        "code": "ALON2020",
        "name": "Free night",
        "description": "Award double points",
        "featureVersion": 2,
        "statusCode": "Enabled",
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
          {
            "offerAttributeId": 147258369,
            "attribute": {
              "attributeId": 147258369,
              "typeCode": "Validation",
              "statusCode": "Enabled",
              "name": "AccountNumber",
              "description": "Account Number",
              "dataTypeCode": "open"
            }
          },
          {
            "defaultValue": "Standard",
            "requireOnUse": true,
            "requireOnHold": true,
            "includeOnResponse": false,
            "attributeTypeCode": "Validation",
            "operatorCode": "EQ",
            "operand": "GI",
            "smartGroupId": 147258369
          }
        ]
      }
    }
  },
  {
    "code": "ZPVWTAROW20Q",
    "statusCode": "Activated",
    "startDate": "",
    "endDate": ""
  }
]
```



```
"useTypeCode": "Single",
"useLimit": 1,
"currentUse": 1,
"couponUses": [
  {
    "useDateTime": "",
    "statusCode": "Used",
    "channelCode": "System",
    "alias": "",
    "useCount": 1,
    "couponUseAttributes": [
      {
        "couponUseAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
          "description": "Account Number",
```



14.14.2. Create Member Redemption

POST

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions

Try it ▶

Redeem redemption item

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-redemption:create

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> createCouponUse boolean

> holdCoupon boolean

> validateOnly boolean

> allowOverRedeem boolean

> allowPoolRedeem boolean

> migrate boolean

> skipFraudCheck boolean

> async boolean



Body Parameters

▼

body

MemberRedemptionRequest

Description

Member Redemption Request Object

>

redemptionItemCode

string

>

redemptionDate

string<date>

mm/dd/yyyy📅

>

referenceId

string

>

quantity

integer<int32> *

Format: int32

>

providedAmount

number

>

transactionAmount

number

>

externalCode

string

>

offerCriteria

map<string, string>

>

orderAddress

MemberRedemptionOrderAddress

>

couponUse

CouponUseRequest

>

lineItems

MemberRedemptionLineItem[]

>

status

string

>

expectedShipDate

string<date>

mm/dd/yyyy📅

>

invoiceNumber

string

Responses

▼

● 200

Success



▼ MemberRedemption

- > memberRedemptionId object
- > memberIdentifier string
- > redemptionItemCode string
- > redemptionItemDescription string
- > referenceId string
- > redemptionDate string<date>
- > customData CustomData
- > quantity integer<int32>
- > currencyValue number
- > partnerCode string
- > expectedShipDate string<date>
- > orderAddress MemberRedemptionOrderAddress
- > bonuses Transaction[]
- > coupons MemberCoupon[]
- > relatedRedemptionId integer<int64>
- > fraudRiskScore number
- > fraudConfirmation boolean
- > status string
- > redemptionRuleCode string
- > transactionAmount number
- > invoiceNumber string
- > lineItems MemberRedemptionLineItem[]
- > poolFunding RedemptionPoolContribution[]
- > poolCancellations RedemptionPoolContribution[]

▼ ● 202 Accepted



▼ AsyncTask

```
> asyncTaskId integer<int64>
> taskTypeCode string
> taskCode string
> requestDate string<date-time>
> requestData object
> statusCode string
> responseDate string<date-time>
> responseData object
> fileName string
> lineNumber integer
> fileData string
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions?createCouponUse=boolean&holdCoupo
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "quantity": "",
    "orderAddress": {
      "address1": "",
      "city": "",
      "stateCode": "",
      "zip": "",
      "countryCode": ""
    }
  }'
```



Responses

● 200



```
// Success
{
  "memberRedemptionId": {},
  "memberIdentifier": "",
  "redemptionItemCode": "GIFT-CARD-50",
  "redemptionItemDescription": "50$ Gift Card",
  "referenceId": 12345,
  "redemptionDate": "2020-01-01",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "quantity": 10,
  "currencyValue": 100,
  "partnerCode": "PARTNER123",
  "expectedShipDate": "2020-01-01",
  "orderAddress": {
    "name": "",
    "address1": "",
    "address2": "",
    "address3": "",
    "city": "",
    "stateCode": "",
    "zip": "",
    "countryCode": "",
    "emailAddress": "",
    "phoneNumber": ""
  }
},
```




```
"bonuses": [  
  {  
    "transactionId": "",  
    "transactionGroupId": "",  
    "transactionGroupReferenceId": "",  
    "parentTransactionId": "",  
    "linkedTransactionId": "",  
    "referenceId": "",  
    "memberIdentifier": "",  
    "accountNumber": "",  
    "memberRedemptionId": "1",  
    "transactionDateTime": "",  
    "partnerName": "",  
    "partnerProgramCode": "",  
    "description": "",  
    "typeCode": "",  
    "typeDescription": "",  
    "subTypeCode": "",  
    "postDate": "",  
    "originationDate": "",  
    "retro": false,  
    "edited": false,  
    "sourceCode": "",  
    "statusCode": "",  
    "updateDateTime": "",  
    "earningRuleDescription": "",  
    "customData": {  
      "ace01": "",  
      "ace01Description": "",  
      "ace02": "",  
      "ace02Description": "",  
      "ace03": "",  
      "ace03Description": "",  
      "ace04": "",  
      "ace04Description": "",  
      "ace05": "",  
      "ace05Description": "",  
      "ace06": "",  
      "ace06Description": "",  
      "ace07": "",  
      "ace07Description": "",  
      "ace08": "",  
      "ace08Description": "",  
      "ace09": "",  
      "ace09Description": "",  
      "ace10": "",  
      "ace10Description": "",  
      "ace11": "",  
      "ace11Description": "",  
      "ace12": "",  
      "ace12Description": "",  
      "ace13": "",  
      "ace13Description": "",  
      "ace14": "",  
      "ace14Description": "",  
      "ace15": "",  
      "ace15Description": ""  
    },  
    },  
    "spend": {  
      }
```



```
"perItem": 0,
"spendAmount": 0,
"currencyTypeCode": "",
"currencyCategoryCode": "",
"isoCurrencyCode": "",
"currencyConversionRatio": "",
"quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
  "totalRevenueAmount": 0,
  "totalRevenueAmountIntl": 0,
  "loyaltyEligibleAmount": 0,
  "loyaltyEligibleAmountIntl": 0,
```



```
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountInt1": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueInt1": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueInt1": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
"transactionAdjustment": {
  "comments": "",
  "referenceId": "",
  "reasonCode": ""
},
"transactionBonus": {
  "bonusCode": "",
  "bonusType": ""
},
"transactionPartner": {
  "partnerName": "",
  "partnerCurrencyCategoryCode": ""
},
"transactionPromotion": {
  "promotionCode": ""
},
"transactionTransfer": {
  "fromMemberAccountNumber": "A234567890",
  "fromMemberName": "Marcel Rufus",
  "toMemberAccountNumber": "B987654321",
```



```
    "toMemberName": "Janel Liesje",
    "transferredBy": "A0123456789"
  },
  "mergeDetails": {
    "fromMemberAccountNumber": "A234567890"
  },
  "transactionRelationships": [
    {
      "type": "B2B_SUCCESSOR",
      "key": "EB-1234",
      "accountNumber": ""
    }
  ]
}
],
"coupons": [
  {
    "memberCouponId": 147258369,
    "memberIdentifier": "",
    "coupon": {
      "couponId": 147258369,
      "couponRequestId": 147258370,
      "offer": {
        "offerId": 147258369,
        "offerTypeCode": "Discount",
        "code": "ALON2020",
        "name": "Free night",
        "description": "Award double points",
        "featureVersion": 2,
        "statusCode": "Enabled",
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
          {
            "offerAttributeId": 147258369,
            "attribute": {
              "attributeId": 147258369,
              "typeCode": "Validation",
              "statusCode": "Enabled",
              "name": "AccountNumber",
              "description": "Account Number",
              "dataTypeCode": "open"
            }
          },
          {
            "defaultValue": "Standard",
            "requireOnUse": true,
            "requireOnHold": true,
            "includeOnResponse": false,
            "attributeTypeCode": "Validation",
            "operatorCode": "EQ",
            "operand": "GI",
            "smartGroupId": 147258369
          }
        ]
      },
      "code": "ZPVWTAROW20Q",
      "statusCode": "Activated",
      "startDate": "",
      "endDate": "",
      "useTypeCode": "Single",
```



```
"useLimit": 1,
"currentUse": 1,
"couponUses": [
  {
    "useDateTime": "",
    "statusCode": "Used",
    "channelCode": "System",
    "alias": "",
    "useCount": 1,
    "couponUseAttributes": [
      {
        "couponUseAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
```





14.14.3. Get Member Redemption By Member Redemption Id

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/{memberRedemptionId}

Try it ▶

Retrieve a member's redemption by member redemption id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-redemption:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberRedemptionId integer<int64> *

Format: int64

Responses

▼ ● 200 Success



▼ MemberRedemption

- > memberRedemptionId object
- > memberIdentifier string
- > redemptionItemCode string
- > redemptionItemDescription string
- > referenceId string
- > redemptionDate string<date>
- > customData CustomData
- > quantity integer<int32>
- > currencyValue number
- > partnerCode string
- > expectedShipDate string<date>
- > orderAddress MemberRedemptionOrderAddress
- > bonuses Transaction[]
- > coupons MemberCoupon[]
- > relatedRedemptionId integer<int64>
- > fraudRiskScore number
- > fraudConfirmation boolean
- > status string
- > redemptionRuleCode string
- > transactionAmount number
- > invoiceNumber string
- > lineItems MemberRedemptionLineItem[]
- > poolFunding RedemptionPoolContribution[]
- > poolCancellations RedemptionPoolContribution[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized



- **403** Forbidden
- **500** Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/{memberRedemptionId} \
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "memberRedemptionId": {},
  "memberIdentifier": "",
  "redemptionItemCode": "GIFT-CARD-50",
  "redemptionItemDescription": "50$ Gift Card",
  "referenceId": 12345,
  "redemptionDate": "2020-01-01",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "quantity": 10,
  "currencyValue": 100,
  "partnerCode": "PARTNER123",
  "expectedShipDate": "2020-01-01",
  "orderAddress": {
    "name": "",
    "address1": "",
    "address2": "",
    "address3": "",
    "city": "",
    "stateCode": "",
    "zip": "",
    "countryCode": "",
    "emailAddress": "",
    "phoneNumber": ""
  }
},
```



```
"bonuses": [  
  {  
    "transactionId": "",  
    "transactionGroupId": "",  
    "transactionGroupReferenceId": "",  
    "parentTransactionId": "",  
    "linkedTransactionId": "",  
    "referenceId": "",  
    "memberIdentifier": "",  
    "accountNumber": "",  
    "memberRedemptionId": "1",  
    "transactionDateTime": "",  
    "partnerName": "",  
    "partnerProgramCode": "",  
    "description": "",  
    "typeCode": "",  
    "typeDescription": "",  
    "subTypeCode": "",  
    "postDate": "",  
    "originationDate": "",  
    "retro": false,  
    "edited": false,  
    "sourceCode": "",  
    "statusCode": "",  
    "updateDateTime": "",  
    "earningRuleDescription": "",  
    "customData": {  
      "ace01": "",  
      "ace01Description": "",  
      "ace02": "",  
      "ace02Description": "",  
      "ace03": "",  
      "ace03Description": "",  
      "ace04": "",  
      "ace04Description": "",  
      "ace05": "",  
      "ace05Description": "",  
      "ace06": "",  
      "ace06Description": "",  
      "ace07": "",  
      "ace07Description": "",  
      "ace08": "",  
      "ace08Description": "",  
      "ace09": "",  
      "ace09Description": "",  
      "ace10": "",  
      "ace10Description": "",  
      "ace11": "",  
      "ace11Description": "",  
      "ace12": "",  
      "ace12Description": "",  
      "ace13": "",  
      "ace13Description": "",  
      "ace14": "",  
      "ace14Description": "",  
      "ace15": "",  
      "ace15Description": ""  
    },  
    },  
    "spend": {  
      "ace01": "",  
      "ace01Description": "",  
      "ace02": "",  
      "ace02Description": "",  
      "ace03": "",  
      "ace03Description": "",  
      "ace04": "",  
      "ace04Description": "",  
      "ace05": "",  
      "ace05Description": "",  
      "ace06": "",  
      "ace06Description": "",  
      "ace07": "",  
      "ace07Description": "",  
      "ace08": "",  
      "ace08Description": "",  
      "ace09": "",  
      "ace09Description": "",  
      "ace10": "",  
      "ace10Description": "",  
      "ace11": "",  
      "ace11Description": "",  
      "ace12": "",  
      "ace12Description": "",  
      "ace13": "",  
      "ace13Description": "",  
      "ace14": "",  
      "ace14Description": "",  
      "ace15": "",  
      "ace15Description": ""  
    }  
  }  
]
```



```
"perItem": 0,
"spendAmount": 0,
"currencyTypeCode": "",
"currencyCategoryCode": "",
"isoCurrencyCode": "",
"currencyConversionRatio": "",
"quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
  "totalRevenueAmount": 0,
  "totalRevenueAmountIntl": 0,
  "loyaltyEligibleAmount": 0,
  "loyaltyEligibleAmountIntl": 0,
```



```
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountInt1": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueInt1": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueInt1": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
"transactionAdjustment": {
  "comments": "",
  "referenceId": "",
  "reasonCode": ""
},
"transactionBonus": {
  "bonusCode": "",
  "bonusType": ""
},
"transactionPartner": {
  "partnerName": "",
  "partnerCurrencyCategoryCode": ""
},
"transactionPromotion": {
  "promotionCode": ""
},
"transactionTransfer": {
  "fromMemberAccountNumber": "A234567890",
  "fromMemberName": "Marcel Rufus",
  "toMemberAccountNumber": "B987654321",
```



```
    "toMemberName": "Janel Liesje",
    "transferredBy": "A0123456789"
  },
  "mergeDetails": {
    "fromMemberAccountNumber": "A234567890"
  },
  "transactionRelationships": [
    {
      "type": "B2B_SUCCESSOR",
      "key": "EB-1234",
      "accountNumber": ""
    }
  ]
},
],
"coupons": [
  {
    "memberCouponId": 147258369,
    "memberIdentifier": "",
    "coupon": {
      "couponId": 147258369,
      "couponRequestId": 147258370,
      "offer": {
        "offerId": 147258369,
        "offerTypeCode": "Discount",
        "code": "ALON2020",
        "name": "Free night",
        "description": "Award double points",
        "featureVersion": 2,
        "statusCode": "Enabled",
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
          {
            "offerAttributeId": 147258369,
            "attribute": {
              "attributeId": 147258369,
              "typeCode": "Validation",
              "statusCode": "Enabled",
              "name": "AccountNumber",
              "description": "Account Number",
              "dataTypeCode": "open"
            }
          },
          {
            "defaultValue": "Standard",
            "requireOnUse": true,
            "requireOnHold": true,
            "includeOnResponse": false,
            "attributeTypeCode": "Validation",
            "operatorCode": "EQ",
            "operand": "GI",
            "smartGroupId": 147258369
          }
        ]
      },
      "code": "ZPVWTAROW20Q",
      "statusCode": "Activated",
      "startDate": "",
      "endDate": "",
      "useTypeCode": "Single",
```



```
"useLimit": 1,
"currentUse": 1,
"couponUses": [
  {
    "useDateTime": "",
    "statusCode": "Used",
    "channelCode": "System",
    "alias": "",
    "useCount": 1,
    "couponUseAttributes": [
      {
        "couponUseAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
```




14.14.4. Patch Member Redemption

PATCH

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/{memberRedemptionId}>

Try it

Performs a partial update (PATCH) on a Member Redemption. Only the status , expectedShipDate, invoiceNumber and transactionAmount fields can be modified.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-redemption:patch

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberRedemptionId integer<int64> *

Format: int64

Body Parameters

▾ body MemberRedemption

> redemptionItemCode string

> referenceId string

> customData CustomData

> quantity integer<int32>

Format: int32

> currencyValue number

> partnerCode string

> expectedShipDate string<date>

mm/dd/yyyy



> orderAddress MemberRedemptionOrderAddress

> fraudConfirmation boolean

> status string

> transactionAmount number

> invoiceNumber string

> lineItems MemberRedemptionLineItem[]



Responses

▼ ● 200 Success

▼ MemberRedemption

- > memberRedemptionId object
- > memberIdentifier string
- > redemptionItemCode string
- > redemptionItemDescription string
- > referenceId string
- > redemptionDate string<date>
- > customData CustomData
- > quantity integer<int32>
- > currencyValue number
- > partnerCode string
- > expectedShipDate string<date>
- > orderAddress MemberRedemptionOrderAddress
- > bonuses Transaction[]
- > coupons MemberCoupon[]
- > relatedRedemptionId integer<int64>
- > fraudRiskScore number
- > fraudConfirmation boolean
- > status string
- > redemptionRuleCode string
- > transactionAmount number
- > invoiceNumber string
- > lineItems MemberRedemptionLineItem[]
- > poolFunding RedemptionPoolContribution[]
- > poolCancellations RedemptionPoolContribution[]

▼ ● 400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request PATCH \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/{memberRedemptionId} \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "orderAddress": {  
      "address1": "",  
      "city": "",  
      "stateCode": "",  
      "zip": "",  
      "countryCode": ""  
    }  
  }'
```



Responses

● 200



```
// Success
{
  "memberRedemptionId": {},
  "memberIdentifier": "",
  "redemptionItemCode": "GIFT-CARD-50",
  "redemptionItemDescription": "50$ Gift Card",
  "referenceId": 12345,
  "redemptionDate": "2020-01-01",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "quantity": 10,
  "currencyValue": 100,
  "partnerCode": "PARTNER123",
  "expectedShipDate": "2020-01-01",
  "orderAddress": {
    "name": "",
    "address1": "",
    "address2": "",
    "address3": "",
    "city": "",
    "stateCode": "",
    "zip": "",
    "countryCode": "",
    "emailAddress": "",
    "phoneNumber": ""
  },
}
```



```
"bonuses": [  
  {  
    "transactionId": "",  
    "transactionGroupId": "",  
    "transactionGroupReferenceId": "",  
    "parentTransactionId": "",  
    "linkedTransactionId": "",  
    "referenceId": "",  
    "memberIdentifier": "",  
    "accountNumber": "",  
    "memberRedemptionId": "1",  
    "transactionDateTime": "",  
    "partnerName": "",  
    "partnerProgramCode": "",  
    "description": "",  
    "typeCode": "",  
    "typeDescription": "",  
    "subTypeCode": "",  
    "postDate": "",  
    "originationDate": "",  
    "retro": false,  
    "edited": false,  
    "sourceCode": "",  
    "statusCode": "",  
    "updateDateTime": "",  
    "earningRuleDescription": "",  
    "customData": {  
      "ace01": "",  
      "ace01Description": "",  
      "ace02": "",  
      "ace02Description": "",  
      "ace03": "",  
      "ace03Description": "",  
      "ace04": "",  
      "ace04Description": "",  
      "ace05": "",  
      "ace05Description": "",  
      "ace06": "",  
      "ace06Description": "",  
      "ace07": "",  
      "ace07Description": "",  
      "ace08": "",  
      "ace08Description": "",  
      "ace09": "",  
      "ace09Description": "",  
      "ace10": "",  
      "ace10Description": "",  
      "ace11": "",  
      "ace11Description": "",  
      "ace12": "",  
      "ace12Description": "",  
      "ace13": "",  
      "ace13Description": "",  
      "ace14": "",  
      "ace14Description": "",  
      "ace15": "",  
      "ace15Description": ""  
    },  
    },  
    "spend": {  
      }
```



```
"perItem": 0,
"spendAmount": 0,
"currencyTypeCode": "",
"currencyCategoryCode": "",
"isoCurrencyCode": "",
"currencyConversionRatio": "",
"quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
  "totalRevenueAmount": 0,
  "totalRevenueAmountIntl": 0,
  "loyaltyEligibleAmount": 0,
  "loyaltyEligibleAmountIntl": 0,
```



```
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountInt1": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueInt1": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueInt1": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
"transactionAdjustment": {
  "comments": "",
  "referenceId": "",
  "reasonCode": ""
},
"transactionBonus": {
  "bonusCode": "",
  "bonusType": ""
},
"transactionPartner": {
  "partnerName": "",
  "partnerCurrencyCategoryCode": ""
},
"transactionPromotion": {
  "promotionCode": ""
},
"transactionTransfer": {
  "fromMemberAccountNumber": "A234567890",
  "fromMemberName": "Marcel Rufus",
  "toMemberAccountNumber": "B987654321",
```




```
    "toMemberName": "Janel Liesje",
    "transferredBy": "A0123456789"
  },
  "mergeDetails": {
    "fromMemberAccountNumber": "A234567890"
  },
  "transactionRelationships": [
    {
      "type": "B2B_SUCCESSOR",
      "key": "EB-1234",
      "accountNumber": ""
    }
  ]
}
],
"coupons": [
  {
    "memberCouponId": 147258369,
    "memberIdentifier": "",
    "coupon": {
      "couponId": 147258369,
      "couponRequestId": 147258370,
      "offer": {
        "offerId": 147258369,
        "offerTypeCode": "Discount",
        "code": "ALON2020",
        "name": "Free night",
        "description": "Award double points",
        "featureVersion": 2,
        "statusCode": "Enabled",
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
          {
            "offerAttributeId": 147258369,
            "attribute": {
              "attributeId": 147258369,
              "typeCode": "Validation",
              "statusCode": "Enabled",
              "name": "AccountNumber",
              "description": "Account Number",
              "dataTypeCode": "open"
            },
            "defaultValue": "Standard",
            "requireOnUse": true,
            "requireOnHold": true,
            "includeOnResponse": false,
            "attributeTypeCode": "Validation",
            "operatorCode": "EQ",
            "operand": "GI",
            "smartGroupId": 147258369
          }
        ],
        "default": {
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
          "description": "Account Number",
          "dataTypeCode": "open"
        }
      }
    }
  ],
  "code": "ZPVWTAROW20Q",
  "statusCode": "Activated",
  "startDate": "",
  "endDate": "",
  "useTypeCode": "Single",
}
```



```
"useLimit": 1,
"currentUse": 1,
"couponUses": [
  {
    "useDateTime": "",
    "statusCode": "Used",
    "channelCode": "System",
    "alias": "",
    "useCount": 1,
    "couponUseAttributes": [
      {
        "couponUseAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
```



14.14.5. Modify Member Redemption

PUT

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/{memberRedemptionId}

Try it ▶

Modify an existing Member Redemption.
This service cancels the existing redemption(and related coupons) and issues a new one with the details supplied.

Note that once modified the existing memberRedemptionId passed in the request will be canceled and the memberRedemptionId in the response should be used as a reference for this redemption.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 2 ▼

Required Scopes: api:member-redemption:create, api:member-redemption:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberRedemptionId integer<int64> *

Format: int64

Query Parameters

> createCouponUse boolean

> holdCoupon boolean

> allowOverRedeem boolean

> allowPoolRedeem boolean

> async boolean

☐

☐

☐

☐

☐



Body Parameters

▼

body

MemberRedemptionRequest

Description

Member Redemption Request Object

>

redemptionItemCode

string

>

redemptionDate

string<date>

mm/dd/yyyy📅

>

referenceId

string

>

quantity

integer<int32> *

Format: int32

>

providedAmount

number

>

transactionAmount

number

>

externalCode

string

>

offerCriteria

map<string, string>

>

orderAddress

MemberRedemptionOrderAddress

>

couponUse

CouponUseRequest

>

lineItems

MemberRedemptionLineItem[]

>

status

string

>

expectedShipDate

string<date>

mm/dd/yyyy📅

>

invoiceNumber

string

Responses

▼

● 200

Success



▼ MemberRedemption

- > memberRedemptionId object
- > memberIdentifier string
- > redemptionItemCode string
- > redemptionItemDescription string
- > referenceId string
- > redemptionDate string<date>
- > customData CustomData
- > quantity integer<int32>
- > currencyValue number
- > partnerCode string
- > expectedShipDate string<date>
- > orderAddress MemberRedemptionOrderAddress
- > bonuses Transaction[]
- > coupons MemberCoupon[]
- > relatedRedemptionId integer<int64>
- > fraudRiskScore number
- > fraudConfirmation boolean
- > status string
- > redemptionRuleCode string
- > transactionAmount number
- > invoiceNumber string
- > lineItems MemberRedemptionLineItem[]
- > poolFunding RedemptionPoolContribution[]
- > poolCancellations RedemptionPoolContribution[]

▼ ● 202 Accepted



▼ AsyncTask

- > **asyncTaskId** integer<int64>
- > **taskTypeCode** string
- > **taskCode** string
- > **requestDate** string<date-time>
- > **requestData** object
- > **statusCode** string
- > **responseDate** string<date-time>
- > **responseData** object
- > **fileName** string
- > **lineNumber** integer
- > **fileData** string

▼ ● 400 Bad request

▼ ErrorResponse

- > **errors** ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request PUT \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/{memberRedemptionId}?createCoupon' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "quantity": "",
    "orderAddress": {
      "address1": "",
      "city": "",
      "stateCode": "",
      "zip": "",
      "countryCode": ""
    }
  }'
```



Responses

● 200



```
// Success
{
  "memberRedemptionId": {},
  "memberIdentifier": "",
  "redemptionItemCode": "GIFT-CARD-50",
  "redemptionItemDescription": "50$ Gift Card",
  "referenceId": 12345,
  "redemptionDate": "2020-01-01",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "quantity": 10,
  "currencyValue": 100,
  "partnerCode": "PARTNER123",
  "expectedShipDate": "2020-01-01",
  "orderAddress": {
    "name": "",
    "address1": "",
    "address2": "",
    "address3": "",
    "city": "",
    "stateCode": "",
    "zip": "",
    "countryCode": "",
    "emailAddress": "",
    "phoneNumber": ""
  }
},
```




```
"bonuses": [  
  {  
    "transactionId": "",  
    "transactionGroupId": "",  
    "transactionGroupReferenceId": "",  
    "parentTransactionId": "",  
    "linkedTransactionId": "",  
    "referenceId": "",  
    "memberIdentifier": "",  
    "accountNumber": "",  
    "memberRedemptionId": "1",  
    "transactionDateTime": "",  
    "partnerName": "",  
    "partnerProgramCode": "",  
    "description": "",  
    "typeCode": "",  
    "typeDescription": "",  
    "subTypeCode": "",  
    "postDate": "",  
    "originationDate": "",  
    "retro": false,  
    "edited": false,  
    "sourceCode": "",  
    "statusCode": "",  
    "updateDateTime": "",  
    "earningRuleDescription": "",  
    "customData": {  
      "ace01": "",  
      "ace01Description": "",  
      "ace02": "",  
      "ace02Description": "",  
      "ace03": "",  
      "ace03Description": "",  
      "ace04": "",  
      "ace04Description": "",  
      "ace05": "",  
      "ace05Description": "",  
      "ace06": "",  
      "ace06Description": "",  
      "ace07": "",  
      "ace07Description": "",  
      "ace08": "",  
      "ace08Description": "",  
      "ace09": "",  
      "ace09Description": "",  
      "ace10": "",  
      "ace10Description": "",  
      "ace11": "",  
      "ace11Description": "",  
      "ace12": "",  
      "ace12Description": "",  
      "ace13": "",  
      "ace13Description": "",  
      "ace14": "",  
      "ace14Description": "",  
      "ace15": "",  
      "ace15Description": ""  
    },  
    },  
    "spend": {  
      "ace01": "",  
      "ace01Description": "",  
      "ace02": "",  
      "ace02Description": "",  
      "ace03": "",  
      "ace03Description": "",  
      "ace04": "",  
      "ace04Description": "",  
      "ace05": "",  
      "ace05Description": "",  
      "ace06": "",  
      "ace06Description": "",  
      "ace07": "",  
      "ace07Description": "",  
      "ace08": "",  
      "ace08Description": "",  
      "ace09": "",  
      "ace09Description": "",  
      "ace10": "",  
      "ace10Description": "",  
      "ace11": "",  
      "ace11Description": "",  
      "ace12": "",  
      "ace12Description": "",  
      "ace13": "",  
      "ace13Description": "",  
      "ace14": "",  
      "ace14Description": "",  
      "ace15": "",  
      "ace15Description": ""  
    }  
  }  
]
```



```
"perItem": 0,
"spendAmount": 0,
"currencyTypeCode": "",
"currencyCategoryCode": "",
"isoCurrencyCode": "",
"currencyConversionRatio": "",
"quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
  "totalRevenueAmount": 0,
  "totalRevenueAmountIntl": 0,
  "loyaltyEligibleAmount": 0,
  "loyaltyEligibleAmountIntl": 0,
```



```
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountInt1": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueInt1": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueInt1": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
"transactionAdjustment": {
  "comments": "",
  "referenceId": "",
  "reasonCode": ""
},
"transactionBonus": {
  "bonusCode": "",
  "bonusType": ""
},
"transactionPartner": {
  "partnerName": "",
  "partnerCurrencyCategoryCode": ""
},
"transactionPromotion": {
  "promotionCode": ""
},
"transactionTransfer": {
  "fromMemberAccountNumber": "A234567890",
  "fromMemberName": "Marcel Rufus",
  "toMemberAccountNumber": "B987654321",
```



```
    "toMemberName": "Janel Liesje",
    "transferredBy": "A0123456789"
  },
  "mergeDetails": {
    "fromMemberAccountNumber": "A234567890"
  },
  "transactionRelationships": [
    {
      "type": "B2B_SUCCESSOR",
      "key": "EB-1234",
      "accountNumber": ""
    }
  ]
}
],
"coupons": [
  {
    "memberCouponId": 147258369,
    "memberIdentifier": "",
    "coupon": {
      "couponId": 147258369,
      "couponRequestId": 147258370,
      "offer": {
        "offerId": 147258369,
        "offerTypeCode": "Discount",
        "code": "ALON2020",
        "name": "Free night",
        "description": "Award double points",
        "featureVersion": 2,
        "statusCode": "Enabled",
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
          {
            "offerAttributeId": 147258369,
            "attribute": {
              "attributeId": 147258369,
              "typeCode": "Validation",
              "statusCode": "Enabled",
              "name": "AccountNumber",
              "description": "Account Number",
              "dataTypeCode": "open"
            },
            "defaultValue": "Standard",
            "requireOnUse": true,
            "requireOnHold": true,
            "includeOnResponse": false,
            "attributeTypeCode": "Validation",
            "operatorCode": "EQ",
            "operand": "GI",
            "smartGroupId": 147258369
          }
        ],
        "default": {
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
          "description": "Account Number",
          "dataTypeCode": "open"
        }
      }
    }
  ],
  "code": "ZPVWTAROW20Q",
  "statusCode": "Activated",
  "startDate": "",
  "endDate": "",
  "useTypeCode": "Single",
```



```
"useLimit": 1,
"currentUse": 1,
"couponUses": [
  {
    "useDateTime": "",
    "statusCode": "Used",
    "channelCode": "System",
    "alias": "",
    "useCount": 1,
    "couponUseAttributes": [
      {
        "couponUseAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
```



14.14.6. Cancel Member Redemption

DELETE

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/{memberRedemptionId}

Try it ▶

Cancel a member's redemption

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-redemption:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> memberRedemptionId integer<int64> *

Format: int64

Query Parameters

> doNotRedeposit boolean

Responses

▼ ● 200 Success



▼ MemberRedemption

- > memberRedemptionId object
- > memberIdentifier string
- > redemptionItemCode string
- > redemptionItemDescription string
- > referenceId string
- > redemptionDate string<date>
- > customData CustomData
- > quantity integer<int32>
- > currencyValue number
- > partnerCode string
- > expectedShipDate string<date>
- > orderAddress MemberRedemptionOrderAddress
- > bonuses Transaction[]
- > coupons MemberCoupon[]
- > relatedRedemptionId integer<int64>
- > fraudRiskScore number
- > fraudConfirmation boolean
- > status string
- > redemptionRuleCode string
- > transactionAmount number
- > invoiceNumber string
- > lineItems MemberRedemptionLineItem[]
- > poolFunding RedemptionPoolContribution[]
- > poolCancellations RedemptionPoolContribution[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized



- **403** Forbidden
- **500** Internal error

Request



```
curl --request DELETE \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/{memberRedemptionId}?doNotRedepos  
  --header 'accept: application/json'
```




Responses

● 200



```
// Success
{
  "memberRedemptionId": {},
  "memberIdentifier": "",
  "redemptionItemCode": "GIFT-CARD-50",
  "redemptionItemDescription": "50$ Gift Card",
  "referenceId": 12345,
  "redemptionDate": "2020-01-01",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "quantity": 10,
  "currencyValue": 100,
  "partnerCode": "PARTNER123",
  "expectedShipDate": "2020-01-01",
  "orderAddress": {
    "name": "",
    "address1": "",
    "address2": "",
    "address3": "",
    "city": "",
    "stateCode": "",
    "zip": "",
    "countryCode": "",
    "emailAddress": "",
    "phoneNumber": ""
  }
},
```



```
"bonuses": [  
  {  
    "transactionId": "",  
    "transactionGroupId": "",  
    "transactionGroupReferenceId": "",  
    "parentTransactionId": "",  
    "linkedTransactionId": "",  
    "referenceId": "",  
    "memberIdentifier": "",  
    "accountNumber": "",  
    "memberRedemptionId": "1",  
    "transactionDateTime": "",  
    "partnerName": "",  
    "partnerProgramCode": "",  
    "description": "",  
    "typeCode": "",  
    "typeDescription": "",  
    "subTypeCode": "",  
    "postDate": "",  
    "originationDate": "",  
    "retro": false,  
    "edited": false,  
    "sourceCode": "",  
    "statusCode": "",  
    "updateDateTime": "",  
    "earningRuleDescription": "",  
    "customData": {  
      "ace01": "",  
      "ace01Description": "",  
      "ace02": "",  
      "ace02Description": "",  
      "ace03": "",  
      "ace03Description": "",  
      "ace04": "",  
      "ace04Description": "",  
      "ace05": "",  
      "ace05Description": "",  
      "ace06": "",  
      "ace06Description": "",  
      "ace07": "",  
      "ace07Description": "",  
      "ace08": "",  
      "ace08Description": "",  
      "ace09": "",  
      "ace09Description": "",  
      "ace10": "",  
      "ace10Description": "",  
      "ace11": "",  
      "ace11Description": "",  
      "ace12": "",  
      "ace12Description": "",  
      "ace13": "",  
      "ace13Description": "",  
      "ace14": "",  
      "ace14Description": "",  
      "ace15": "",  
      "ace15Description": ""  
    },  
    },  
    "spend": {  
      }
```



```
"perItem": 0,
"spendAmount": 0,
"currencyTypeCode": "",
"currencyCategoryCode": "",
"isoCurrencyCode": "",
"currencyConversionRatio": "",
"quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
  "totalRevenueAmount": 0,
  "totalRevenueAmountIntl": 0,
  "loyaltyEligibleAmount": 0,
  "loyaltyEligibleAmountIntl": 0,
```



```
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountInt1": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueInt1": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueInt1": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
"transactionAdjustment": {
  "comments": "",
  "referenceId": "",
  "reasonCode": ""
},
"transactionBonus": {
  "bonusCode": "",
  "bonusType": ""
},
"transactionPartner": {
  "partnerName": "",
  "partnerCurrencyCategoryCode": ""
},
"transactionPromotion": {
  "promotionCode": ""
},
"transactionTransfer": {
  "fromMemberAccountNumber": "A234567890",
  "fromMemberName": "Marcel Rufus",
  "toMemberAccountNumber": "B987654321",
```



```
    "toMemberName": "Janel Liesje",
    "transferredBy": "A0123456789"
  },
  "mergeDetails": {
    "fromMemberAccountNumber": "A234567890"
  },
  "transactionRelationships": [
    {
      "type": "B2B_SUCCESSOR",
      "key": "EB-1234",
      "accountNumber": ""
    }
  ]
}
],
"coupons": [
  {
    "memberCouponId": 147258369,
    "memberIdentifier": "",
    "coupon": {
      "couponId": 147258369,
      "couponRequestId": 147258370,
      "offer": {
        "offerId": 147258369,
        "offerTypeCode": "Discount",
        "code": "ALON2020",
        "name": "Free night",
        "description": "Award double points",
        "featureVersion": 2,
        "statusCode": "Enabled",
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
          {
            "offerAttributeId": 147258369,
            "attribute": {
              "attributeId": 147258369,
              "typeCode": "Validation",
              "statusCode": "Enabled",
              "name": "AccountNumber",
              "description": "Account Number",
              "dataTypeCode": "open"
            },
            "defaultValue": "Standard",
            "requireOnUse": true,
            "requireOnHold": true,
            "includeOnResponse": false,
            "attributeTypeCode": "Validation",
            "operatorCode": "EQ",
            "operand": "GI",
            "smartGroupId": 147258369
          }
        ],
        "default": {
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
          "description": "Account Number",
          "dataTypeCode": "open"
        }
      }
    },
    "code": "ZPVWTAROW20Q",
    "statusCode": "Activated",
    "startDate": "",
    "endDate": "",
    "useTypeCode": "Single",
```



```
"useLimit": 1,
"currentUse": 1,
"couponUses": [
  {
    "useDateTime": "",
    "statusCode": "Used",
    "channelCode": "System",
    "alias": "",
    "useCount": 1,
    "couponUseAttributes": [
      {
        "couponUseAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
```



14.14.7. Get Member Redemption By Reference Id

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/referenceId;{referenceId}

Try it ▶

Retrieve a member's redemption by reference id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-redemption:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> referenceId string *

Responses

▼ ● 200 Success



▼ MemberRedemption

- > memberRedemptionId object
- > memberIdentifier string
- > redemptionItemCode string
- > redemptionItemDescription string
- > referenceId string
- > redemptionDate string<date>
- > customData CustomData
- > quantity integer<int32>
- > currencyValue number
- > partnerCode string
- > expectedShipDate string<date>
- > orderAddress MemberRedemptionOrderAddress
- > bonuses Transaction[]
- > coupons MemberCoupon[]
- > relatedRedemptionId integer<int64>
- > fraudRiskScore number
- > fraudConfirmation boolean
- > status string
- > redemptionRuleCode string
- > transactionAmount number
- > invoiceNumber string
- > lineItems MemberRedemptionLineItem[]
- > poolFunding RedemptionPoolContribution[]
- > poolCancellations RedemptionPoolContribution[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized



- **403** Forbidden
- **500** Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/referenceId;{referenceId} \
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "memberRedemptionId": {},
  "memberIdentifier": "",
  "redemptionItemCode": "GIFT-CARD-50",
  "redemptionItemDescription": "50$ Gift Card",
  "referenceId": 12345,
  "redemptionDate": "2020-01-01",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "quantity": 10,
  "currencyValue": 100,
  "partnerCode": "PARTNER123",
  "expectedShipDate": "2020-01-01",
  "orderAddress": {
    "name": "",
    "address1": "",
    "address2": "",
    "address3": "",
    "city": "",
    "stateCode": "",
    "zip": "",
    "countryCode": "",
    "emailAddress": "",
    "phoneNumber": ""
  }
},
```



```
"bonuses": [  
  {  
    "transactionId": "",  
    "transactionGroupId": "",  
    "transactionGroupReferenceId": "",  
    "parentTransactionId": "",  
    "linkedTransactionId": "",  
    "referenceId": "",  
    "memberIdentifier": "",  
    "accountNumber": "",  
    "memberRedemptionId": "1",  
    "transactionDateTime": "",  
    "partnerName": "",  
    "partnerProgramCode": "",  
    "description": "",  
    "typeCode": "",  
    "typeDescription": "",  
    "subTypeCode": "",  
    "postDate": "",  
    "originationDate": "",  
    "retro": false,  
    "edited": false,  
    "sourceCode": "",  
    "statusCode": "",  
    "updateDateTime": "",  
    "earningRuleDescription": "",  
    "customData": {  
      "ace01": "",  
      "ace01Description": "",  
      "ace02": "",  
      "ace02Description": "",  
      "ace03": "",  
      "ace03Description": "",  
      "ace04": "",  
      "ace04Description": "",  
      "ace05": "",  
      "ace05Description": "",  
      "ace06": "",  
      "ace06Description": "",  
      "ace07": "",  
      "ace07Description": "",  
      "ace08": "",  
      "ace08Description": "",  
      "ace09": "",  
      "ace09Description": "",  
      "ace10": "",  
      "ace10Description": "",  
      "ace11": "",  
      "ace11Description": "",  
      "ace12": "",  
      "ace12Description": "",  
      "ace13": "",  
      "ace13Description": "",  
      "ace14": "",  
      "ace14Description": "",  
      "ace15": "",  
      "ace15Description": ""  
    },  
    },  
    "spend": {  
  }
```



```
"perItem": 0,
"spendAmount": 0,
"currencyTypeCode": "",
"currencyCategoryCode": "",
"isoCurrencyCode": "",
"currencyConversionRatio": "",
"quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
  "totalRevenueAmount": 0,
  "totalRevenueAmountIntl": 0,
  "loyaltyEligibleAmount": 0,
  "loyaltyEligibleAmountIntl": 0,
```



```
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountInt1": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueInt1": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueInt1": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
"transactionAdjustment": {
  "comments": "",
  "referenceId": "",
  "reasonCode": ""
},
"transactionBonus": {
  "bonusCode": "",
  "bonusType": ""
},
"transactionPartner": {
  "partnerName": "",
  "partnerCurrencyCategoryCode": ""
},
"transactionPromotion": {
  "promotionCode": ""
},
"transactionTransfer": {
  "fromMemberAccountNumber": "A234567890",
  "fromMemberName": "Marcel Rufus",
  "toMemberAccountNumber": "B987654321",
```



```
    "toMemberName": "Janel Liesje",
    "transferredBy": "A0123456789"
  },
  "mergeDetails": {
    "fromMemberAccountNumber": "A234567890"
  },
  "transactionRelationships": [
    {
      "type": "B2B_SUCCESSOR",
      "key": "EB-1234",
      "accountNumber": ""
    }
  ]
},
],
"coupons": [
  {
    "memberCouponId": 147258369,
    "memberIdentifier": "",
    "coupon": {
      "couponId": 147258369,
      "couponRequestId": 147258370,
      "offer": {
        "offerId": 147258369,
        "offerTypeCode": "Discount",
        "code": "ALON2020",
        "name": "Free night",
        "description": "Award double points",
        "featureVersion": 2,
        "statusCode": "Enabled",
        "startDateTime": "",
        "endDateTime": "",
        "offerAttributes": [
          {
            "offerAttributeId": 147258369,
            "attribute": {
              "attributeId": 147258369,
              "typeCode": "Validation",
              "statusCode": "Enabled",
              "name": "AccountNumber",
              "description": "Account Number",
              "dataTypeCode": "open"
            },
            "defaultValue": "Standard",
            "requireOnUse": true,
            "requireOnHold": true,
            "includeOnResponse": false,
            "attributeTypeCode": "Validation",
            "operatorCode": "EQ",
            "operand": "GI",
            "smartGroupId": 147258369
          }
        ],
        "default": {
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
          "description": "Account Number",
          "dataTypeCode": "open"
        }
      }
    },
    "code": "ZPVWTAROW20Q",
    "statusCode": "Activated",
    "startDate": "",
    "endDate": "",
    "useTypeCode": "Single",
```




```
"useLimit": 1,
"currentUse": 1,
"couponUses": [
  {
    "useDateTime": "",
    "statusCode": "Used",
    "channelCode": "System",
    "alias": "",
    "useCount": 1,
    "couponUseAttributes": [
      {
        "couponUseAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
```



14.14.8. Get Member Redemption Aggregates

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/aggregates`

Try it ▶

Search for member's redemption aggregates

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-redemption:read

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** string *

Query Parameters

> **fromDate** string<date>

mm/dd/yyyy



> **toDate** string<date>

mm/dd/yyyy



Responses

✓ ● 200 Success

▼ TransactionCurrency[]

> **amount** number

> **currencyTypeCode** string

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/redemptions/aggregates \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "amount": 1000,
    "currencyTypeCode": "USD"
  }
]
```



14.15. member-tier

14.15.1. Get Member Tiers

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/tiers`

Try it ▶

Retrieve a member's tiers. This will return current and all future tiers.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:member-tier:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Query Parameters

> `includeRolloverNights` `boolean`

Responses

✓ ● 200 Success

▾ MemberTiersResponse

> `pastTiers` `MemberTier[]`> `currentTier` `MemberTier`> `projectedTier` `MemberTier`> `earningTier` `MemberFutureTier`> `aspirationTiers` `MemberFutureTier[]`> `nextTierRolloverDate` `string<date>`

✓ ● 400 Bad request

▾ ErrorResponse

> `errors` `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/tiers?includeRolloverNights=boolean' \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "pastTiers": [
    {
      "tierCode": "B",
      "description": "Base",
      "rank": 10,
      "typeCode": "Earned",
      "startDate": "2020-01-01",
      "endDate": "2020-12-31",
      "grandFathered": false,
      "earned": true,
      "lifetime": false,
      "rolloverNights": 10,
      "assignmentReason": "bonus",
      "assignmentReasonObjectExternalReference": "",
      "assignmentReasonObjectDescription": "Adjustment Bonus 1000 Points",
      "createdBy": "",
      "sourceCouponUse": {
        "couponCode": "113456816",
        "offerCode": "Q1UP",
        "offerName": "Q1 upgrades",
        "startDate": "2025-01-15",
        "endDate": "2026-01-15"
      }
    }
  ],
  "currentTier": {
    "tierCode": "B",
    "description": "Base",
    "rank": 10,
    "typeCode": "Earned",
    "startDate": "2020-01-01",
    "endDate": "2020-12-31",
    "grandFathered": false,
    "earned": true,
    "lifetime": false,
    "rolloverNights": 10,
    "assignmentReason": "bonus",
    "assignmentReasonObjectExternalReference": "",
    "assignmentReasonObjectDescription": "Adjustment Bonus 1000 Points",
    "createdBy": "",
    "sourceCouponUse": {
      "couponCode": "113456816",
      "offerCode": "Q1UP",
      "offerName": "Q1 upgrades",
      "startDate": "2025-01-15",
      "endDate": "2026-01-15"
    }
  },
  "projectedTier": {
    "tierCode": "B",
    "description": "Base",
    "rank": 10,
    "typeCode": "Earned",
    "startDate": "2020-01-01",
```



```
"endDate": "2020-12-31",  
"grandFathered": false,  
"earned": true,  
"lifetime": false,
```




14.15.2. Update Member Tier

POST

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/tiers

Try it ▶

Updates a member's current Tier.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-tier:adjust

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> migrate boolean

> async boolean

Body Parameters

▼ body MemberTierRequest *

> tierCode string *

> tierYear integer<int32> *

> comments string *

> forceTierChange boolean

> assignmentReason string<bonus | creditcard | management | manual | promotio...

Format: int32

Select... ▾

Responses

▼ • 200 Success



▼ MemberTier

- > tierCode string *
- > description string
- > rank integer<int32> *
- > typeCode string
- > startDate string<date>
- > endDate string<date>
- > grandFathered boolean
- > earned boolean
- > lifetime boolean
- > rolloverNights integer
- > assignmentReason string<bonus | creditcard | management | manual | promo...
- > assignmentReasonObjectExternalReference string
- > assignmentReasonObjectDescription string
- > createdBy string
- > sourceCouponUse MemberTierCouponUse

▼ ● 202 Accepted



▼ AsyncTask

```
> asyncTaskId integer<int64>
> taskTypeCode string
> taskCode string
> requestDate string<date-time>
> requestData object
> statusCode string
> responseDate string<date-time>
> responseData object
> fileName string
> lineNumber integer
> fileData string
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/tiers?migrate=boolean&async=boolean' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "tierCode": "B",
    "tierYear": 2027,
    "comments": "Tier manually updated to Platinum because of xyz reason.",
    "forceTierChange": true
  }'
```



Responses

● 200

```
// Success
{
  "tierCode": "B",
  "description": "Base",
  "rank": 10,
  "typeCode": "Earned",
  "startDate": "2020-01-01",
  "endDate": "2020-12-31",
  "grandFathered": false,
  "earned": true,
  "lifetime": false,
  "rolloverNights": 10,
  "assignmentReason": "bonus",
  "assignmentReasonObjectExternalReference": "",
  "assignmentReasonObjectDescription": "Adjustment Bonus 1000 Points",
  "createdBy": "",
  "sourceCouponUse": {
    "couponCode": "113456816",
    "offerCode": "Q1UP",
    "offerName": "Q1 upgrades",
    "startDate": "2025-01-15",
    "endDate": "2026-01-15"
  }
}
```



14.15.3. Rollover Member

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/rollover`

Try it ▶

Trigger a member's tier rollover

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-tier:adjust

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/rollover \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.16. member-transaction

14.16.1. Post Earning

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/earnings`

Try it

Submits a transaction through the earnings process. It will go through the entire process synchronously. If there is a failure, the transaction will be rolled back.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: `api:transaction:submit`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Query Parameters

> `migrate` `boolean`> `async` `boolean`> `isRetro` `boolean`



Body Parameters

▼

body

TransactionRequest

*

Description

Transaction Request

>

referenceId

string

>

member

TransactionMember

>

transactionDateTime

string<date-time>

mm/dd/yyyy, --:-- --📅

>

description

string

>

originationDate

string<date>

mm/dd/yyyy📅

>

awardNumber

string

>

providedAmount

number

>

sourceCode

string

>

migrateStatusCode

string

>

customData

CustomData

>

spend

TransactionSpend

>

hotelTransaction

HotelTransaction

>

retailTransaction

RetailTransaction

>

travelTransaction

TravelTransaction

>

transactionPartner

TransactionPartnerEarn

>

transactionMemberPaymentCardAccount

TransactionMemberPaymentCardAccount

>

transactionReference

map<string, object>

>

useCouponCodes

string[]

+

Responses

▼

● 200

Success



▼ Transaction

- > transactionId string
- > transactionGroupId string
- > transactionGroupReferenceId string
- > parentTransactionId string
- > linkedTransactionId string
- > referenceId string
- > memberIdentifier string
- > accountNumber string
- > memberRedemptionId string
- > transactionDateTime string<date-time>
- > partnerName string
- > partnerProgramCode string
- > description string
- > typeCode string
- > typeDescription string
- > subTypeCode string
- > postDate string<date>
- > originationDate string<date>
- > retro boolean
- > edited boolean
- > sourceCode string
- > statusCode string
- > updateDateTime string<date-time>
- > earningRuleDescription string
- > customData CustomData
- > spend TransactionSpend
- > earn TransactionEarn
- > redemption TransactionRedemption



- > **expiredCurrency** TransactionExpiredCurrency
- > **hotelTransaction** HotelTransaction
- > **travelTransaction** TravelTransaction
- > **transactionReference** map<string, object>
- > **cancelable** boolean
- > **editable** boolean
- > **createUserId** integer<int64>
- > **createUserName** string
- > **updateUserId** integer<int64>
- > **updateUserName** string
- > **memberCommentId** integer<int64>
- > **cancellationSubReason** string
- > **businessEligible** boolean
- > **businessAccountNumber** string
- > **awardNumber** string
- > **baseAccountNumber** string
- > **transactionAdjustment** TransactionAdjustment
- > **transactionBonus** TransactionBonus
- > **transactionPartner** TransactionPartner
- > **transactionPromotion** TransactionPromotion
- > **transactionTransfer** TransactionTransfer
- > **mergeDetails** MergeDetail
- > **transactionRelationships** TransactionRelationship[]

▼ ● 202 Accepted



▼ AsyncTask

> asyncTaskId integer<int64>

> taskTypeCode string

> taskCode string

> requestDate string<date-time>

> requestData object

> statusCode string

> responseDate string<date-time>

> responseData object

> fileName string

> lineNumber integer

> fileData string

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request POST \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/earnings?migrate=boolean&async=boolean&isRetr\  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "transactionId": "",
  "transactionGroupId": "",
  "transactionGroupReferenceId": "",
  "parentTransactionId": "",
  "linkedTransactionId": "",
  "referenceId": "",
  "memberIdentifier": "",
  "accountNumber": "",
  "memberRedemptionId": "1",
  "transactionDateTime": "",
  "partnerName": "",
  "partnerProgramCode": "",
  "description": "",
  "typeCode": "",
  "typeDescription": "",
  "subTypeCode": "",
  "postDate": "",
  "originationDate": "",
  "retro": false,
  "edited": false,
  "sourceCode": "",
  "statusCode": "",
  "updateDateTime": "",
  "earningRuleDescription": "",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
```



```
    "ace15Description": ""
  },
  "spend": {
    "perItem": 0,
    "spendAmount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "isoCurrencyCode": "",
    "currencyConversionRatio": "",
    "quantity": ""
  },
  "earn": {
    "amount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "earningsLimited": false,
    "tierQualifying": false,
    "transactionCounters": [
      {
        "counterTypeCode": "",
        "counterValue": 0
      }
    ]
  },
  "redemption": {
    "redemptionItemCode": "",
    "currencyRequired": 0,
    "minimumQuantity": "",
    "maximumQuantity": "",
    "redeemIncrement": "",
    "faceValue": 0,
    "amountProvided": false,
    "poolName": "PL-16",
    "lineItems": [
      {
        "description": "",
        "itemCode": "",
        "quantity": "",
        "amount": 0
      }
    ]
  },
  "expiredCurrency": {
    "expiredAmount": 0,
    "expiredDate": "",
    "currencyTypeCode": "",
    "currencyCategoryCode": ""
  },
  "hotelTransaction": {
    "checkInDate": "",
    "checkOutDate": "",
    "eligibleStays": "",
    "eligibleCappedNights": "",
    "eligibleNights": "",
    "roomNumber": "",
    "roomType": "",
    "roomRate": 0,
    "roomRateIntl": 0,
    "totalRevenueAmount": 0,
```



```
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",
```



14.16.2. Post Transaction Group for Earnings Processing

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/earnings-group>

Try it

This will submit a transaction group through the earnings process.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:transaction:submit

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> validateOnly boolean

☐

Body Parameters

▼ body TransactionGroupRequest *

Description

Transaction Group Request

> transactionGroupReferenceId string *

> activityStartDateTime string<date-time>

mm/dd/yyyy, --:-- --

> activityEndDateTime string<date-time>

mm/dd/yyyy, --:-- --

> originationDate string<date>

mm/dd/yyyy

> totalSpendAmount number

> totalQualifiedSpendAmount number

> customData CustomData

> transactions TransactionRequest[]

Responses

▼ ● 200 Success



▼ TransactionGroup

- > transactionGroupIdd string *
- > transactionGroupReferenceId string *
- > transactionGroupDescription string
- > transactionGroupDateTime string<date-time>
- > transactionGroupStatusCode string
- > activityStartDateTime string<date-time>
- > activityEndDateTime string<date-time>
- > originationDate string<date>
- > totalSpendAmount number
- > totalQualifiedSpendAmount number
- > statusCode string
- > postDate string<date>
- > typeCode string
- > typeDescription string
- > customData CustomData
- > transactionGroupSpend TransactionGroupSpend[]
- > transactionGroupEarn TransactionGroupEarn[]
- > transactionGroupReference map<string, object>
- > transactions Transaction[]
- > cancelable boolean
- > editable boolean
- > createUserId integer<int64>
- > createUserName string
- > updateUserId integer<int64>
- > updateUserName string

▼ ● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/earnings-group?validateOnly=true' \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "transactionGroupReferenceId": ""  
  }'
```



Responses

● 200



```
// Success
{
  "transactionGroupId": "",
  "transactionGroupReferenceId": "",
  "transactionGroupDescription": "",
  "transactionGroupDateTime": "",
  "transactionGroupStatusCode": "",
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "originationDate": "",
  "totalSpendAmount": 0,
  "totalQualifiedSpendAmount": 0,
  "statusCode": "",
  "postDate": "",
  "typeCode": "",
  "typeDescription": "",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "transactionGroupSpend": [
    {
      "spendAmount": 0,
      "currencyTypeCode": "",
      "currencyCategoryCode": "",
      "currencyCategoryDescription": "",
      "isoCurrencyCode": ""
    }
  ]
}
```



```
],
"transactionGroupEarn": [
  {
    "amount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "currencyCategoryDescription": ""
  }
],
"transactionGroupReference": "",
"transactions": [
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": ""
    }
  }
]
```



```
"ace12": "",
"ace12Description": "",
"ace13": "",
"ace13Description": "",
"ace14": "",
"ace14Description": "",
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": ""
}
```



```
"eligibleCappedNights": "",
"eligibleNights": "",
"roomNumber": "",
"roomType": "",
"roomRate": 0,
"roomRateIntl": 0,
"totalRevenueAmount": 0,
"totalRevenueAmountIntl": 0,
"loyaltyEligibleAmount": 0,
"loyaltyEligibleAmountIntl": 0,
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountIntl": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueIntl": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueIntl": 0,
"checkNumber": ""
```



14.16.3. Get Member Transactions

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions`

Try it

Search for member's transactions, redemptions, and others

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:member-transaction:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Query Parameters

> `fromDate` `string<date>`

mm/dd/yyyy

> `toDate` `string<date>`

mm/dd/yyyy

> `type` `string`

Select...

> `includeMergeDetails` `boolean`☐> `includeRelationships` `boolean`☐

Responses

▾ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



- > expiredCurrency TransactionExpiredCurrency
- > hotelTransaction HotelTransaction
- > travelTransaction TravelTransaction
- > transactionReference map<string, object>
- > cancelable boolean
- > editable boolean
- > createUserId integer<int64>
- > createUserName string
- > updateUserId integer<int64>
- > updateUserName string
- > memberCommentId integer<int64>
- > cancellationSubReason string
- > businessEligible boolean
- > businessAccountNumber string
- > awardNumber string
- > baseAccountNumber string
- > transactionAdjustment TransactionAdjustment
- > transactionBonus TransactionBonus
- > transactionPartner TransactionPartner
- > transactionPromotion TransactionPromotion
- > transactionTransfer TransactionTransfer
- > mergeDetails MergeDetail
- > transactionRelationships TransactionRelationship[]

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions?includeMergeDetails=boolean&incl  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```



```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```



14.16.4. Get Member Transaction

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId}

Try it ▶

Get a member's transaction

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-transaction:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionId string *

Responses

▼ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



```
> expiredCurrency TransactionExpiredCurrency
> hotelTransaction HotelTransaction
> travelTransaction TravelTransaction
> transactionReference map<string, object>
> cancelable boolean
> editable boolean
> createUserId integer<int64>
> createUserName string
> updateUserId integer<int64>
> updateUserName string
> memberCommentId integer<int64>
> cancellationSubReason string
> businessEligible boolean
> businessAccountNumber string
> awardNumber string
> baseAccountNumber string
> transactionAdjustment TransactionAdjustment
> transactionBonus TransactionBonus
> transactionPartner TransactionPartner
> transactionPromotion TransactionPromotion
> transactionTransfer TransactionTransfer
> mergeDetails MergeDetail
> transactionRelationships TransactionRelationship[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId} \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```



```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```



14.16.5. Edit Member Transaction

PATCH

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId}

Try it ▶

Edit a member's transaction

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-transaction:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionId string *

Query Parameters

> validateOnly boolean

☐

Body Parameters

▾ body MemberTransactionUpdateRequest *

> referenceId string

> transactionDateTime string

> hotelTransaction HotelTransaction

> transactionAdjustment object *

Responses

▾ ● 200 Success



▼ EditMemberTransactionResponse

- > totalRevenueAmount number
- > totalRevenueAmountIntl number
- > loyaltyEligibleAmount number
- > loyaltyEligibleAmountIntl number
- > loyaltyIneligibleAmount number
- > loyaltyIneligibleAmountIntl number
- > basePoints number

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request PATCH \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId}?validateOnly=boo
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "transactionAdjustment": {
      "reasonCode": "RETRO_CREDIT",
      "comments": "Retroactive points credit for qualifying stay"
    }
  }'
```

Responses

● 200

```
// Success
{
  "totalRevenueAmount": 375,
  "totalRevenueAmountIntl": 8500,
  "loyaltyEligibleAmount": 275,
  "loyaltyEligibleAmountIntl": 5250,
  "loyaltyIneligibleAmount": 100,
  "loyaltyIneligibleAmountIntl": 3250,
  "basePoints": 950
}
```




14.16.6. Cancel Member Transaction

DELETE

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId}

Try it ▶

Cancel a member's transaction

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-transaction:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionId string *

Query Parameters

> async boolean

☐

Body Parameters

▼ body CancelCallLog

Description

Cancellation comment details

> commentSubReason string

> comments string

Responses

▼ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



- > expiredCurrency TransactionExpiredCurrency
- > hotelTransaction HotelTransaction
- > travelTransaction TravelTransaction
- > transactionReference map<string, object>
- > cancelable boolean
- > editable boolean
- > createUserId integer<int64>
- > createUserName string
- > updateUserId integer<int64>
- > updateUserName string
- > memberCommentId integer<int64>
- > cancellationSubReason string
- > businessEligible boolean
- > businessAccountNumber string
- > awardNumber string
- > baseAccountNumber string
- > transactionAdjustment TransactionAdjustment
- > transactionBonus TransactionBonus
- > transactionPartner TransactionPartner
- > transactionPromotion TransactionPromotion
- > transactionTransfer TransactionTransfer
- > mergeDetails MergeDetail
- > transactionRelationships TransactionRelationship[]

▼ ● 202 Accepted



▼ AsyncTask

- > asyncTaskId integer<int64>
- > taskTypeCode string
- > taskCode string
- > requestDate string<date-time>
- > requestData object
- > statusCode string
- > responseDate string<date-time>
- > responseData object
- > fileName string
- > lineNumber integer
- > fileData string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request DELETE \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId}?async=boolean' \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```



```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```




14.16.7. Cancel Member Transaction (POST)

POST

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId}/cancel

Try it ▶

Cancel a member's transaction using POST for OAuth compatibility

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-transaction:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionId string *

Query Parameters

> async boolean

☐

Body Parameters

▾ body CancelCallLog *

Description

Cancellation comment details

> commentSubReason string

> comments string

Responses

▾ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



- > expiredCurrency TransactionExpiredCurrency
- > hotelTransaction HotelTransaction
- > travelTransaction TravelTransaction
- > transactionReference map<string, object>
- > cancelable boolean
- > editable boolean
- > createUserId integer<int64>
- > createUserName string
- > updateUserId integer<int64>
- > updateUserName string
- > memberCommentId integer<int64>
- > cancellationSubReason string
- > businessEligible boolean
- > businessAccountNumber string
- > awardNumber string
- > baseAccountNumber string
- > transactionAdjustment TransactionAdjustment
- > transactionBonus TransactionBonus
- > transactionPartner TransactionPartner
- > transactionPromotion TransactionPromotion
- > transactionTransfer TransactionTransfer
- > mergeDetails MergeDetail
- > transactionRelationships TransactionRelationship[]

▼ ● 202 Accepted



▼ AsyncTask

```
> asyncTaskId integer<int64>
> taskTypeCode string
> taskCode string
> requestDate string<date-time>
> requestData object
> statusCode string
> responseDate string<date-time>
> responseData object
> fileName string
> lineNumber integer
> fileData string
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId}/cancel?async=boo
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```



```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```




14.16.8. Clone Member Transaction

POST

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId}/clone/{targetMemberIdentifier}

Try it ▶

Clones an existing transaction from the source member to the target member. The cloned transaction will go through the complete earnings process including validation, bonus rules, and promotions in the context of the target member.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:transaction:clone

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionId string *

> targetMemberIdentifier string *

Query Parameters

> prefix string

> cancelSource boolean

Body Parameters

▼ body CloneMemberTransactionRequest

Description

Optional clone transaction comment details

> memberComment CloneMemberTransactionComment

Responses

▼

● 200 Success



▼ Transaction

- > transactionId string
- > transactionGroupId string
- > transactionGroupReferenceId string
- > parentTransactionId string
- > linkedTransactionId string
- > referenceId string
- > memberIdentifier string
- > accountNumber string
- > memberRedemptionId string
- > transactionDateTime string<date-time>
- > partnerName string
- > partnerProgramCode string
- > description string
- > typeCode string
- > typeDescription string
- > subTypeCode string
- > postDate string<date>
- > originationDate string<date>
- > retro boolean
- > edited boolean
- > sourceCode string
- > statusCode string
- > updateDateTime string<date-time>
- > earningRuleDescription string
- > customData CustomData
- > spend TransactionSpend
- > earn TransactionEarn
- > redemption TransactionRedemption



```
> expiredCurrency TransactionExpiredCurrency
> hotelTransaction HotelTransaction
> travelTransaction TravelTransaction
> transactionReference map<string, object>
> cancelable boolean
> editable boolean
> createUserId integer<int64>
> createUserName string
> updateUserId integer<int64>
> updateUserName string
> memberCommentId integer<int64>
> cancellationSubReason string
> businessEligible boolean
> businessAccountNumber string
> awardNumber string
> baseAccountNumber string
> transactionAdjustment TransactionAdjustment
> transactionBonus TransactionBonus
> transactionPartner TransactionPartner
> transactionPromotion TransactionPromotion
> transactionTransfer TransactionTransfer
> mergeDetails MergeDetail
> transactionRelationships TransactionRelationship[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request POST \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/{transactionId}/clone/{targetMem'  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "transactionId": "",
  "transactionGroupId": "",
  "transactionGroupReferenceId": "",
  "parentTransactionId": "",
  "linkedTransactionId": "",
  "referenceId": "",
  "memberIdentifier": "",
  "accountNumber": "",
  "memberRedemptionId": "1",
  "transactionDateTime": "",
  "partnerName": "",
  "partnerProgramCode": "",
  "description": "",
  "typeCode": "",
  "typeDescription": "",
  "subTypeCode": "",
  "postDate": "",
  "originationDate": "",
  "retro": false,
  "edited": false,
  "sourceCode": "",
  "statusCode": "",
  "updateDateTime": "",
  "earningRuleDescription": "",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
```



```
    "ace15Description": ""
  },
  "spend": {
    "perItem": 0,
    "spendAmount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "isoCurrencyCode": "",
    "currencyConversionRatio": "",
    "quantity": ""
  },
  "earn": {
    "amount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "earningsLimited": false,
    "tierQualifying": false,
    "transactionCounters": [
      {
        "counterTypeCode": "",
        "counterValue": 0
      }
    ]
  },
  "redemption": {
    "redemptionItemCode": "",
    "currencyRequired": 0,
    "minimumQuantity": "",
    "maximumQuantity": "",
    "redeemIncrement": "",
    "faceValue": 0,
    "amountProvided": false,
    "poolName": "PL-16",
    "lineItems": [
      {
        "description": "",
        "itemCode": "",
        "quantity": "",
        "amount": 0
      }
    ]
  },
  "expiredCurrency": {
    "expiredAmount": 0,
    "expiredDate": "",
    "currencyTypeCode": "",
    "currencyCategoryCode": ""
  },
  "hotelTransaction": {
    "checkInDate": "",
    "checkOutDate": "",
    "eligibleStays": "",
    "eligibleCappedNights": "",
    "eligibleNights": "",
    "roomNumber": "",
    "roomType": "",
    "roomRate": 0,
    "roomRateIntl": 0,
    "totalRevenueAmount": 0,
```



```
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",
```




14.16.9. Get Member Transaction By Reference Id

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/referenceId;{referenceId}

Try it ▶

Get a member's transaction by reference id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-transaction:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> referenceId string *

Query Parameters

> includeAdjustments boolean

Responses

▼ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



```
> expiredCurrency TransactionExpiredCurrency
> hotelTransaction HotelTransaction
> travelTransaction TravelTransaction
> transactionReference map<string, object>
> cancelable boolean
> editable boolean
> createUserId integer<int64>
> createUserName string
> updateUserId integer<int64>
> updateUserName string
> memberCommentId integer<int64>
> cancellationSubReason string
> businessEligible boolean
> businessAccountNumber string
> awardNumber string
> baseAccountNumber string
> transactionAdjustment TransactionAdjustment
> transactionBonus TransactionBonus
> transactionPartner TransactionPartner
> transactionPromotion TransactionPromotion
> transactionTransfer TransactionTransfer
> mergeDetails MergeDetail
> transactionRelationships TransactionRelationship[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/referenceId;{referenceId} \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```



```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```






14.16.10. Cancel Member Transaction By Reference Id

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/referenceId;{referenceId}`

Try it

Cancel a member's transaction by reference id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-transaction:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> referenceId string *

Query Parameters

> transactionType string

Select... ▾

Body Parameters

▼ body CancelCallLog

Description

Cancellation comment details

> commentSubReason string

> comments string

Responses

▼ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



```
> expiredCurrency TransactionExpiredCurrency
> hotelTransaction HotelTransaction
> travelTransaction TravelTransaction
> transactionReference map<string, object>
> cancelable boolean
> editable boolean
> createUserId integer<int64>
> createUserName string
> updateUserId integer<int64>
> updateUserName string
> memberCommentId integer<int64>
> cancellationSubReason string
> businessEligible boolean
> businessAccountNumber string
> awardNumber string
> baseAccountNumber string
> transactionAdjustment TransactionAdjustment
> transactionBonus TransactionBonus
> transactionPartner TransactionPartner
> transactionPromotion TransactionPromotion
> transactionTransfer TransactionTransfer
> mergeDetails MergeDetail
> transactionRelationships TransactionRelationship[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request DELETE \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/referenceId;{referenceId} \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```




```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```



14.16.11. Cancel Member Transaction By Reference Id (POST)

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/referenceId/{referenceId}/c>

Try it ▶

Cancel a member's transaction by reference id using POST for OAuth compatibility

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-transaction:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> referenceId string *

Body Parameters

▼ body CancelCallLog *

Description

Cancellation comment details

> commentSubReason string

> comments string

Responses

▼ ● 200 Success



- ▼ Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



```
> expiredCurrency TransactionExpiredCurrency
> hotelTransaction HotelTransaction
> travelTransaction TravelTransaction
> transactionReference map<string, object>
> cancelable boolean
> editable boolean
> createUserId integer<int64>
> createUserName string
> updateUserId integer<int64>
> updateUserName string
> memberCommentId integer<int64>
> cancellationSubReason string
> businessEligible boolean
> businessAccountNumber string
> awardNumber string
> baseAccountNumber string
> transactionAdjustment TransactionAdjustment
> transactionBonus TransactionBonus
> transactionPartner TransactionPartner
> transactionPromotion TransactionPromotion
> transactionTransfer TransactionTransfer
> mergeDetails MergeDetail
> transactionRelationships TransactionRelationship[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/referenceId;{referenceId}/cancel \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```




```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```



14.16.12. Get Member Transaction Groups

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups`

Try it

Search for member's earnings, redemptions, and others aggregated by transaction group

Credentials

OAuth 2.0

See <https://docs.wearephaedon.com/docs/platform-api-authentication>Flow: **Client Credentials**

Client ID

Client Secret

Scopes **1** Required Scopes: `api:member-transaction:read`**Authorize**

Use Your Own Token

Path Parameters

> **memberIdentifier** `string` *

Query Parameters

> **fromDate** `string<date>`

mm/dd/yyyy

> **toDate** `string<date>`

mm/dd/yyyy

> **type** `string`

Select...

> **includeMergeDetails** `boolean`☐> **includeRelationships** `boolean`☐

Responses

 • **200** Success



TransactionGroup[]

- > transactionGroupId string *
- > transactionGroupReferenceId string *
- > transactionGroupDescription string
- > transactionGroupDateTime string<date-time>
- > transactionGroupStatusCode string
- > activityStartDateTime string<date-time>
- > activityEndDateTime string<date-time>
- > originationDate string<date>
- > totalSpendAmount number
- > totalQualifiedSpendAmount number
- > statusCode string
- > postDate string<date>
- > typeCode string
- > typeDescription string
- > customData CustomData
- > transactionGroupSpend TransactionGroupSpend[]
- > transactionGroupEarn TransactionGroupEarn[]
- > transactionGroupReference map<string, object>
- > transactions Transaction[]
- > cancelable boolean
- > editable boolean
- > createUserId integer<int64>
- > createUserName string
- > updateUserId integer<int64>
- > updateUserName string

● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups?includeMergeDetails=boolean' \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "transactionGroupDescription": "",
    "transactionGroupDateTime": "",
    "transactionGroupStatusCode": "",
    "activityStartDateTime": "",
    "activityEndDateTime": "",
    "originationDate": "",
    "totalSpendAmount": 0,
    "totalQualifiedSpendAmount": 0,
    "statusCode": "",
    "postDate": "",
    "typeCode": "",
    "typeDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": "",
      "ace15": "",
      "ace15Description": ""
    },
    "transactionGroupSpend": [
      {
        "spendAmount": 0,
        "currencyTypeCode": "",
        "currencyCategoryCode": "",
        "currencyCategoryDescription": "",
        "isoCurrencyCode": ""
      }
    ]
  }
]
```



```
    }
  ],
  "transactionGroupEarn": [
    {
      "amount": 0,
      "currencyTypeCode": "",
      "currencyCategoryCode": "",
      "currencyCategoryDescription": ""
    }
  ],
  "transactionGroupReference": "",
  "transactions": [
    {
      "transactionId": "",
      "transactionGroupId": "",
      "transactionGroupReferenceId": "",
      "parentTransactionId": "",
      "linkedTransactionId": "",
      "referenceId": "",
      "memberIdentifier": "",
      "accountNumber": "",
      "memberRedemptionId": "1",
      "transactionDateTime": "",
      "partnerName": "",
      "partnerProgramCode": "",
      "description": "",
      "typeCode": "",
      "typeDescription": "",
      "subTypeCode": "",
      "postDate": "",
      "originationDate": "",
      "retro": false,
      "edited": false,
      "sourceCode": "",
      "statusCode": "",
      "updateDateTime": "",
      "earningRuleDescription": "",
      "customData": {
        "ace01": "",
        "ace01Description": "",
        "ace02": "",
        "ace02Description": "",
        "ace03": "",
        "ace03Description": "",
        "ace04": "",
        "ace04Description": "",
        "ace05": "",
        "ace05Description": "",
        "ace06": "",
        "ace06Description": "",
        "ace07": "",
        "ace07Description": "",
        "ace08": "",
        "ace08Description": "",
        "ace09": "",
        "ace09Description": "",
        "ace10": "",
        "ace10Description": "",
        "ace11": "",
      }
    }
  ]
}
```



```
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "spend": {
    "perItem": 0,
    "spendAmount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "isoCurrencyCode": "",
    "currencyConversionRatio": "",
    "quantity": ""
  },
  "earn": {
    "amount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "earningsLimited": false,
    "tierQualifying": false,
    "transactionCounters": [
      {
        "counterTypeCode": "",
        "counterValue": 0
      }
    ]
  },
  "redemption": {
    "redemptionItemCode": "",
    "currencyRequired": 0,
    "minimumQuantity": "",
    "maximumQuantity": "",
    "redeemIncrement": "",
    "faceValue": 0,
    "amountProvided": false,
    "poolName": "PL-16",
    "lineItems": [
      {
        "description": "",
        "itemCode": "",
        "quantity": "",
        "amount": 0
      }
    ]
  },
  "expiredCurrency": {
    "expiredAmount": 0,
    "expiredDate": "",
    "currencyTypeCode": "",
    "currencyCategoryCode": ""
  },
  "hotelTransaction": {
    "checkInDate": "",
    "checkOutDate": ""
  }
}
```




```
"eligibleStays": "",
"eligibleCappedNights": "",
"eligibleNights": "",
"roomNumber": "",
"roomType": "",
"roomRate": 0,
"roomRateInt1": 0,
"totalRevenueAmount": 0,
"totalRevenueAmountInt1": 0,
"loyaltyEligibleAmount": 0,
"loyaltyEligibleAmountInt1": 0,
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountInt1": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueInt1": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueInt1": 0,
"checkNumber": ""
},
"travelTransaction": {
```



14.16.13. Get Member Transaction Group By Transaction Group Id

GET

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/{transactionGroupId}>

Try it ▶

Get member transaction group by transaction group id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-transaction:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionGroupId string *

Responses

▼ ● 200 Success



TransactionGroup

- > transactionGroupId string *
- > transactionGroupReferenceId string *
- > transactionGroupDescription string
- > transactionGroupDateTime string<date-time>
- > transactionGroupStatusCode string
- > activityStartDateTime string<date-time>
- > activityEndDateTime string<date-time>
- > originationDate string<date>
- > totalSpendAmount number
- > totalQualifiedSpendAmount number
- > statusCode string
- > postDate string<date>
- > typeCode string
- > typeDescription string
- > customData CustomData
- > transactionGroupSpend TransactionGroupSpend[]
- > transactionGroupEarn TransactionGroupEarn[]
- > transactionGroupReference map<string, object>
- > transactions Transaction[]
- > cancelable boolean
- > editable boolean
- > createUserId integer<int64>
- > createUserName string
- > updateUserId integer<int64>
- > updateUserName string

400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/{transactionGroupId} \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "transactionGroupId": "",
  "transactionGroupReferenceId": "",
  "transactionGroupDescription": "",
  "transactionGroupDateTime": "",
  "transactionGroupStatusCode": "",
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "originationDate": "",
  "totalSpendAmount": 0,
  "totalQualifiedSpendAmount": 0,
  "statusCode": "",
  "postDate": "",
  "typeCode": "",
  "typeDescription": "",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "transactionGroupSpend": [
    {
      "spendAmount": 0,
      "currencyTypeCode": "",
      "currencyCategoryCode": "",
      "currencyCategoryDescription": "",
      "isoCurrencyCode": ""
    }
  ]
}
```



```
],
"transactionGroupEarn": [
  {
    "amount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "currencyCategoryDescription": ""
  }
],
"transactionGroupReference": "",
"transactions": [
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": ""
    }
  }
]
```



```
"ace12": "",
"ace12Description": "",
"ace13": "",
"ace13Description": "",
"ace14": "",
"ace14Description": "",
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": ""
}
```




```
"eligibleCappedNights": "",
"eligibleNights": "",
"roomNumber": "",
"roomType": "",
"roomRate": 0,
"roomRateIntl": 0,
"totalRevenueAmount": 0,
"totalRevenueAmountIntl": 0,
"loyaltyEligibleAmount": 0,
"loyaltyEligibleAmountIntl": 0,
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountIntl": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueIntl": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueIntl": 0,
"checkNumber": ""
```



14.16.14. Cancel Member Transaction Group By Transaction Group Id

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/{transactionGroupId}`

Try it ▶

Cancel a member's transaction group by transaction group id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-transaction:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionGroupId string *

Body Parameters

▼ body CancelCallLog

Description

Cancellation comment details

> commentSubReason string

> comments string

Responses

▼ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



```
> expiredCurrency TransactionExpiredCurrency
> hotelTransaction HotelTransaction
> travelTransaction TravelTransaction
> transactionReference map<string, object>
> cancelable boolean
> editable boolean
> createUserId integer<int64>
> createUserName string
> updateUserId integer<int64>
> updateUserName string
> memberCommentId integer<int64>
> cancellationSubReason string
> businessEligible boolean
> businessAccountNumber string
> awardNumber string
> baseAccountNumber string
> transactionAdjustment TransactionAdjustment
> transactionBonus TransactionBonus
> transactionPartner TransactionPartner
> transactionPromotion TransactionPromotion
> transactionTransfer TransactionTransfer
> mergeDetails MergeDetail
> transactionRelationships TransactionRelationship[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request DELETE \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/{transactionGroupId} \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```




```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```



14.16.15. Cancel Member Transaction Group By Transaction Group Id (POST)

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/{transactionGroupId}>

Try it ▶

Cancel a member's transaction group by transaction group id using POST for OAuth compatibility

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-transaction:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionGroupId string *

Body Parameters

▼ body CancelCallLog *

Description

Cancellation comment details

> commentSubReason string

> comments string

Responses

▼ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



```
> expiredCurrency TransactionExpiredCurrency
> hotelTransaction HotelTransaction
> travelTransaction TravelTransaction
> transactionReference map<string, object>
> cancelable boolean
> editable boolean
> createUserId integer<int64>
> createUserName string
> updateUserId integer<int64>
> updateUserName string
> memberCommentId integer<int64>
> cancellationSubReason string
> businessEligible boolean
> businessAccountNumber string
> awardNumber string
> baseAccountNumber string
> transactionAdjustment TransactionAdjustment
> transactionBonus TransactionBonus
> transactionPartner TransactionPartner
> transactionPromotion TransactionPromotion
> transactionTransfer TransactionTransfer
> mergeDetails MergeDetail
> transactionRelationships TransactionRelationship[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/{transactionGroupId}/cancel \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```




```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```



14.16.16. Get Member Transaction Group By Transaction Group Reference Group Id

GET

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/transactionGroupRef>

Try it ▶

Get member transaction group by transaction Group reference group id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-transaction:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionGroupReferenceId string *

Responses

▼ ● 200 Success



TransactionGroup

- > transactionGroupIdd string *
- > transactionGroupReferenceId string *
- > transactionGroupDescription string
- > transactionGroupDateTime string<date-time>
- > transactionGroupStatusCode string
- > activityStartDateTime string<date-time>
- > activityEndDateTime string<date-time>
- > originationDate string<date>
- > totalSpendAmount number
- > totalQualifiedSpendAmount number
- > statusCode string
- > postDate string<date>
- > typeCode string
- > typeDescription string
- > customData CustomData
- > transactionGroupSpend TransactionGroupSpend[]
- > transactionGroupEarn TransactionGroupEarn[]
- > transactionGroupReference map<string, object>
- > transactions Transaction[]
- > cancelable boolean
- > editable boolean
- > createUserId integer<int64>
- > createUserName string
- > updateUserId integer<int64>
- > updateUserName string

● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/transactionGroupReferenceId \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "transactionGroupId": "",
  "transactionGroupReferenceId": "",
  "transactionGroupDescription": "",
  "transactionGroupDateTime": "",
  "transactionGroupStatusCode": "",
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "originationDate": "",
  "totalSpendAmount": 0,
  "totalQualifiedSpendAmount": 0,
  "statusCode": "",
  "postDate": "",
  "typeCode": "",
  "typeDescription": "",
  "customData": {
    "ace01": "",
    "ace01Description": "",
    "ace02": "",
    "ace02Description": "",
    "ace03": "",
    "ace03Description": "",
    "ace04": "",
    "ace04Description": "",
    "ace05": "",
    "ace05Description": "",
    "ace06": "",
    "ace06Description": "",
    "ace07": "",
    "ace07Description": "",
    "ace08": "",
    "ace08Description": "",
    "ace09": "",
    "ace09Description": "",
    "ace10": "",
    "ace10Description": "",
    "ace11": "",
    "ace11Description": "",
    "ace12": "",
    "ace12Description": "",
    "ace13": "",
    "ace13Description": "",
    "ace14": "",
    "ace14Description": "",
    "ace15": "",
    "ace15Description": ""
  },
  "transactionGroupSpend": [
    {
      "spendAmount": 0,
      "currencyTypeCode": "",
      "currencyCategoryCode": "",
      "currencyCategoryDescription": "",
      "isoCurrencyCode": ""
    }
  ]
}
```



```
],
"transactionGroupEarn": [
  {
    "amount": 0,
    "currencyTypeCode": "",
    "currencyCategoryCode": "",
    "currencyCategoryDescription": ""
  }
],
"transactionGroupReference": "",
"transactions": [
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": ""
    }
  }
]
```



```
"ace12": "",
"ace12Description": "",
"ace13": "",
"ace13Description": "",
"ace14": "",
"ace14Description": "",
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": ""
}
```




```
"eligibleCappedNights": "",
"eligibleNights": "",
"roomNumber": "",
"roomType": "",
"roomRate": 0,
"roomRateIntl": 0,
"totalRevenueAmount": 0,
"totalRevenueAmountIntl": 0,
"loyaltyEligibleAmount": 0,
"loyaltyEligibleAmountIntl": 0,
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountIntl": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueIntl": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueIntl": 0,
"checkNumber": ""
```



14.16.17. Cancel Member Transaction Group By Transaction Group Reference Id

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/transactionGroupRef`

Try it ▶

Cancel a member's transaction group by transaction group reference id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-transaction:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionGroupReferenceId string *

Body Parameters

▼ body CancelCallLog

Description

Cancellation comment details

> commentSubReason string

> comments string

Responses

▼ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



```
> expiredCurrency TransactionExpiredCurrency
> hotelTransaction HotelTransaction
> travelTransaction TravelTransaction
> transactionReference map<string, object>
> cancelable boolean
> editable boolean
> createUserId integer<int64>
> createUserName string
> updateUserId integer<int64>
> updateUserName string
> memberCommentId integer<int64>
> cancellationSubReason string
> businessEligible boolean
> businessAccountNumber string
> awardNumber string
> baseAccountNumber string
> transactionAdjustment TransactionAdjustment
> transactionBonus TransactionBonus
> transactionPartner TransactionPartner
> transactionPromotion TransactionPromotion
> transactionTransfer TransactionTransfer
> mergeDetails MergeDetail
> transactionRelationships TransactionRelationship[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request DELETE \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/transactionGroupReferenceId \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```




```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```



14.16.18. Cancel Member Transaction Group By Transaction Group Reference Id (POST)

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/transactionGroupRef>

Try it ▶

Cancel a member's transaction group by transaction group reference id using POST for OAuth compatibility

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-transaction:cancel

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> transactionGroupReferenceId string *

Body Parameters

▼ body CancelCallLog *

Description

Cancellation comment details

> commentSubReason string

> comments string

Responses

▼ ● 200 Success



- Transaction[]
- > transactionId string
 - > transactionGroupId string
 - > transactionGroupReferenceId string
 - > parentTransactionId string
 - > linkedTransactionId string
 - > referenceId string
 - > memberIdentifier string
 - > accountNumber string
 - > memberRedemptionId string
 - > transactionDateTime string<date-time>
 - > partnerName string
 - > partnerProgramCode string
 - > description string
 - > typeCode string
 - > typeDescription string
 - > subTypeCode string
 - > postDate string<date>
 - > originationDate string<date>
 - > retro boolean
 - > edited boolean
 - > sourceCode string
 - > statusCode string
 - > updateDateTime string<date-time>
 - > earningRuleDescription string
 - > customData CustomData
 - > spend TransactionSpend
 - > earn TransactionEarn
 - > redemption TransactionRedemption



- > expiredCurrency TransactionExpiredCurrency
- > hotelTransaction HotelTransaction
- > travelTransaction TravelTransaction
- > transactionReference map<string, object>
- > cancelable boolean
- > editable boolean
- > createUserId integer<int64>
- > createUserName string
- > updateUserId integer<int64>
- > updateUserName string
- > memberCommentId integer<int64>
- > cancellationSubReason string
- > businessEligible boolean
- > businessAccountNumber string
- > awardNumber string
- > baseAccountNumber string
- > transactionAdjustment TransactionAdjustment
- > transactionBonus TransactionBonus
- > transactionPartner TransactionPartner
- > transactionPromotion TransactionPromotion
- > transactionTransfer TransactionTransfer
- > mergeDetails MergeDetail
- > transactionRelationships TransactionRelationship[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transaction-groups/transactionGroupReferenceId \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "transactionId": "",
    "transactionGroupId": "",
    "transactionGroupReferenceId": "",
    "parentTransactionId": "",
    "linkedTransactionId": "",
    "referenceId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "memberRedemptionId": "1",
    "transactionDateTime": "",
    "partnerName": "",
    "partnerProgramCode": "",
    "description": "",
    "typeCode": "",
    "typeDescription": "",
    "subTypeCode": "",
    "postDate": "",
    "originationDate": "",
    "retro": false,
    "edited": false,
    "sourceCode": "",
    "statusCode": "",
    "updateDateTime": "",
    "earningRuleDescription": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": ""
    }
  }
]
```



```
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": "",
  "eligibleNights": "",
  "roomNumber": "",
  "roomType": "",
  "roomRate": 0,
  "roomRateIntl": 0,
```




```
"totalRevenueAmount": 0,  
"totalRevenueAmountInt1": 0,  
"loyaltyEligibleAmount": 0,  
"loyaltyEligibleAmountInt1": 0,  
"loyaltyIneligibleAmount": 0,  
"loyaltyIneligibleAmountInt1": 0,  
"isoCurrencyCode": "",  
"bookingSource": "",  
"paymentCode": "",  
"purposeOfVisit": "",  
"travelAgentName": "",  
"groupName": "",  
"companyName": "",  
"reservationConfirmationNumber": "",  
"noShow": false,  
"pmsConfirmationNumber": "",  
"overlappingStayPoolId": "",  
"nonStayIndicator": false,
```



14.16.19. Get Member Transaction Aggregates

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/aggregates`

Try it ▶

Search for member's transactions aggregates

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-transaction:read

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** string *

Query Parameters

> **fromDate** string<date>

mm/dd/yyyy



> **toDate** string<date>

mm/dd/yyyy



> **type** string

Select...



Responses

▼ ● 200 Success

▼ TransactionAggregate

> **baseEarnings** TransactionCurrency[]

> **bonusEarnings** TransactionCurrency[]

> **partnerEarnings** PartnerEarn[]

> **promotionEarnings** TransactionCurrency[]

> **transactionAmounts** TransactionCurrency[]

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error





14.16.20. Retrieve member transaction data, aggregated and grouped by year.

GET

<https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/aggregates/yearly>

Try it ▶

Retrieve member transaction data, aggregated and grouped by year.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-transaction:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> fromYear integer<int32>

Format: int32

> toYear integer<int32>

Format: int32

Responses

▼ ● 200 Success

▼ TransactionAggregateYearGroup[]

Description

Summation of the amounts deposited to the member's account. Grouped and sorted descending by year.

- > year integer<int32>
- > baseEarnings TransactionAggregateCounter
- > bonusEarnings TransactionAggregateCounter
- > partnerEarnings TransactionAggregateCounter
- > promotionEarnings TransactionAggregateCounter
- > tierRollover TransactionAggregateCounter

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized



- **403** Forbidden
- **500** Internal error

Request



```
curl --request GET \  
--url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/transactions/aggregates/yearly \  
--header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "year": 2025,
    "baseEarnings": {
      "points": [
        {
          "pointCategory": "Points",
          "amount": 1000
        }
      ],
      "counters": [
        {
          "counterTypeCode": "",
          "counterValue": 0
        }
      ]
    }
  }
],
}
```



14.16.21. Post Transaction for Earnings Processing

POST`https://api.b.preview.loyaltyservices.io/api/v3/transactions`

Try it ▶

This will submit a transaction through the earnings process. If a member identifier is associated to the transaction, it will go through the entire process. If not, it will be staged for future processing.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:transaction:submit`

Authorize

Use Your Own Token



Body Parameters

▼

body

TransactionRequest

*

Description

Transaction Request

>

referenceId

string

>

member

TransactionMember

>

transactionDateTime

string<date-time>

mm/dd/yyyy, --:-- --📅

>

description

string

>

originationDate

string<date>

mm/dd/yyyy📅

>

awardNumber

string

>

providedAmount

number

>

sourceCode

string

>

migrateStatusCode

string

>

customData

CustomData

>

spend

TransactionSpend

>

hotelTransaction

HotelTransaction

>

retailTransaction

RetailTransaction

>

travelTransaction

TravelTransaction

>

transactionPartner

TransactionPartnerEarn

>

transactionMemberPaymentCardAccount

TransactionMemberPaymentCardAccount

>

transactionReference

map<string, object>

>

useCouponCodes

string[]

+

Responses

- 200

Success
- ▼

●

400

Bad request

▼

ErrorResponse

>

errors

ValidationError[]
- 401

Unauthorized
- 403

Forbidden
- 500

Internal error



Request

 Curl

```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/transactions \  
  --header 'accept: application/json'
```

Responses

 200

```
// Success
```



14.16.22. Get Available Manual Bonus Rules

GET`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/bonuses`**Try it** ▶

Return a collection of all currently active, displayable online, and manually processed bonuses, filtered by type for which the member is eligible.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>Flow: **Client Credentials**

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-bonus:read

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** string *

Query Parameters

> **typeCode** string

Responses

▼ ● **200** Success



▼ BonusRule[]

- > bonusRuleCode string
- > bonusRuleDescription string
- > bonusRuleTypeDescription string
- > rewardCurrencyCategoryDescription string
- > rewardCurrencyTypeDescription string
- > rewardCurrencyAmount number
- > bonusPointInstanceLimit number
- > billable boolean
- > billableToTypeCode integer<int64>
- > startDate string<date>
- > endDate string<date>
- > graceLimit integer<int32>
- > rewards RuleReward[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/bonuses \  
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "bonusRuleCode": "Adjustment1000",
    "bonusRuleDescription": "Adjustment Bonus 1000 Points",
    "bonusRuleTypeDescription": "Enrollment Bonus",
    "rewardCurrencyCategoryDescription": "Loyalty",
    "rewardCurrencyTypeDescription": "Provided",
    "rewardCurrencyAmount": 500,
    "bonusPointInstanceLimit": 200,
    "billable": true,
    "billableToTypeCode": 23200,
    "startDate": "",
    "endDate": "",
    "graceLimit": "",
    "rewards": [
      {
        "rewardType": "COUNTER",
        "rewardResultType": "Provided",
        "description": "Tier Qualifying Nights",
        "code": "NIGHTS",
        "amount": 10
      }
    ]
  }
]
```



14.16.23. Post Member Bonus

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/bonuses`

Try it

Submit a bonus to a member account.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: **Client Credentials**

Client ID

Client Secret

Scopes **1** ▾Required Scopes: `api:member-bonus:add`

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** `string` *

Query Parameters

> **async** `boolean`☐

Body Parameters

▾ **body** `BonusRequest` *> **bonusCode** `string`> **amount** `number`**counterAmounts** `CountersMap`> **referenceId** `string`> **comment** `string`> **quantity** `integer<int32>`

Format: int32

> **customData** `CustomData`

Responses

▾ ● **200** Success



▼ MemberBonus

- > bonus Bonus
- > transactions Transaction[]
- > coupons Coupon[]
- > memberTier MemberTier
- > memberList MemberList

▼ ● 202 Accepted

▼ AsyncTask

- > asyncTaskId integer<int64>
- > taskTypeCode string
- > taskCode string
- > requestDate string<date-time>
- > requestData object
- > statusCode string
- > responseDate string<date-time>
- > responseData object
- > fileName string
- > lineNumber integer
- > fileData string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/bonuses?async=boolean' \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "bonus": {
    "bonusCode": "",
    "bonusDescription": ""
  },
  "transactions": [
    {
      "transactionId": "",
      "transactionGroupId": "",
      "transactionGroupReferenceId": "",
      "parentTransactionId": "",
      "linkedTransactionId": "",
      "referenceId": "",
      "memberIdentifier": "",
      "accountNumber": "",
      "memberRedemptionId": "1",
      "transactionDateTime": "",
      "partnerName": "",
      "partnerProgramCode": "",
      "description": "",
      "typeCode": "",
      "typeDescription": "",
      "subTypeCode": "",
      "postDate": "",
      "originationDate": "",
      "retro": false,
      "edited": false,
      "sourceCode": "",
      "statusCode": "",
      "updateDateTime": "",
      "earningRuleDescription": "",
      "customData": {
        "ace01": "",
        "ace01Description": "",
        "ace02": "",
        "ace02Description": "",
        "ace03": "",
        "ace03Description": "",
        "ace04": "",
        "ace04Description": "",
        "ace05": "",
        "ace05Description": "",
        "ace06": "",
        "ace06Description": "",
        "ace07": "",
        "ace07Description": "",
        "ace08": "",
        "ace08Description": "",
        "ace09": "",
        "ace09Description": "",
        "ace10": "",
        "ace10Description": "",
        "ace11": "",
        "ace11Description": "",
        "ace12": "",

```



```
"ace12Description": "",
"ace13": "",
"ace13Description": "",
"ace14": "",
"ace14Description": "",
"ace15": "",
"ace15Description": ""
},
"spend": {
  "perItem": 0,
  "spendAmount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "isoCurrencyCode": "",
  "currencyConversionRatio": "",
  "quantity": ""
},
"earn": {
  "amount": 0,
  "currencyTypeCode": "",
  "currencyCategoryCode": "",
  "earningsLimited": false,
  "tierQualifying": false,
  "transactionCounters": [
    {
      "counterTypeCode": "",
      "counterValue": 0
    }
  ]
},
"redemption": {
  "redemptionItemCode": "",
  "currencyRequired": 0,
  "minimumQuantity": "",
  "maximumQuantity": "",
  "redeemIncrement": "",
  "faceValue": 0,
  "amountProvided": false,
  "poolName": "PL-16",
  "lineItems": [
    {
      "description": "",
      "itemCode": "",
      "quantity": "",
      "amount": 0
    }
  ]
},
"expiredCurrency": {
  "expiredAmount": 0,
  "expiredDate": "",
  "currencyTypeCode": "",
  "currencyCategoryCode": ""
},
"hotelTransaction": {
  "checkInDate": "",
  "checkOutDate": "",
  "eligibleStays": "",
  "eligibleCappedNights": ""
}
```



```
"eligibleNights": "",
"roomNumber": "",
"roomType": "",
"roomRate": 0,
"roomRateIntl": 0,
"totalRevenueAmount": 0,
"totalRevenueAmountIntl": 0,
"loyaltyEligibleAmount": 0,
"loyaltyEligibleAmountIntl": 0,
"loyaltyIneligibleAmount": 0,
"loyaltyIneligibleAmountIntl": 0,
"isoCurrencyCode": "",
"bookingSource": "",
"paymentCode": "",
"purposeOfVisit": "",
"travelAgentName": "",
"groupName": "",
"companyName": "",
"reservationConfirmationNumber": "",
"noShow": false,
"pmsConfirmationNumber": "",
"overlappingStayPoolId": "",
"nonStayIndicator": false,
"nonStayLocationId": "",
"nonStayEligibleRevenue": 0,
"nonStayEligibleRevenueIntl": 0,
"nonStayIneligibleRevenue": 0,
"nonStayIneligibleRevenueIntl": 0,
"checkNumber": ""
},
"travelTransaction": {
  "activityStartDateTime": "",
  "activityEndDateTime": "",
  "totalSpendAmount": 0
},
"transactionReference": "",
"cancelable": false,
"editable": false,
"createUserId": "",
"createUserName": "",
"updateUserId": "",
"updateUserName": "",
"memberCommentId": "",
"cancellationSubReason": "",
"businessEligible": false,
"businessAccountNumber": "",
"awardNumber": "",
"baseAccountNumber": "",
"transactionAdjustment": {
  "comments": "",
  "referenceId": "",
  "reasonCode": ""
},
"transactionBonus": {
  "bonusCode": "",
  "bonusType": ""
},
"transactionPartner": {
  "partnerName": "",
```



```
    "partnerCurrencyCategoryCode": ""
  },
  "transactionPromotion": {
    "promotionCode": ""
  },
  "transactionTransfer": {
    "fromMemberAccountNumber": "A234567890",
    "fromMemberName": "Marcel Rufus",
    "toMemberAccountNumber": "B987654321",
    "toMemberName": "Janel Liesje",
    "transferredBy": "A0123456789"
  },
  "mergeDetails": {
    "fromMemberAccountNumber": "A234567890"
  },
  "transactionRelationships": [
    {
      "type": "B2B_SUCCESSOR",
      "key": "EB-1234",
      "accountNumber": ""
    }
  ]
},
"coupons": [
  {
    "couponId": 147258369,
    "couponRequestId": 147258370,
    "offer": {
      "offerId": 147258369,
      "offerTypeCode": "Discount",
      "code": "ALON2020",
      "name": "Free night",
      "description": "Award double points",
      "featureVersion": 2,
      "statusCode": "Enabled",
      "startDateTime": "",
      "endDateTime": "",
      "offerAttributes": [
        {
          "offerAttributeId": 147258369,
          "attribute": {
            "attributeId": 147258369,
            "typeCode": "Validation",
            "statusCode": "Enabled",
            "name": "AccountNumber",
            "description": "Account Number",
            "dataTypeCode": "open"
          },
          "defaultValue": "Standard",
          "requireOnUse": true,
          "requireOnHold": true,
          "includeOnResponse": false,
          "attributeTypeCode": "Validation",
          "operatorCode": "EQ",
          "operand": "GI",
          "smartGroupId": 147258369
        }
      ]
    }
  ]
}
```



```
},
"code": "ZPVWTAROW20Q",
"statusCode": "Activated",
"startDate": "",
"endDate": "",
"useTypeCode": "Single",
"useLimit": 1,
"currentUse": 1,
"couponUses": [
  {
    "useDateTime": "",
    "statusCode": "Used",
    "channelCode": "System",
    "alias": "",
    "useCount": 1,
    "couponUseAttributes": [
      {
        "couponUseAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
```



14.17. ping

14.17.1. Ping Service

GET

<https://api.b.preview.loyaltyservices.io/api/v3/ping>

Try it ▶

Validate Service Connectivity

Responses

- ▼

● 200

Success

string
- ▼

● 400

Bad request

▼

ErrorResponse

> errors ValidationError[]
- 401

Unauthorized
- 403

Forbidden
- 500

Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/ping \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
""
```



14.18. promotion

14.18.1. Get Promotions

GET

<https://api.b.preview.loyaltyservices.io/api/v3/promotions>

Try it ▶

Retrieve promotions

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:promotion:read

Authorize

Use Your Own Token

Query Parameters

> memberIdentifier string

> code string

> typeCode string

> includeUnavailable boolean

> includeInactive boolean

Responses

▼ ● 200 Success



```
▼ Promotion[]
  > promotionId integer<int64>
  > promotionCode string
  > promotionName string
  > promotionDescription string
  > promotionTypeCode string
  > promotionSubTypeCode string
  > registerStartDateTime string<date-time>
  > registerEndDateTime string<date-time>
  > registrationChannelCodes string[]
  > autoRegister boolean
  > startDate string<date>
  > endDate string<date>
  > accrualTypeCode string
  > callCenterOnly boolean
  > displayOnline boolean
  > displayOnlineUntilDateTime string<date-time>
  > customData SmartGroupCustomData
  > genericItem GenericItem
  > graceLimit integer<int32>
  > hasMinUserRoleToRegister boolean
  > minimumAccrualAmount number
  > occurrenceLimit integer<int64>
  > partnerProgramCode string
  > currencyCategoryCode string
  > processRetroAccruals boolean
  > restrict boolean
  > memberListId integer<int64>
  > slidingQualificationPeriod integer<int64>
```




```
> tierCode string
> trackRegistrationSource boolean
> preQualificationCounters PromotionCounter[]
> counters PromotionCounter[]
> resultCommunication boolean
> resultCommunicationId integer<int64>
> resultCurrency boolean
> resultCurrencyAmount number
> resultCurrencyCategoryCode string
> resultCurrencyTierQualifier boolean
> resultExternal boolean
> resultExternalCode string
> resultPromotion boolean
> resultPromotionId integer<int64>
> resultTier boolean
> resultTierCode string
> resultTierExpireDateTime string<date-time>
> rewardThresholds boolean
> results PromotionResult[]
> statusCode string
> renewableCounters boolean
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/promotions?includeUnavailable=boolean&includeInactive=boolean' \  
  --header 'accept: application/json'
```



Responses

200



```
// Success
[
  {
    "promotionId": "",
    "promotionCode": "",
    "promotionName": "",
    "promotionDescription": "",
    "promotionTypeCode": "",
    "promotionSubTypeCode": "",
    "registerStartDateTime": "",
    "registerEndDateTime": "",
    "registrationChannelCodes": [
      ""
    ],
    "autoRegister": false,
    "startDate": "",
    "endDate": "",
    "accrualTypeCode": "",
    "callCenterOnly": false,
    "displayOnline": false,
    "displayOnlineUntilDateTime": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": "",
      "ace09Description": "",
      "ace10": "",
      "ace10Description": "",
      "ace11": "",
      "ace11Description": "",
      "ace12": "",
      "ace12Description": "",
      "ace13": "",
      "ace13Description": "",
      "ace14": "",
      "ace14Description": "",
      "ace15": "",
      "ace15Description": "",
      "ace01SmartGroupId": "",
      "ace02SmartGroupId": "",
      "ace03SmartGroupId": "",
      "ace04SmartGroupId": ""
    }
  }
]
```



```
"ace05SmartGroupId": "",
"ace06SmartGroupId": "",
"ace07SmartGroupId": "",
"ace08SmartGroupId": "",
"ace09SmartGroupId": "",
"ace10SmartGroupId": "",
"ace11SmartGroupId": "",
"ace12SmartGroupId": "",
"ace13SmartGroupId": "",
"ace14SmartGroupId": "",
"ace15SmartGroupId": ""
},
"genericItem": {
  "contentCode": "Bonus100",
  "contentTypeCode": "HEADER",
  "name": "Bonus 100 points",
  "shortDescription": "Award 100 points",
  "description": "Award 100 points on 100th stay",
  "alternateDescription": "Award 100 points on 100th stay",
  "termsAndConditions": "Terms and conditions for this promotion",
  "title": "Award 100 points",
  "label": "100 points awarded",
  "logoKey": "100-points.png",
  "displayTypeKey": "100 points",
  "startDate": "",
  "endDate": "",
  "url": "/loyaltyprogram/bonus/100-points",
  "tierCode": "B",
  "displayForAll": true,
  "descriptorCode": "LIMITED_TIME"
},
"graceLimit": "",
"hasMinUserRoleToRegister": false,
"minimumAccrualAmount": 0,
"occurrenceLimit": "",
"partnerProgramCode": "",
"currencyCategoryCode": "",
"processRetroAccruals": false,
"restrict": false,
"memberListId": "",
"slidingQualificationPeriod": "",
"tierCode": "",
"trackRegistrationSource": false,
"preQualificationCounters": [
  {
    "counterTypeCode": "",
    "target": 0,
    "result": [
      {
        "type": "",
```



14.18.2. Generate one click promotion token

GET

`https://api.b.preview.loyaltyservices.io/api/v3/promotions/tokens`

Try it ▶

Generate one click promotion token using `promotionCode` and `memberIdentifier`

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:promotion-token:manage`

Authorize

Use Your Own Token

Query Parameters

> `memberIdentifier` string *> `promotionCode` string *

Responses

✓ ● 200 Success

GeneratePromotionTokenResponse

> `token` string

✓ ● 400 Bad request

ErrorResponse

> `errors` ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/promotions/tokens?memberIdentifier=123456789&promotionCode=PROM0100' \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
{
  "token": ""
}
```





14.18.3. Search ProBonuses

GET

`https://api.b.preview.loyaltyservices.io/api/v3/probonuses`

Try it ▶

Search for promotions and bonus rules by code

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:promotion:read

Authorize

Use Your Own Token

Query Parameters

> **code** string> **type** string

Responses

▼ ● 200 Success

▼ ProBonusSearchResult

Description

Result of a ProBonus search, containing matching promotions and bonus rules

> **count** integer> **promotions** Promotion[]> **bonusRules** BonusRule[]

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/probonuses \
  --header 'accept: application/json'
```




Responses

● 200



```
// Success
{
  "count": 0,
  "promotions": [
    {
      "promotionId": "",
      "promotionCode": "",
      "promotionName": "",
      "promotionDescription": "",
      "promotionTypeCode": "",
      "promotionSubTypeCode": "",
      "registerStartDateTime": "",
      "registerEndDateTime": "",
      "registrationChannelCodes": [
        ""
      ],
      "autoRegister": false,
      "startDate": "",
      "endDate": "",
      "accrualTypeCode": "",
      "callCenterOnly": false,
      "displayOnline": false,
      "displayOnlineUntilDateTime": "",
      "customData": {
        "ace01": "",
        "ace01Description": "",
        "ace02": "",
        "ace02Description": "",
        "ace03": "",
        "ace03Description": "",
        "ace04": "",
        "ace04Description": "",
        "ace05": "",
        "ace05Description": "",
        "ace06": "",
        "ace06Description": "",
        "ace07": "",
        "ace07Description": "",
        "ace08": "",
        "ace08Description": "",
        "ace09": "",
        "ace09Description": "",
        "ace10": "",
        "ace10Description": "",
        "ace11": "",
        "ace11Description": "",
        "ace12": "",
        "ace12Description": "",
        "ace13": "",
        "ace13Description": "",
        "ace14": "",
        "ace14Description": "",
        "ace15": "",
        "ace15Description": "",
        "ace01SmartGroupId": "",
        "ace02SmartGroupId": ""
      }
    }
  ]
}
```



```
"ace03SmartGroupId": "",
"ace04SmartGroupId": "",
"ace05SmartGroupId": "",
"ace06SmartGroupId": "",
"ace07SmartGroupId": "",
"ace08SmartGroupId": "",
"ace09SmartGroupId": "",
"ace10SmartGroupId": "",
"ace11SmartGroupId": "",
"ace12SmartGroupId": "",
"ace13SmartGroupId": "",
"ace14SmartGroupId": "",
"ace15SmartGroupId": ""
},
"genericItem": {
  "contentCode": "Bonus100",
  "contentTypeCode": "HEADER",
  "name": "Bonus 100 points",
  "shortDescription": "Award 100 points",
  "description": "Award 100 points on 100th stay",
  "alternateDescription": "Award 100 points on 100th stay",
  "termsAndConditions": "Terms and conditions for this promotion",
  "title": "Award 100 points",
  "label": "100 points awarded",
  "logoKey": "100-points.png",
  "displayTypeKey": "100 points",
  "startDate": "",
  "endDate": "",
  "url": "/loyaltyprogram/bonus/100-points",
  "tierCode": "B",
  "displayForAll": true,
  "descriptorCode": "LIMITED_TIME"
},
"graceLimit": "",
"hasMinUserRoleToRegister": false,
"minimumAccrualAmount": 0,
"occurrenceLimit": "",
"partnerProgramCode": "",
"currencyCategoryCode": "",
"processRetroAccruals": false,
"restrict": false,
"memberListId": "",
"slidingQualificationPeriod": "",
"tierCode": "",
"trackRegistrationSource": false,
"preQualificationCounters": [
  {
    "counterTypeCode": "",
    "target": 0,
    "result": [
      {
        "type": "",
        "value": "",
        "valueType": "",
        "description": "",
        "resultAwardId": "",
        "resultAwardCode": "",
        "bonusByActivity": false,
        "displayName": ""
      }
    ]
  }
]
```



```
    }
  ]
}
],
"counters": [
  {
    "counterTypeCode": "",
    "target": 0,
    "result": [
      {
        "type": "",
        "value": "",
        "valueType": "",
        "description": "",
        "resultAwardId": "",
        "resultAwardCode": "",
        "bonusByActivity": false,
        "displayName": ""
      }
    ]
  }
]
```





14.19. tier

14.19.1. Get Tiers Information

GET

https://api.b.preview.loyaltyservices.io/api/v3/tiers

Try it ▶

Retrieve available tiers

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾ Required Scopes: api:tier:read

Authorize

Use Your Own Token

Query Parameters

> includeHiddenTiers boolean



Responses

✓ • 200 Success

▾ Tier[]

- > code string
- > description string
- > rank integer<int32>
- > displayOrder integer<int32>
- > type string
- > stepDownTier string
- > active boolean
- > display boolean
- > lifeTime boolean

✓ • 400 Bad request

▾ ErrorResponse

- > errors ValidationError[]



- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

 Curl

```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/tiers?includeHiddenTiers=boolean' \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "code": "",  
    "description": "",  
    "rank": "",  
    "displayOrder": "",  
    "type": "Earned, Trial, Exclusive",  
    "stepDownTier": "",  
    "active": false,  
    "display": false,  
    "lifeTime": false  
  }  
]
```

14.20.1. Get Redemption Items

Try it 

Retrieve redemption items. Deprecated in favor of GET /v3/rewards/redemption-rules

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Use Your Own Token

```
> ace04Keys string[]
```




> ace05Keys string[]	+
> ace06Keys string[]	+
> ace07Keys string[]	+
> ace08Keys string[]	+
> ace09Keys string[]	+
> ace10Keys string[]	+
> ace11Keys string[]	+
> ace12Keys string[]	+
> ace13Keys string[]	+
> ace14Keys string[]	+
> ace15Keys string[]	+

Responses

▼ ● 200 Success



▼ RedemptionItem[]

- > redemptionItemId integer<int64>
- > redemptionItemCode string
- > typeCode string
- > description string
- > currencyRequired number
- > minimumQuantity integer<int32>
- > maximumQuantity integer<int32>
- > redeemIncrement integer<int32>
- > faceValue number
- > convertedFaceValue number
- > convertedFaceValueIsoCurrencyCode string
- > amountProvided boolean
- > companyCode string
- > redeemText string
- > redeemable boolean
- > poolRedeemable boolean nullable
- > requireAddress boolean
- > requirePartnerAccountNumber boolean
- > fulfillmentPathCode string
- > sortValue string
- > customData SmartGroupCustomData
- > genericItem GenericItem
- > bonuses RedemptionItemBonus[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/redemption-items \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "redemptionItemId": "",
    "redemptionItemCode": "",
    "typeCode": "",
    "description": "",
    "currencyRequired": 0,
    "minimumQuantity": "",
    "maximumQuantity": "",
    "redeemIncrement": "",
    "faceValue": 0,
    "convertedFaceValue": 0,
    "convertedFaceValueIsoCurrencyCode": "",
    "amountProvided": false,
    "companyCode": "",
    "redeemText": "",
    "redeemable": false,
    "poolRedeemable": false,
    "requireAddress": false,
    "requirePartnerAccountNumber": false,
    "fulfillmentPathCode": "",
    "sortValue": "",
    "customData": {
      "ace01": "",
      "ace01Description": "",
      "ace02": "",
      "ace02Description": "",
      "ace03": "",
      "ace03Description": "",
      "ace04": "",
      "ace04Description": "",
      "ace05": "",
      "ace05Description": "",
      "ace06": "",
      "ace06Description": "",
      "ace07": "",
      "ace07Description": "",
      "ace08": "",
      "ace08Description": "",
      "ace09": ""
    }
  }
]
```



14.20.2. Find Redemption Rules

GET

`https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules`

Try it ▶

Search for redemption rules.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:redemption:read

Authorize

Use Your Own Token

Query Parameters

> externalCode string

> pageNumber integer

> pageSize integer

1

Responses

✓ ● 200 Success

FindRedemptionRulesResponse

> redemptionRules RedemptionRule[]

> pageInfo PageInfo

✓ ● 400 Bad request

ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules?pageNumber=1' \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "redemptionRules": [
    {
      "id": "",
      "code": "GOLD_REWARD_2026",
      "externalCode": "GOLD_REWARD",
      "description": "Gold Member Exclusive Reward",
      "statusCode": "",
      "startDate": "2025-01-01",
      "endDate": "2050-12-31",
      "redemptionTypeCode": "I = Item, T = Travel",
      "rank": 10,
      "redeemable": true,
      "workflowStatus": "",
      "rawExpression": "",
      "faceValue": 500,
      "dataRelease": "",
      "partnerCode": "",
      "minimumRoleRequired": "",
      "quantities": {
        "pointCategoryCode": "POINTS",
        "amountProvided": false,
        "pointsRequired": 5000,
        "providedAmountRate": 100,
        "minimumQuantity": 1,
        "maximumQuantity": "",
        "redeemIncrement": 1
      },
      "fulfillment": {
        "fulfillmentPathCode": "COUPON",
        "communicationTypeCode": "RedemptionConfirmationEmail",
        "couponAward": "",
        "redemptionRuleBonusCode": "",
        "merchant": ""
      },
      "content": ""
    }
  ],
  "pageInfo": {
    "number": 0,
    "size": 0,
    "totalItems": 0,
    "hasNext": false
  }
}
```



14.20.3. Create Redemption Rule

POST`https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules`

Try it ▶

Create a new redemption rule

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:redemption:modify

Authorize

Use Your Own Token



Body Parameters

▼

body RedemptionRule *

> code string *

> externalCode string

> description string *

> statusCode string

A

> startDate string<date> *

mm/dd/yyyy

📅

> endDate string<date>

mm/dd/yyyy

📅

> redemptionTypeCode string *

> rank integer<int32>

1

> redeemable boolean

☐

> workflowStatus string

WORKING

> rawExpression string

> faceValue number

> dataRelease integer<int64>

Format: int64

> partnerCode string

> minimumRoleRequired integer<int64>

Format: int64

> quantities RedemptionRuleQuantities *

> fulfillment RedemptionRuleFulfillment

> content string

Responses

▼

● 200 Success



▼ RedemptionRule

- > id integer<int64>
- > code string *
- > externalCode string
- > description string *
- > statusCode string
- > startDate string<date> *
- > endDate string<date>
- > redemptionTypeCode string *
- > rank integer<int32>
- > redeemable boolean
- > workflowStatus string
- > rawExpression string
- > faceValue number
- > dataRelease integer<int64>
- > partnerCode string
- > minimumRoleRequired integer<int64>
- > quantities RedemptionRuleQuantities *
- > fulfillment RedemptionRuleFulfillment
- > content string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "code": "GOLD_REWARD_2026",
    "description": "Gold Member Exclusive Reward",
    "statusCode": "A",
    "startDate": "2025-01-01",
    "redemptionTypeCode": "I = Item, T = Travel",
    "rank": 1,
    "redeemable": true,
    "workflowStatus": "WORKING",
    "quantities": {
      "pointCategoryCode": "POINTS",
      "amountProvided": false,
      "minimumQuantity": 1
    },
    "fulfillment": {
      "fulfillmentPathCode": "COUPON"
    }
  }'
```



Responses

● 200

```
// Success
{
  "id": "",
  "code": "GOLD_REWARD_2026",
  "externalCode": "GOLD_REWARD",
  "description": "Gold Member Exclusive Reward",
  "statusCode": "",
  "startDate": "2025-01-01",
  "endDate": "2050-12-31",
  "redemptionTypeCode": "I = Item, T = Travel",
  "rank": 10,
  "redeemable": true,
  "workflowStatus": "",
  "rawExpression": "",
  "faceValue": 500,
  "dataRelease": "",
  "partnerCode": "",
  "minimumRoleRequired": "",
  "quantities": {
    "pointCategoryCode": "POINTS",
    "amountProvided": false,
    "pointsRequired": 5000,
    "providedAmountRate": 100,
    "minimumQuantity": 1,
    "maximumQuantity": "",
    "redeemIncrement": 1
  },
  "fulfillment": {
    "fulfillmentPathCode": "COUPON",
    "communicationTypeCode": "RedemptionConfirmationEmail",
    "couponAward": "",
    "redemptionRuleBonusCode": "",
    "merchant": ""
  },
  "content": ""
}
```



14.20.4. Replace Redemption Rule

PUT`https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules/{code}`

Try it ▶

Fully replaces an unreleased redemption rule. All fields are overwritten with the values provided. Only rules in WORKING or READY workflow status may be replaced.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:redemption:modify

Authorize

Use Your Own Token

Path Parameters

`code` string *



Body Parameters

▼

body RedemptionRule *

> code string *

> externalCode string

> description string *

> statusCode string

A

> startDate string<date> *

mm/dd/yyyy

> endDate string<date>

mm/dd/yyyy

> redemptionTypeCode string *

> rank integer<int32>

1

> redeemable boolean

☐

> workflowStatus string

WORKING

> rawExpression string

> faceValue number

> dataRelease integer<int64>

Format: int64

> partnerCode string

> minimumRoleRequired integer<int64>

Format: int64

> quantities RedemptionRuleQuantities *

> fulfillment RedemptionRuleFulfillment

> content string

Responses

▼

●

200 Success



▼ RedemptionRule

- > id integer<int64>
- > code string *
- > externalCode string
- > description string *
- > statusCode string
- > startDate string<date> *
- > endDate string<date>
- > redemptionTypeCode string *
- > rank integer<int32>
- > redeemable boolean
- > workflowStatus string
- > rawExpression string
- > faceValue number
- > dataRelease integer<int64>
- > partnerCode string
- > minimumRoleRequired integer<int64>
- > quantities RedemptionRuleQuantities *
- > fulfillment RedemptionRuleFulfillment
- > content string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 404 Redemption Rule not found
- 500 Internal error



Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules/{code} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "code": "GOLD_REWARD_2026",
    "description": "Gold Member Exclusive Reward",
    "statusCode": "A",
    "startDate": "2025-01-01",
    "redemptionTypeCode": "I = Item, T = Travel",
    "rank": 1,
    "redeemable": true,
    "workflowStatus": "WORKING",
    "quantities": {
      "pointCategoryCode": "POINTS",
      "amountProvided": false,
      "minimumQuantity": 1
    },
    "fulfillment": {
      "fulfillmentPathCode": "COUPON"
    }
  }'
```




Responses

● 200

```
// Success
{
  "id": "",
  "code": "GOLD_REWARD_2026",
  "externalCode": "GOLD_REWARD",
  "description": "Gold Member Exclusive Reward",
  "statusCode": "",
  "startDate": "2025-01-01",
  "endDate": "2050-12-31",
  "redemptionTypeCode": "I = Item, T = Travel",
  "rank": 10,
  "redeemable": true,
  "workflowStatus": "",
  "rawExpression": "",
  "faceValue": 500,
  "dataRelease": "",
  "partnerCode": "",
  "minimumRoleRequired": "",
  "quantities": {
    "pointCategoryCode": "POINTS",
    "amountProvided": false,
    "pointsRequired": 5000,
    "providedAmountRate": 100,
    "minimumQuantity": 1,
    "maximumQuantity": "",
    "redeemIncrement": 1
  },
  "fulfillment": {
    "fulfillmentPathCode": "COUPON",
    "communicationTypeCode": "RedemptionConfirmationEmail",
    "couponAward": "",
    "redemptionRuleBonusCode": "",
    "merchant": ""
  },
  "content": ""
}
```



14.20.5. Update Redemption Rule

PATCH

`https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules/{code}`

Try it ▶

Performs a partial update on a redemption rule. Only fields included in the request body are modified; absent fields remain unchanged. The following fields can be explicitly set to null to clear their value: endDate, faceValue, minimumRoleRequired, rank, rawExpression, externalCode.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:redemption:modify

Authorize

Use Your Own Token

Path Parameters

`code string *`

Body Parameters

▼ `body RedemptionRulePatchRequest *`> `description string`> `statusCode string`> `startDate string<date>`

mm/dd/yyyy

> `endDate string<date> nullable`

mm/dd/yyyy

> `redeemable boolean`> `faceValue number nullable`> `minimumRoleRequired integer<int64> nullable`

Format: int64

> `rank integer<int32> nullable`

Format: int32

> `rawExpression string nullable`> `externalCode string nullable`> `content string nullable`

Responses

▼ ● 200 Success



▼ RedemptionRule

```
> id integer<int64>
> code string *
> externalCode string
> description string *
> statusCode string
> startDate string<date> *
> endDate string<date>
> redemptionTypeCode string *
> rank integer<int32>
> redeemable boolean
> workflowStatus string
> rawExpression string
> faceValue number
> dataRelease integer<int64>
> partnerCode string
> minimumRoleRequired integer<int64>
> quantities RedemptionRuleQuantities *
> fulfillment RedemptionRuleFulfillment
> content string
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 404 Redemption Rule not found
- 500 Internal error



Request



```
curl --request PATCH \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules/{code} \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "id": "",  
  "code": "GOLD_REWARD_2026",  
  "externalCode": "GOLD_REWARD",  
  "description": "Gold Member Exclusive Reward",  
  "statusCode": "",  
  "startDate": "2025-01-01",  
  "endDate": "2050-12-31",  
  "redemptionTypeCode": "I = Item, T = Travel",  
  "rank": 10,  
  "redeemable": true,  
  "workflowStatus": "",  
  "rawExpression": "",  
  "faceValue": 500,  
  "dataRelease": "",  
  "partnerCode": "",  
  "minimumRoleRequired": "",  
  "quantities": {  
    "pointCategoryCode": "POINTS",  
    "amountProvided": false,  
    "pointsRequired": 5000,  
    "providedAmountRate": 100,  
    "minimumQuantity": 1,  
    "maximumQuantity": "",  
    "redeemIncrement": 1  
  },  
  "fulfillment": {  
    "fulfillmentPathCode": "COUPON",  
    "communicationTypeCode": "RedemptionConfirmationEmail",  
    "couponAward": "",  
    "redemptionRuleBonusCode": "",  
    "merchant": ""  
  },  
  "content": ""  
}
```



14.20.6. Delete Redemption Rule

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules/{code}`

Try it

Delete an existing redemption rule

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:redemption:modify

Authorize

Use Your Own Token

Path Parameters

`code string *`

Responses

● 200 Success

▾ ● 400 Bad request

▾ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/redemption-rules/{code} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.21. reward



14.21.1. Find Redemption Rewards

GET

`https://api.b.preview.loyaltyservices.io/api/v3/rewards/redemption-rules`

Try it ▶

Search for redemption rewards. When the `memberIdentifier` query parameter is provided, the `redeemable`, `poolRedeemable`, and `redeemText` fields are computed for that member. Otherwise, `redeemable` is false, and `poolRedeemable` and `redeemText` are omitted. This consumer-facing service returns the redemption options available to a member. Targeted redemption rewards are included only when `memberIdentifier` is provided.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:member-redemption:read`

Authorize

Use Your Own Token

Query Parameters

> `memberIdentifier` string> `code` string> `typeCode` string

Responses

▼ ● 200 Success

▼ FindRedemptionRewardsResponse

> `redemptionRewards` RedemptionReward[]

▼ ● 400 Bad request

▼ ErrorResponse

> `errors` ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/rewards/redemption-rules \  
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "redemptionRewards": [
    {
      "code": "GOLD_REWARD_2026",
      "typeCode": "I = Item, T = Travel",
      "description": "Gold Member Exclusive Reward",
      "currencyRequired": 5000,
      "minimumQuantity": 1,
      "maximumQuantity": 10,
      "redeemIncrement": 1,
      "faceValue": 500,
      "amountProvided": false,
      "companyCode": "",
      "sortValue": "",
      "rewardContext": {
        "redeemable": false,
        "poolRedeemable": false,
        "redeemText": "Click to Redeem",
        "requireAddress": false,
        "requirePartnerAccountNumber": false
      }
    }
  ]
}
```




14.22. smart-group

14.22.1. Get Smart Groups

GET

<https://api.b.preview.loyaltyservices.io/api/v3/smart-groups>

Try it

Get the Smart Groups in the system. Smart Groups allow for creating collections of Ace, DataCode, and Date data that can be referenced in Tally.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:smartgroup:read

Authorize

Use Your Own Token

Query Parameters

> includeSmartGroupDetails boolean



Responses

✓ 200 Success

SmartGroup[]

```
> smartGroupId integer<int64>
> typeCode string *
> typeDescription string
> name string *
> description string
> statusCode string
> isExclusive boolean
> details SmartGroupDetail[]
> updateDateTime string<date-time>
```

✓ 400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/smart-groups \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "smartGroupId": "",
    "typeCode": "",
    "typeDescription": "",
    "name": "",
    "description": "",
    "statusCode": "A",
    "isExclusive": false,
    "details": [
      {
        "smartGroupDetailId": "",
        "dataId": "",
        "data": "",
        "dataKey": "",
        "dataKeyDescription": "",
        "updateDateTime": ""
      }
    ],
    "updateDateTime": ""
  }
]
```



14.22.2. Create Smart Group

POST

<https://api.b.preview.loyaltyservices.io/api/v3/smart-groups>

Try it ▶

Create a new smart group

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:smartgroup:modify

Authorize

Use Your Own Token

Body Parameters

▼ body SmartGroup *

Description

Create Smart Group Request Object

> typeCode string *

> name string *

> description string

> statusCode string

> isExclusive boolean

> details SmartGroupDetail[]

☐

Responses

▼ ● 200 Success



SmartGroup

- > smartGroupId integer<int64>
- > typeCode string *
- > typeDescription string
- > name string *
- > description string
- > statusCode string
- > isExclusive boolean
- > details SmartGroupDetail[]
- > updateDateTime string<date-time>

● 400 Bad request

ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/smart-groups \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "typeCode": "",
    "name": "",
    "statusCode": "A",
    "details": [
      {
        "dataKey": ""
      }
    ]
  }'
```



Responses

● 200

```
// Success
{
  "smartGroupId": "",
  "typeCode": "",
  "typeDescription": "",
  "name": "",
  "description": "",
  "statusCode": "A",
  "isExclusive": false,
  "details": [
    {
      "smartGroupDetailId": "",
      "dataId": "",
      "data": "",
      "dataKey": "",
      "dataKeyDescription": "",
      "updateDateTime": ""
    }
  ],
  "updateDateTime": ""
}
```



14.22.3. Get Smart Group

GET

`https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId}`

Try it

Get information about a smart group

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:smartgroup:read`

Authorize

Use Your Own Token

Path Parameters

> `smartGroupId` `integer<int64>` *

Format: int64

Query Parameters

> `includeSmartGroupDetails` `boolean`

Responses

✓ ● 200 Success

SmartGroup

> `smartGroupId` `integer<int64>`

> `typeCode` `string` *

> `typeDescription` `string`

> `name` `string` *

> `description` `string`

> `statusCode` `string`

> `isExclusive` `boolean`

> `details` `SmartGroupDetail[]`

> `updateDateTime` `string<date-time>`

✓ ● 400 Bad request

ErrorResponse

> `errors` `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId} \
  --header 'accept: application/json'
```

Responses



```
// Success
{
  "smartGroupId": "",
  "typeCode": "",
  "typeDescription": "",
  "name": "",
  "description": "",
  "statusCode": "A",
  "isExclusive": false,
  "details": [
    {
      "smartGroupDetailId": "",
      "dataId": "",
      "data": "",
      "dataKey": "",
      "dataKeyDescription": "",
      "updateDateTime": ""
    }
  ],
  "updateDateTime": ""
}
```



14.22.4. Get Smart Group Details

GET

`https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId}/details`

Try it ▶

Get smart group detail information

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:smartgroup:read

Authorize

Use Your Own Token

Path Parameters

> smartGroupId integer<int64> *

Format: int64

Responses

▼ ● 200 Success

▼ SmartGroupDetail[]

> smartGroupDetailId integer<int64>
> dataId integer<int64>
> data string
> dataKey string *
> dataKeyDescription string
> updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId}/details \
  --header 'accept: application/json'
```




Responses

● 200

```
// Success
[
  {
    "smartGroupDetailId": "",
    "dataId": "",
    "data": "",
    "dataKey": "",
    "dataKeyDescription": "",
    "updateDateTime": ""
  }
]
```



14.22.5. Add Smart Group Detail Record

POST

https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId}/details

Try it ▶

Adds new smart group detail record. This operation will error if the detail record already exists.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:smartgroup:modify

Authorize

Use Your Own Token

Path Parameters

> smartGroupId integer<int64> *

Format: int64

Query Parameters

> async boolean

Body Parameters

▼ body SmartGroupDetail *

Description

Create Smart Group Detail

> dataKey string *

Responses

▼ ● 200 Success

▼ SmartGroupDetail

> smartGroupDetailId integer<int64>

> dataId integer<int64>

> data string

> dataKey string *

> dataKeyDescription string

> updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized



- **403** Forbidden
- **500** Internal error

Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId}/details?async=boolean' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "dataKey": ""
  }'
```

Responses



```
// Success
{
  "smartGroupDetailId": "",
  "dataId": "",
  "data": "",
  "dataKey": "",
  "dataKeyDescription": "",
  "updateDateTime": ""
}
```



14.22.6. Get Smart Group Detail Information

GET

`https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId}/details/{smartGroupDetailId}`

Try it ▶

Get smart group detail information

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:smartgroup:read

Authorize

Use Your Own Token

Path Parameters

> smartGroupId integer<int64> *

Format: int64

> smartGroupDetailId integer<int64> *

Format: int64

Responses

▼ ● 200 Success

▼ SmartGroupDetail

> smartGroupDetailId integer<int64>

> dataId integer<int64>

> data string

> dataKey string *

> dataKeyDescription string

> updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId}/details/{smartGroupDetailId} \  
  --header 'accept: application/json'
```

Responses



```
// Success  
{  
  "smartGroupDetailId": "",  
  "dataId": "",  
  "data": "",  
  "dataKey": "",  
  "dataKeyDescription": "",  
  "updateDateTime": ""  
}
```



14.22.7. Delete (remove) Smart Group Detail record

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId}/details/{smartGroupDetailId}`

Try it ▶

Remove smart group detail from smart group

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:smartgroup:modify

Authorize

Use Your Own Token

Path Parameters

> **smartGroupId** `integer<int64>` *

Format: int64

> **smartGroupDetailId** `integer<int64>` *

Format: int64

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/smart-groups/{smartGroupId}/details/{smartGroupDetailId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.23. member-transfer-points



14.23.1. Transfer Points

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{accountNumber}/transfer`

Try it

Transfers points from one member to another

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-transfer:transfer

Authorize

Use Your Own Token

Path Parameters

> **accountNumber** string *

Body Parameters

▾ **body** PointsTransferRequest *> **partnerTransactionNumber** string> **rewardCode** string> **recipientAccountNumber** string> **bonusCode** string> **transferAmount** integer<int32>> **transferredBy** string> **customData** CustomData

Responses

▾ ● 200 Success



▼ PointsTransferResponse

- > partnerTransactionNumber string
- > accountNumber string
- > correctedAccountNumber string
- > rewardCode string
- > orderNumber integer<int32>
- > recipientAccountNumber string
- > correctedRecipientAccountNumber string
- > bonusCode string
- > transactionId string
- > transferAmount integer<int32>

▼ ● 400 Bad Request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 500 Internal Error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{accountNumber}/transfer \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
{
  "partnerTransactionNumber": "",
  "accountNumber": "",
  "correctedAccountNumber": "",
  "rewardCode": "",
  "orderNumber": "",
  "recipientAccountNumber": "",
  "correctedRecipientAccountNumber": "",
  "bonusCode": "",
  "transactionId": "EN-321654211",
  "transferAmount": ""
}
```



14.24. member-comment

14.24.1. Get Member Comments

GET

`https://api.b.preview.loyaltyservices.io/api/v4/members/{memberIdentifier}/comments`

Try it ▶

Retrieve a paginated list of comments of a member.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:member-comments:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Query Parameters

> `startDate` `string<date>`

mm/dd/yyyy

> `endDate` `string<date>`

mm/dd/yyyy

> `includeClosed` `boolean`☐> `commentReason` `string`> `pageNumber` `integer`

1

> `pageSize` `integer`

500

Responses

✓ ● 200 Success

▼ MemberCommentResponseV4

> `results` `MemberComment[]`> `pageInfo` `PageInfo`

✓ ● 400 Bad request

▼ ErrorResponse

> `errors` `ValidationError[]`



- **401** Unauthorized
- **403** Forbidden
- **500** Internal error

Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v4/members/{memberIdentifier}/comments?includeClosed=boolean&pageNumber=1&p  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "results": [
    {
      "memberCommentId": "",
      "memberCommentType": "",
      "memberIdentifier": "",
      "accountNumber": "",
      "companyName": "",
      "memberName": "",
      "memberType": "",
      "commentName": "",
      "title": "",
      "status": "",
      "followUpFlag": false,
      "followUpDate": "",
      "assignedTo": "",
      "createDateTime": "",
      "updateDateTime": "",
      "exceedsSLA": false,
      "locked": false,
      "lockerId": "",
      "lockerName": "",
      "lockerRank": "",
      "closedDate": "",
      "reason": "",
      "subReason": "",
      "correspondenceToSendToMemberId": "",
      "externalReference": "",
      "comments": [
        {
          "createDate": "",
          "createdBy": "",
          "userEmail": "",
          "comment": "",
          "assignedToUser": "",
          "assignedToRole": "",
          "reassigned": false,
          "assignedFromUser": "",
          "assignedFromRole": ""
        }
      ],
      "customData": {
        "ace01": "",

```



14.24.2. Get Member Comments / Call Logs

GET

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/comments

Try it ▶

Retrieve list of comments / call logs for a member.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-comments:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> startDate string<date>

mm/dd/yyyy



> endDate string<date>

mm/dd/yyyy



> includeClosed boolean

☐

> commentReason string

> title string

> text string

Responses

▼ • 200 Success



▼ MemberComment[]

- > memberCommentId integer<int64>
- > memberCommentType string
- > memberIdentifier string
- > accountNumber string
- > companyName string
- > memberName string
- > memberType string
- > commentName string
- > title string
- > status string
- > followUpFlag boolean
- > followUpDate string<date>
- > assignedTo string
- > createDateTime string<date-time>
- > updateDateTime string<date-time>
- > exceedsSLA boolean
- > locked boolean
- > lockerId integer<int64>
- > lockerName string
- > lockerRank integer<int32>
- > closedDate string<date>
- > reason string
- > subReason string
- > correspondenceToSendToMemberId integer<int64>
- > externalReference string
- > comments MemberCommentDetail[]
- > customData CustomData
- > externalProgram string



- > externalProgramStatus string
- > sourceApplicationIP string
- > commentCloseReason string
- > memberCommentImages MemberCommentImageData[]
- > closedDateTime string<date-time>
- > createdBy string
- > updatedBy string
- > firstComment string
- > statusDescription string
- > reasonDescription string
- > subReasonDescription string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

[Curl](#)

```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/comments?includeClosed=boolean' \  
  --header 'accept: application/json'
```




Responses

● 200



```
// Success
[
  {
    "memberCommentId": "",
    "memberCommentType": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "companyName": "",
    "memberName": "",
    "memberType": "",
    "commentName": "",
    "title": "",
    "status": "",
    "followUpFlag": false,
    "followUpDate": "",
    "assignedTo": "",
    "createDateTime": "",
    "updateDateTime": "",
    "exceedsSLA": false,
    "locked": false,
    "lockerId": "",
    "lockerName": "",
    "lockerRank": "",
    "closedDate": "",
    "reason": "",
    "subReason": "",
    "correspondenceToSendToMemberId": "",
    "externalReference": "",
    "comments": [
      {
        "createDate": "",
        "createdBy": "",
        "userEmail": "",
        "comment": "",
        "assignedToUser": "",
        "assignedToRole": ""
      }
    ]
  }
]
```



14.24.3. Create a member comment / call log

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/comments`

Try it ▶

Create a comment / call log for a member

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-comments:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> async boolean



Body Parameters

▼

body MemberCommentRequest *

> memberCommentId integer<int64>

Format: int64

> reason string

> subReason string

> status string *

C

> comment string *

> followUpFlag boolean

☐

> followUpDate string<date>

mm/dd/yyyy

> assignedToCode integer<int64>

Format: int64

> correspondenceToSendToMemberId integer<int64>

Format: int64

> externalReferenceId string

> bonusRuleId integer<int64>

Format: int64

> commentCloseReason string

> commentType string

> title string

> closedDate string<date>

mm/dd/yyyy

Responses

▼

●

200 Success



▼ MemberComment

- > memberCommentId integer<int64>
- > memberCommentType string
- > memberIdentifier string
- > accountNumber string
- > companyName string
- > memberName string
- > memberType string
- > commentName string
- > title string
- > status string
- > followUpFlag boolean
- > followUpDate string<date>
- > assignedTo string
- > createDateTime string<date-time>
- > updateDateTime string<date-time>
- > exceedsSLA boolean
- > locked boolean
- > lockerId integer<int64>
- > lockerName string
- > lockerRank integer<int32>
- > closedDate string<date>
- > reason string
- > subReason string
- > correspondenceToSendToMemberId integer<int64>
- > externalReference string
- > comments MemberCommentDetail[]
- > customData CustomData
- > externalProgram string



- > externalProgramStatus string
- > sourceApplicationIP string
- > commentCloseReason string
- > memberCommentImages MemberCommentImageData[]
- > closedDateTime string<date-time>
- > createdBy string
- > updatedBy string
- > firstComment string
- > statusDescription string
- > reasonDescription string
- > subReasonDescription string

▼ ● 202 Accepted

▼ AsyncTask

- > asyncTaskId integer<int64>
- > taskTypeCode string
- > taskCode string
- > requestDate string<date-time>
- > requestData object
- > statusCode string
- > responseDate string<date-time>
- > responseData object
- > fileName string
- > lineNumber integer
- > fileData string

▼ ● 400 Bad Request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal Error

Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/comments?async=boolean' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "status": "C",
    "comment": "Adding member adjustment to account 12345X"
  }'
```



Responses

● 200



```
// Success
{
  "memberCommentId": "",
  "memberCommentType": "",
  "memberIdentifier": "",
  "accountNumber": "",
  "companyName": "",
  "memberName": "",
  "memberType": "",
  "commentName": "",
  "title": "",
  "status": "",
  "followUpFlag": false,
  "followUpDate": "",
  "assignedTo": "",
  "createDateTime": "",
  "updateDateTime": "",
  "exceedsSLA": false,
  "locked": false,
  "lockerId": "",
  "lockerName": "",
  "lockerRank": "",
  "closedDate": "",
  "reason": "",
  "subReason": "",
  "correspondenceToSendToMemberId": "",
  "externalReference": "",
  "comments": [
    {
      "createDate": "",
      "createdBy": "",
      "userEmail": "",
      "comment": "",
      "assignedToUser": ""
    }
  ]
}
```



14.24.4. Get the comment associated with given commentId

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/comments/{commentId}`

Try it ▶

Retrieve a member's comment for given commentId.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-comments:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> commentId integer<int64> *

Format: int64

Responses

▼ ● 200 Success



▼ MemberComment

- > memberCommentId integer<int64>
- > memberCommentType string
- > memberIdentifier string
- > accountNumber string
- > companyName string
- > memberName string
- > memberType string
- > commentName string
- > title string
- > status string
- > followUpFlag boolean
- > followUpDate string<date>
- > assignedTo string
- > createDateTime string<date-time>
- > updateDateTime string<date-time>
- > exceedsSLA boolean
- > locked boolean
- > lockerId integer<int64>
- > lockerName string
- > lockerRank integer<int32>
- > closedDate string<date>
- > reason string
- > subReason string
- > correspondenceToSendToMemberId integer<int64>
- > externalReference string
- > comments MemberCommentDetail[]
- > customData CustomData
- > externalProgram string



- > externalProgramStatus string
- > sourceApplicationIP string
- > commentCloseReason string
- > memberCommentImages MemberCommentImageData[]
- > closedDateTime string<date-time>
- > createdBy string
- > updatedBy string
- > firstComment string
- > statusDescription string
- > reasonDescription string
- > subReasonDescription string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

[Curl](#)

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/comments/{commentId} \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "memberCommentId": "",
  "memberCommentType": "",
  "memberIdentifier": "",
  "accountNumber": "",
  "companyName": "",
  "memberName": "",
  "memberType": "",
  "commentName": "",
  "title": "",
  "status": "",
  "followUpFlag": false,
  "followUpDate": "",
  "assignedTo": "",
  "createDateTime": "",
  "updateDateTime": "",
  "exceedsSLA": false,
  "locked": false,
  "lockerId": "",
  "lockerName": "",
  "lockerRank": "",
  "closedDate": "",
  "reason": "",
  "subReason": "",
  "correspondenceToSendToMemberId": "",
  "externalReference": "",
  "comments": [
    {
      "createDate": "",
      "createdBy": "",
      "userEmail": "",
      "comment": "",
      "assignedToUser": ""
    }
  ]
}
```



14.24.5. Update Member Comments

PUT`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/comments/{commentId}`

Try it ▶

Update Member Comments

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-comments:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> commentId string *



Body Parameters

▼

body MemberCommentRequest *

> memberCommentId integer<int64>

Format: int64

> reason string

> subReason string

> status string *

C

> comment string *

> followUpFlag boolean

☐

> followUpDate string<date>

mm/dd/yyyy

> assignedToCode integer<int64>

Format: int64

> correspondenceToSendToMemberId integer<int64>

Format: int64

> externalReferenceId string

> bonusRuleId integer<int64>

Format: int64

> commentCloseReason string

> commentType string

> title string

> closedDate string<date>

mm/dd/yyyy

Responses

▼

●

200 Success



- ▼ MemberComment
- > memberCommentId integer<int64>
 - > memberCommentType string
 - > memberIdentifier string
 - > accountNumber string
 - > companyName string
 - > memberName string
 - > memberType string
 - > commentName string
 - > title string
 - > status string
 - > followUpFlag boolean
 - > followUpDate string<date>
 - > assignedTo string
 - > createDateTime string<date-time>
 - > updateDateTime string<date-time>
 - > exceedsSLA boolean
 - > locked boolean
 - > lockerId integer<int64>
 - > lockerName string
 - > lockerRank integer<int32>
 - > closedDate string<date>
 - > reason string
 - > subReason string
 - > correspondenceToSendToMemberId integer<int64>
 - > externalReference string
 - > comments MemberCommentDetail[]
 - > customData CustomData
 - > externalProgram string



- > externalProgramStatus string
- > sourceApplicationIP string
- > commentCloseReason string
- > memberCommentImages MemberCommentImageData[]
- > closedDateTime string<date-time>
- > createdBy string
- > updatedBy string
- > firstComment string
- > statusDescription string
- > reasonDescription string
- > subReasonDescription string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

[Curl](#)

```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/comments/{commentId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "status": "C",
    "comment": "Adding member adjustment to account 12345X"
  }'
```



Responses

● 200



```
// Success
{
  "memberCommentId": "",
  "memberCommentType": "",
  "memberIdentifier": "",
  "accountNumber": "",
  "companyName": "",
  "memberName": "",
  "memberType": "",
  "commentName": "",
  "title": "",
  "status": "",
  "followUpFlag": false,
  "followUpDate": "",
  "assignedTo": "",
  "createDateTime": "",
  "updateDateTime": "",
  "exceedsSLA": false,
  "locked": false,
  "lockerId": "",
  "lockerName": "",
  "lockerRank": "",
  "closedDate": "",
  "reason": "",
  "subReason": "",
  "correspondenceToSendToMemberId": "",
  "externalReference": "",
  "comments": [
    {
      "createDate": "",
      "createdBy": "",
      "userEmail": "",
      "comment": "",
      "assignedToUser": ""
    }
  ]
}
```



14.24.6. Update Status for Member Comments

PUT

https://api.b.preview.loyaltyservices.io/api/v3/members/comments/status

Try it ▶

Update the status of multiple member comments based on their type

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-comments:modify

Authorize

Use Your Own Token

Body Parameters

▼ body object *

commentIds integer[] *

0

+

> status string *

Responses

▼ • 200 Success



▼ MemberComment[]

- > memberCommentId integer<int64>
- > memberCommentType string
- > memberIdentifier string
- > accountNumber string
- > companyName string
- > memberName string
- > memberType string
- > commentName string
- > title string
- > status string
- > followUpFlag boolean
- > followUpDate string<date>
- > assignedTo string
- > createDateTime string<date-time>
- > updateDateTime string<date-time>
- > exceedsSLA boolean
- > locked boolean
- > lockerId integer<int64>
- > lockerName string
- > lockerRank integer<int32>
- > closedDate string<date>
- > reason string
- > subReason string
- > correspondenceToSendToMemberId integer<int64>
- > externalReference string
- > comments MemberCommentDetail[]
- > customData CustomData
- > externalProgram string



- > externalProgramStatus string
- > sourceApplicationIP string
- > commentCloseReason string
- > memberCommentImages MemberCommentImageData[]
- > closedDateTime string<date-time>
- > createdBy string
- > updatedBy string
- > firstComment string
- > statusDescription string
- > reasonDescription string
- > subReasonDescription string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

[Curl](#)

```
curl --request PUT \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/comments/status \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "commentIds": [],  
    "status": ""  
  }'
```



Responses

● 200



```
// Success
[
  {
    "memberCommentId": "",
    "memberCommentType": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "companyName": "",
    "memberName": "",
    "memberType": "",
    "commentName": "",
    "title": "",
    "status": "",
    "followUpFlag": false,
    "followUpDate": "",
    "assignedTo": "",
    "createDateTime": "",
    "updateDateTime": "",
    "exceedsSLA": false,
    "locked": false,
    "lockerId": "",
    "lockerName": "",
    "lockerRank": "",
    "closedDate": "",
    "reason": "",
    "subReason": "",
    "correspondenceToSendToMemberId": "",
    "externalReference": "",
    "comments": [
      {
        "createDate": "",
        "createdBy": "",
        "userEmail": "",
        "comment": "",
        "assignedToUser": "",
        "assignedToRole": ""
      }
    ]
  }
]
```



14.24.7. Retrieve member comments

GET

<https://api.b.preview.loyaltyservices.io/api/v3/members/comments>

Try it

Retrieve member comments and/or user tasks

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-comments:read

Authorize

Use Your Own Token

Query Parameters

> startDate string<date>

mm/dd/yyyy



> endDate string<date>

mm/dd/yyyy



> includeClosed boolean

☐

> taskType string

> assignee string

> commentReason string

> accountNumber string

> firstName string

> lastName string

> emailAddress string

> statuses string[]

+

> title string

Responses

✓ ● 200 Success



▼ MemberComment[]

- > memberCommentId integer<int64>
- > memberCommentType string
- > memberIdentifier string
- > accountNumber string
- > companyName string
- > memberName string
- > memberType string
- > commentName string
- > title string
- > status string
- > followUpFlag boolean
- > followUpDate string<date>
- > assignedTo string
- > createDateTime string<date-time>
- > updateDateTime string<date-time>
- > exceedsSLA boolean
- > locked boolean
- > lockerId integer<int64>
- > lockerName string
- > lockerRank integer<int32>
- > closedDate string<date>
- > reason string
- > subReason string
- > correspondenceToSendToMemberId integer<int64>
- > externalReference string
- > comments MemberCommentDetail[]
- > customData CustomData
- > externalProgram string



- > externalProgramStatus string
- > sourceApplicationIP string
- > commentCloseReason string
- > memberCommentImages MemberCommentImageData[]
- > closedDateTime string<date-time>
- > createdBy string
- > updatedBy string
- > firstComment string
- > statusDescription string
- > reasonDescription string
- > subReasonDescription string

▼ ● 400 Bad Request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal Server Error

Request

Curl

```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/comments?includeClosed=boolean' \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
[
  {
    "memberCommentId": "",
    "memberCommentType": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "companyName": "",
    "memberName": "",
    "memberType": "",
    "commentName": "",
    "title": "",
    "status": "",
    "followUpFlag": false,
    "followUpDate": "",
    "assignedTo": "",
    "createDateTime": "",
    "updateDateTime": "",
    "exceedsSLA": false,
    "locked": false,
    "lockerId": "",
    "lockerName": "",
    "lockerRank": "",
    "closedDate": "",
    "reason": "",
    "subReason": "",
    "correspondenceToSendToMemberId": "",
    "externalReference": "",
    "comments": [
      {
        "createDate": "",
        "createdBy": "",
        "userEmail": "",
        "comment": "",
        "assignedToUser": "",
        "assignedToRole": ""
      }
    ]
  }
]
```



14.24.8. Create member comments for multiple members

POST

<https://api.b.preview.loyaltyservices.io/api/v3/members/comments>

Try it ▶

Create member comments for multiple members

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-comments:modify

Authorize

Use Your Own Token

Body Parameters

body MemberCommentBulkRequest *

accountIds string[] *

> reason string *

> title string *

> closedDate string<date> *

> comment string *

+

mm/dd/yyyy



Responses

200 Success

MemberCommentBulkResponse

> totalRecords integer<int32>

> totalRecordsUpdated integer<int32>

> totalRecordsFailed integer<int32>

> errorDetails ErrorResponse

400 Bad Request

ErrorResponse

> errors ValidationError[]

401 Unauthorized



- 403 Forbidden
- 500 Internal Error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/comments \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "accountIds": [],
    "reason": "",
    "title": "",
    "closedDate": "2023-04-06",
    "comment": ""
  }'
```

Responses



```
// Success
{
  "totalRecords": 1000,
  "totalRecordsUpdated": 950,
  "totalRecordsFailed": 50,
  "errorDetails": {
    "errors": [
      {
        "code": "",
        "fieldName": "",
        "message": "",
        "messageParameters": [
          {}
        ]
      }
    ]
  }
}
```




14.24.9. Lock call log

POST

https://api.b.preview.loyaltyservices.io/api/v3/call-logs/{memberCommentId}/locks

Try it ▶

Lock a call log

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-comments:modify

Authorize

Use Your Own Token

Path Parameters

> memberCommentId integer<int64> *

Format: int64

Responses

▼ ● 200 Success



▼ MemberComment

- > memberCommentId integer<int64>
- > memberCommentType string
- > memberIdentifier string
- > accountNumber string
- > companyName string
- > memberName string
- > memberType string
- > commentName string
- > title string
- > status string
- > followUpFlag boolean
- > followUpDate string<date>
- > assignedTo string
- > createDateTime string<date-time>
- > updateDateTime string<date-time>
- > exceedsSLA boolean
- > locked boolean
- > lockerId integer<int64>
- > lockerName string
- > lockerRank integer<int32>
- > closedDate string<date>
- > reason string
- > subReason string
- > correspondenceToSendToMemberId integer<int64>
- > externalReference string
- > comments MemberCommentDetail[]
- > customData CustomData
- > externalProgram string



- > externalProgramStatus string
- > sourceApplicationIP string
- > commentCloseReason string
- > memberCommentImages MemberCommentImageData[]
- > closedDateTime string<date-time>
- > createdBy string
- > updatedBy string
- > firstComment string
- > statusDescription string
- > reasonDescription string
- > subReasonDescription string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

[Curl](#)

```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/call-logs/{memberCommentId}/locks \  
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "memberCommentId": "",
  "memberCommentType": "",
  "memberIdentifier": "",
  "accountNumber": "",
  "companyName": "",
  "memberName": "",
  "memberType": "",
  "commentName": "",
  "title": "",
  "status": "",
  "followUpFlag": false,
  "followUpDate": "",
  "assignedTo": "",
  "createDateTime": "",
  "updateDateTime": "",
  "exceedsSLA": false,
  "locked": false,
  "lockerId": "",
  "lockerName": "",
  "lockerRank": "",
  "closedDate": "",
  "reason": "",
  "subReason": "",
  "correspondenceToSendToMemberId": "",
  "externalReference": "",
  "comments": [
    {
      "createDate": "",
      "createdBy": "",
      "userEmail": "",
      "comment": "",
      "assignedToUser": "",
    }
  ]
}
```



14.24.10. Unlock call log

DELETE

https://api.b.preview.loyaltyservices.io/api/v3/call-logs/{memberCommentId}/locks

Try it ▶

Unlock a call log

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-comments:modify

Authorize

Use Your Own Token

Path Parameters

> memberCommentId integer<int64> *

Format: int64

Responses

▼ ● 200 Success



▼ MemberComment

- > memberCommentId integer<int64>
- > memberCommentType string
- > memberIdentifier string
- > accountNumber string
- > companyName string
- > memberName string
- > memberType string
- > commentName string
- > title string
- > status string
- > followUpFlag boolean
- > followUpDate string<date>
- > assignedTo string
- > createDateTime string<date-time>
- > updateDateTime string<date-time>
- > exceedsSLA boolean
- > locked boolean
- > lockerId integer<int64>
- > lockerName string
- > lockerRank integer<int32>
- > closedDate string<date>
- > reason string
- > subReason string
- > correspondenceToSendToMemberId integer<int64>
- > externalReference string
- > comments MemberCommentDetail[]
- > customData CustomData
- > externalProgram string



- > externalProgramStatus string
- > sourceApplicationIP string
- > commentCloseReason string
- > memberCommentImages MemberCommentImageData[]
- > closedDateTime string<date-time>
- > createdBy string
- > updatedBy string
- > firstComment string
- > statusDescription string
- > reasonDescription string
- > subReasonDescription string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

[Curl](#)

```
curl --request DELETE \  
--url https://api.b.preview.loyaltyservices.io/api/v3/call-logs/{memberCommentId}/locks \  
--header 'accept: application/json'
```




Responses

● 200



```
// Success
{
  "memberCommentId": "",
  "memberCommentType": "",
  "memberIdentifier": "",
  "accountNumber": "",
  "companyName": "",
  "memberName": "",
  "memberType": "",
  "commentName": "",
  "title": "",
  "status": "",
  "followUpFlag": false,
  "followUpDate": "",
  "assignedTo": "",
  "createDateTime": "",
  "updateDateTime": "",
  "exceedsSLA": false,
  "locked": false,
  "lockerId": "",
  "lockerName": "",
  "lockerRank": "",
  "closedDate": "",
  "reason": "",
  "subReason": "",
  "correspondenceToSendToMemberId": "",
  "externalReference": "",
  "comments": [
    {
      "createDate": "",
      "createdBy": "",
      "userEmail": "",
      "comment": "",
      "assignedToUser": ""
    }
  ]
}
```



14.25. member-badge

14.25.1. Get Member badges

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/badges`

Try it ▶

Retrieve a member's badges

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:member-badge:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Responses

✓ ● 200 Success

▼ MemberBadge[]

> `memberBadgeld` `integer<int64>`

> `badgeDetails` `BadgeData`

> `sourcecode` `string`

> `statusCode` `string`

> `startDate` `string<date>`

> `endDate` `string<date>`

> `updateUserName` `string`

> `createUserName` `string`

> `sourceDescription` `string`

✓ ● 400 Bad request

▼ ErrorResponse

> `errors` `ValidationError[]`



- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/badges \  
  --header 'accept: application/json'
```

Responses



```
// Success  
[  
  {  
    "memberBadgeId": 147258369,  
    "badgeDetails": {  
      "badgeId": 147258369,  
      "badgeTypeCode": "General",  
      "badgeName": "Event Organizer",  
      "badgeDescription": "An organizer who brings in over $100,000 year in business",  
      "statusCode": "Active",  
      "expirationDate": "4000-01-01",  
      "expirationType": "Exact",  
      "expirationDayPeriod": 60,  
      "expirationTimePeriod": 10,  
      "logoKey": "elite-explorer.png",  
      "content": {  
        "contentCode": "Bonus100",  
        "contentTypeCode": "HEADER",  
        "name": "Bonus 100 points",  
        "shortDescription": "Award 100 points",  
        "description": "Award 100 points on 100th stay",  
        "alternateDescription": "Award 100 points on 100th stay",  
        "termsAndConditions": "Terms and conditions for this promotion",  
        "title": "Award 100 points",  
        "label": "100 points awarded",  
        "logoKey": "100-points.png",  
        "displayTypeKey": "100 points",  
        "startDate": "",  
        "endDate": "",  
        "url": "/loyaltyprogram/bonus/100-points",  
        "tierCode": "B",  
        "displayForAll": true,  
        "descriptorCode": "LIMITED_TIME"  
      }  
    },  
    "sourcecode": "Promotion",  
    "statusCode": "Activated",  
    "startDate": "2020-01-01",  
    "endDate": "2020-12-31",  
    "updateUserName": "",  
    "createUserName": "",  
    "sourceDescription": ""  
  }  
]
```





14.25.2. Create Member badge

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/badges/{badgeId}`

Try it

Award a badge to a member

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:member-badge:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

> badgeId integer<int64> *

Format: int64

Body Parameters

body CreateMemberBadgeRequest *

> comment string *

Responses

200 Success

MemberBadge

> memberBadgeId integer<int64>

> badgeDetails BadgeData

> sourcecode string

> statusCode string

> startDate string<date>

> endDate string<date>

> updateUserName string

> createUserName string

> sourceDescription string

400 Bad request

 ErrorResponse

> errors `ValidationError[]`

- **401** Unauthorized
- **403** Forbidden
- **500** Internal error

Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/badges/{badgeId} \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "comment": ""  
  }'
```



Responses

● 200

```
// Success
{
  "memberBadgeId": 147258369,
  "badgeDetails": {
    "badgeId": 147258369,
    "badgeTypeCode": "General",
    "badgeName": "Event Organizer",
    "badgeDescription": "An organizer who brings in over $100,000 year in business",
    "statusCode": "Active",
    "expirationDate": "4000-01-01",
    "expirationType": "Exact",
    "expirationDayPeriod": 60,
    "expirationTimePeriod": 10,
    "logoKey": "elite-explorer.png",
    "content": {
      "contentCode": "Bonus100",
      "contentTypeCode": "HEADER",
      "name": "Bonus 100 points",
      "shortDescription": "Award 100 points",
      "description": "Award 100 points on 100th stay",
      "alternateDescription": "Award 100 points on 100th stay",
      "termsAndConditions": "Terms and conditions for this promotion",
      "title": "Award 100 points",
      "label": "100 points awarded",
      "logoKey": "100-points.png",
      "displayTypeKey": "100 points",
      "startDate": "",
      "endDate": "",
      "url": "/loyaltyprogram/bonus/100-points",
      "tierCode": "B",
      "displayForAll": true,
      "descriptorCode": "LIMITED_TIME"
    }
  },
  "sourcecode": "Promotion",
  "statusCode": "Activated",
  "startDate": "2020-01-01",
  "endDate": "2020-12-31",
  "updateUserName": "",
  "createUserName": "",
  "sourceDescription": ""
}
```




14.25.3. Delete Member Badge

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/badges/{memberBadgeId}`

Try it ▶

Remove badge from member

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-badge:modify

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** *string* *

> **memberBadgeId** *integer<int64>* *

Format: int64

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** *ValidationError[]*

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/badges/{memberBadgeId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.26. coupon-compensation

14.26.1. Get Compensation by Coupon Code

GET

`https://api.b.preview.loyaltyservices.io/api/v3/compensation/coupons/{couponCode}`

Try it ▶

Retrieve coupon compensations.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:coupon-compensation:read`

Authorize

Use Your Own Token

Path Parameters

> `couponCode` string *

Responses

▼ • 200 Success

▼ CompensationsResponse

> `compensations` CompensationDetail[]

▼ • 400 Bad request

▼ ErrorResponse

> `errors` ValidationError[]

• 401 Unauthorized

• 403 Forbidden

• 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/compensation/coupons/{couponCode} \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "compensations": [
    {
      "accountingPeriod": "",
      "runStatus": "Processed",
      "type": "Calculated",
      "reason": "NORULE",
      "couponCode": "",
      "totalAmount": 0,
      "compensationDetails": [
        {
          "couponUseAmount": 0,
          "confirmationNumber": "",
          "startDateTime": "",
          "endDateTime": "",
          "customData": {
            "ace01": "",
            "ace01Description": "",
            "ace02": "",
            "ace02Description": "",
            "ace03": "",
            "ace03Description": "",
            "ace04": "",
            "ace04Description": "",
            "ace05": "",
            "ace05Description": "",
            "ace06": "",
            "ace06Description": "",
            "ace07": "",
            "ace07Description": "",
            "ace08": "",
            "ace08Description": "",
            "ace09": "",
            "ace09Description": "",
            "ace10": "",
            "ace10Description": "",
            "ace11": "",
            "ace11Description": "",
            "ace12": "",
            "ace12Description": "",
            "ace13": "",
            "ace13Description": "",
            "ace14": "",
            "ace14Description": "",
            "ace15": "",
            "ace15Description": ""
          }
        }
      ]
    }
  ]
}
```





14.26.2. Get Compensation by Coupon Code or Confirmation Number

GET

`https://api.b.preview.loyaltyservices.io/api/v4/compensation`

Try it ▶

Retrieve coupon compensations.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:coupon-compensation:read`

Authorize

Use Your Own Token

Query Parameters

> `couponCode` string

> `confirmationNumber` string

Responses

▼ ● 200 Success

▼ CompensationsResponseV4

> `compensations` Compensation[]

▼ ● 400 Bad request

▼ ErrorResponse

> `errors` ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

📄 Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v4/compensation \
  --header 'accept: application/json'
```



Responses

● 200



```
// Success
{
  "compensations": [
    {
      "compensationId": 0,
      "coupon": {
        "couponId": 147258369,
        "couponRequestId": 147258370,
        "offer": {
          "offerId": 147258369,
          "offerTypeCode": "Discount",
          "code": "ALON2020",
          "name": "Free night",
          "description": "Award double points",
          "featureVersion": 2,
          "statusCode": "Enabled",
          "startDateTime": "",
          "endDateTime": "",
          "offerAttributes": [
            {
              "offerAttributeId": 147258369,
              "attribute": {
                "attributeId": 147258369,
                "typeCode": "Validation",
                "statusCode": "Enabled",
                "name": "AccountNumber",
                "description": "Account Number",
                "dataTypeCode": "open"
              },
              "defaultValue": "Standard",
              "requireOnUse": true,
              "requireOnHold": true,
              "includeOnResponse": false,
              "attributeTypeCode": "Validation",
              "operatorCode": "EQ",
              "operand": "GI",
              "smartGroupId": 147258369
            }
          ]
        },
        "code": "ZPVWTAROW20Q",
        "statusCode": "Activated",
        "startDate": "",
        "endDate": "",
        "useTypeCode": "Single",
        "useLimit": 1,
        "currentUse": 1,
        "couponUses": [
          {
            "useDateTime": "",
            "statusCode": "Used",
            "channelCode": "System",
            "alias": "",
            "useCount": 1,
            "couponUseAttributes": [
              {

```



```
        "couponUseAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
          "description": "Account Number",
          "dataTypeCode": "open"
        },
        "value": "Standard"
      ]
    },
    "couponAttributes": [
      {
        "couponAttributeId": 147258369,
        "attribute": {
          "attributeId": 147258369,
          "typeCode": "Validation",
          "statusCode": "Enabled",
          "name": "AccountNumber",
          "description": "Account Number",
          "dataTypeCode": "open"
        },
        "value": "2021-12-31T12:00:00",
        "requireOnUse": true,
        "requireOnHold": false
      }
    ]
  }
}
```




14.26.3. Add Compensation Nights

POST

<https://api.b.preview.loyaltyservices.io/api/v4/compensation/{compensationId}/nights>

Try it

Add nights to an existing compensation.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:coupon-compensation:create

Authorize

Use Your Own Token

Path Parameters

> compensationId integer<int64> *

Format: int64

Body Parameters

body CompensationNight[] *

Description

List of Compensation Night items to add

> compensationDate string<date>

mm/dd/yyyy



> compensationNightStatus string

> comments string

> compensationDetailAmounts CompensationDetailAmount[]



Responses

200 Success

CompensationNight[]

> compensationDate string<date>

> compensationNightStatus string

> comments string

> compensationDetailAmounts CompensationDetailAmount[]

400 Bad request

ErrorResponse

> errors ValidationError[]

401 Unauthorized



- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v4/compensation/{compensationId}/nights \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '[
    {
      "compensationDetailAmounts": [
        {
          "couponUseAmount": "",
          "category": "",
          "reason": ""
        }
      ]
    }
  ]'
```

Responses



```
// Success
[
  {
    "compensationDate": "",
    "compensationNightStatus": "Processed",
    "comments": "",
    "compensationDetailAmounts": [
      {
        "couponUseAmount": 0,
        "category": "",
        "reason": ""
      }
    ]
  }
]
```



14.26.4. Edit Compensation Nights

PATCH

<https://api.b.preview.loyaltyservices.io/api/v4/compensation/{compensationId}/nights>

Try it

edit nights on an existing compensation.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:coupon-compensation:modify

Authorize

Use Your Own Token

Path Parameters

> compensationId integer<int64> *

Format: int64

Body Parameters

body CompensationNight[] *

Description

List of Compensation Night items to edit

> compensationDate string<date>

mm/dd/yyyy



> compensationNightStatus string

> comments string

> compensationDetailAmounts CompensationDetailAmount[]



Responses

200 Success

CompensationNight[]

> compensationDate string<date>

> compensationNightStatus string

> comments string

> compensationDetailAmounts CompensationDetailAmount[]

400 Bad request

ErrorResponse

> errors ValidationError[]

401 Unauthorized



- 403 Forbidden
- 500 Internal error

Request



```
curl --request PATCH \
  --url https://api.b.preview.loyaltyservices.io/api/v4/compensation/{compensationId}/nights \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '[
  {
    "compensationDetailAmounts": [
      {
        "couponUseAmount": "",
        "category": "",
        "reason": ""
      }
    ]
  }
]
```

Responses



```
// Success
[
  {
    "compensationDate": "",
    "compensationNightStatus": "Processed",
    "comments": "",
    "compensationDetailAmounts": [
      {
        "couponUseAmount": 0,
        "category": "",
        "reason": ""
      }
    ]
  }
]
```



14.27. member-referral

14.27.1. Get Referrals

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referrals`

Try it ▶

Retrieves referrals that have been created by a member

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:member-referral:read`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Responses

▼ ● 200 Success

▼ `MemberReferral[]`

> `memberReferralId` `integer<int64>`

> `referralType` `string`

> `referralValue` `string`

> `referralDate` `string<date>`

> `referredAccountNumber` `string`

> `active` `boolean`

> `referredEarnedDate` `string<date>`

> `updateDateTime` `string<date-time>`

▼ ● 400 Bad request

▼ `ErrorResponse`

> `errors` `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referrals \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "memberReferralId": 147258369,  
    "referralType": "Email",  
    "referralValue": "friend.email.2@gmail.com",  
    "referralDate": "2023-10-28",  
    "referredAccountNumber": "78508238",  
    "active": true,  
    "referredEarnedDate": "2023-11-15",  
    "updateDateTime": ""  
  }  
]
```



14.27.2. Create a new referral for a member

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referrals`

Try it ▶

Creates a referral where both referrer and referee earn rewards

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-referral:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Body Parameters

▾ body CreateMemberReferralRequest *

> referralType string

> referralValue string

> active boolean

☐

Responses

▾ ● 200 Success

▾ MemberReferralResponse

> memberReferralId integer<int64>

> referralType string

> referralValue string

> referralDate string<date>

> active boolean

> updateDateTime string<date-time>

▾ ● 400 Bad Request

▾ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal Error

Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referrals \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "active": true  
  }'
```

Responses

● 200

```
// Success  
{  
  "memberReferralId": 4,  
  "referralType": "Email",  
  "referralValue": "friend.email.3@gmail.com",  
  "referralDate": "2023-06-18",  
  "active": false,  
  "updateDateTime": ""  
}
```




14.27.3. Find Referral Codes

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referral-codes`

Try it ▶

Find referral codes by member ID

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-referral-codes:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Responses

▼ ● 200 Success

▼ MemberReferralCode[]

> memberReferralCodeId integer<int64>

> referralCode string

> active boolean

> updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referral-codes \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "memberReferralCodeId": 147258369,
    "referralCode": "F9DA265FD56D496C",
    "active": false,
    "updateDateTime": ""
  }
]
```



14.27.4. Find referral code details

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/referral-codes/{referralCode}`

Try it ▶

Find referral code details by referral code

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-referral-codes:read

Authorize

Use Your Own Token

Path Parameters

> referralCode string *

Responses

▼ ● 200 Success

▼ MemberReferralCode

> memberReferralCodeId integer<int64>

> referralCode string

> active boolean

> updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/referral-codes/{referralCode} \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "memberReferralCodeId": 147258369,
  "referralCode": "F9DA265FD56D496C",
  "active": false,
  "updateDateTime": ""
}
```



14.27.5. Finds Referrer

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referred-by`

Try it ▶

Find the referrer for an enrolled referee

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 2 ▼

Required Scopes: api:member-referral:read, api:member:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Responses

▼ ● 200 Success

▼ ReferredByResponse

> accountNumber string

> memberIdentifier string

> firstName string

> lastName string

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referred-by \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "accountNumber": "121279666",
  "memberIdentifier": "1000032",
  "firstName": "John",
  "lastName": "Smith"
}
```



14.28. background-task

14.28.1. Get Background task

GET

https://api.b.preview.loyaltyservices.io/api/v3/background-tasks/{backgroundTaskId}

Try it ▶

Get a background task

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:background-task:read

Authorize

Use Your Own Token

Path Parameters

> backgroundTaskId integer<int64> *

Format: int64

Responses

▼ ● 200 Success

▼ BackgroundTask

> status string

> commandResponse object

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/background-tasks/{backgroundTaskId} \  
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "status": "Processed",
  "commandResponse": {}
}
```




14.29. async-task



14.29.1. Find Async Tasks

GET

<https://api.b.preview.loyaltyservices.io/api/v3/async-tasks>

Try it ▶

Retrieve platform API and file task details

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:async-task:read

Authorize

Use Your Own Token

Query Parameters

> startDate string<date>

mm/dd/yyyy



> endDate string<date>

mm/dd/yyyy



> taskTypeCode string

Select...



> taskCode string

> statusCode string

Select...



> fileName string

> pageNumber integer

1

> pageSize integer

500

Responses

▼ ● 200 Success

▼ FindAsyncTasksResponse

> asyncTasks AsyncTask[]

> pageInfo PageInfo

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/async-tasks?pageNumber=1&pageSize=500' \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
{
  "asyncTasks": [
    {
      "asyncTaskId": "",
      "taskTypeCode": "PlatformAPI",
      "taskCode": "enrollMember",
      "requestDate": "",
      "requestData": {},
      "statusCode": "",
      "responseDate": "",
      "responseData": {},
      "fileName": "",
      "lineNumber": 0,
      "fileData": ""
    }
  ],
  "pageInfo": {
    "number": 0,
    "size": 0,
    "totalItems": 0,
    "hasNext": false
  }
}
```



14.29.2. Get Async Task

GET

`https://api.b.preview.loyaltyservices.io/api/v3/async-tasks/{asyncTaskId}`

Try it ▶

Get async task details

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:async-task:read

Authorize

Use Your Own Token

Path Parameters

> `asyncTaskId` `integer<int64>` *

Format: int64

Responses

▼ ● 200 Success

▼ AsyncTask

> `asyncTaskId` `integer<int64>`

> `taskTypeCode` `string`

> `taskCode` `string`

> `requestDate` `string<date-time>`

> `requestData` `object`

> `statusCode` `string`

> `responseDate` `string<date-time>`

> `responseData` `object`

> `fileName` `string`

> `lineNumber` `integer`

> `fileData` `string`

▼ ● 400 Bad request

▼ ErrorResponse

> `errors` `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/async-tasks/{asyncTaskId} \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
{
  "asyncTaskId": "",
  "taskTypeCode": "PlatformAPI",
  "taskCode": "enrollMember",
  "requestDate": "",
  "requestData": {},
  "statusCode": "",
  "responseDate": "",
  "responseData": {},
  "fileName": "",
  "lineNumber": 0,
  "fileData": ""
}
```



14.29.3. Retry Async Task

POST`https://api.b.preview.loyaltyservices.io/api/v3/async-tasks/{asyncTaskId}/retry`

Try it

Creates a new async task that retries a failed or queued task. Only tasks with ERROR or QUEUED status can be retried.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:async-task:create

Authorize

Use Your Own Token

Path Parameters

> `asyncTaskId` `integer<int64>` *

Format: int64

Responses

✓ ● 202 Accepted - New async task created and queued for processing

▾ AsyncTask

```
> asyncTaskId integer<int64>
> taskTypeCode string
> taskCode string
> requestDate string<date-time>
> requestData object
> statusCode string
> responseDate string<date-time>
> responseData object
> fileName string
> lineNumber integer
> fileData string
```

✓ ● 400 Bad request - Task has invalid status (must be ERROR or QUEUED)

▾ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized



● 403 Forbidden - Missing api:async-task

scope

▼ ● 404 Task not found

▼ ErrorResponse

> errors ValidationError[]

● 500 Internal error

Request

Curl

```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/async-tasks/{asyncTaskId}/retry \  
  --header 'accept: application/json'
```

Responses

● 202

```
// Accepted - New async task created and queued for processing  
{  
  "asyncTaskId": "",  
  "taskTypeCode": "PlatformAPI",  
  "taskCode": "enrollMember",  
  "requestDate": "",  
  "requestData": {},  
  "statusCode": "",  
  "responseDate": "",  
  "responseData": {},  
  "fileName": "",  
  "lineNumber": 0,  
  "fileData": ""  
}
```



14.29.4. Find Async Task Batches

GET

`https://api.b.preview.loyaltyservices.io/api/v3/async-task-batches`

Try it ▶

Search for async task batches

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:async-task:read

Authorize

Use Your Own Token

Query Parameters

> fileName string

Responses

▼ ● 200 Success

▼ AsyncTaskBatch[]

asyncTaskBatchId integer

> taskTypeCode string

> taskCode string

> schema string

> fileName string

> lineCount integer

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/async-task-batches \
  --header 'accept: application/json'
```




Responses

● 200

```
// Success
[
  {
    "asyncTaskBatchId": 0,
    "taskTypeCode": "PlatformAPI",
    "taskCode": "enrollMember",
    "schema": "DefaultSchema",
    "fileName": "",
    "lineCount": 0
  }
]
```



14.29.5. Submit multiple async operations

POST<https://api.b.preview.loyaltyservices.io/api/v3/async-task-batches>

Try it ▶

Submit a batch of operations in CSV format, with one operation per line, for asynchronous execution.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:async-task:create

Authorize

Use Your Own Token

Query Parameters

> typeCode string *

Form Parameters

> file string<binary>

Choose file

Responses

▼ ● 200 Success

▼ AsyncTaskBatch

asyncTaskBatchId integer

> taskTypeCode string

> taskCode string

> schema string

> fileName string

> lineCount integer

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/async-task-batches?typeCode=postBonusTransactionDefaultSchema' \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
{
  "asyncTaskBatchId": 0,
  "taskTypeCode": "PlatformAPI",
  "taskCode": "enrollMember",
  "schema": "DefaultSchema",
  "fileName": "",
  "lineCount": 0
}
```



14.29.6. Get Async Task Batch

GET

`https://api.b.preview.loyaltyservices.io/api/v3/async-task-batches/{asyncTaskBatchId}`

Try it

Get async task batch details

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:async-task:read

Authorize

Use Your Own Token

Path Parameters

> **asyncTaskBatchId** `integer<int64>` *

Format: int64

Responses

✓ ● 200 Success

▾ AsyncTaskBatch

asyncTaskBatchId `integer`

> **taskTypeCode** `string`

> **taskCode** `string`

> **schema** `string`

> **fileName** `string`

> **lineCount** `integer`

▾ ● 400 Bad request

▾ ErrorResponse

> **errors** `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/async-task-batches/{asyncTaskBatchId} \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "asyncTaskBatchId": 0,
  "taskTypeCode": "PlatformAPI",
  "taskCode": "enrollMember",
  "schema": "DefaultSchema",
  "fileName": "",
  "lineCount": 0
}
```



14.29.7. Find Async Task Batch Types

GET`https://api.b.preview.loyaltyservices.io/api/v3/async-task-batch-types`

Try it ▶

Return details of async task batch types i.e. file uploads the user is entitled to perform.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:async-task:read

Authorize

Use Your Own Token

Responses

✓ ● 200 Success

▾ AsyncTaskBatchType[]

- > typeCode string
- > taskTypeCode string
- > taskCode string
- > schema string
- > header string
- > sampleData string[]

▾ ● 400 Bad request

▾ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/async-task-batch-types \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "typeCode": "postBonusTransactionDefaultSchema",
    "taskTypeCode": "PlatformAPI",
    "taskCode": "enrollMember",
    "schema": "DefaultSchema",
    "header": "",
    "sampleData": [
      ""
    ]
  }
]
```



14.29.8. Get Async Task Batch Type

GET

`https://api.b.preview.loyaltyservices.io/api/v3/async-task-batch-types/{typeCode}`

Try it ▶

Get async task batch type details

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:async-task:read

Authorize

Use Your Own Token

Path Parameters

> typeCode string *

Responses

▼ ● 200 Success

▼ AsyncTaskBatchType

> typeCode string

> taskTypeCode string

> taskCode string

> schema string

> header string

> sampleData string[]

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/async-task-batch-types/{typeCode} \
  --header 'accept: application/json'
```




Responses

● 200

```
// Success
{
  "typeCode": "postBonusTransactionDefaultSchema",
  "taskTypeCode": "PlatformAPI",
  "taskCode": "enrollMember",
  "schema": "DefaultSchema",
  "header": "",
  "sampleData": [
    ""
  ]
}
```



14.30. system-broadcast-message

14.30.1. Create New Broadcast Message

POST

https://api.b.preview.loyaltyservices.io/api/v3/broadcast-messages

Try it ▶

Create New Broadcast Message

Credentials

OAuth 2.0

OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow:

Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:broadcast-message:modify

Authorize

Use Your Own Token

Body Parameters

▼ body BroadcastMessage

Description

Broadcast Message

> broadcastMessageId integer<int64>

Format: int64

> message string

> title string

> type string *

general

> startDate string<date> *

mm/dd/yyyy

📅

> endDate string<date> *

mm/dd/yyyy

📅

> status string *

active

> rawExpression string

Responses

▼ ● 200 Success



▼ BroadcastMessage

- > broadcastMessageId integer<int64>
- > message string
- > title string
- > type string *
- > startDate string<date> *
- > endDate string<date> *
- > status string *
- > rawExpression string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/broadcast-messages \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "type": "general",  
    "startDate": "2024-03-20",  
    "endDate": "2024-03-20",  
    "status": "active"  
  }'
```

Responses



```
// Success  
{  
  "broadcastMessageId": 97432,  
  "message": "",  
  "title": "",  
  "type": "",  
  "startDate": "2024-03-20",  
  "endDate": "2024-03-20",  
  "status": "",  
  "rawExpression": ""  
}
```





14.30.2. Find Broadcast Messages

GET

https://api.b.preview.loyaltyservices.io/api/v3/broadcast-messages

Try it ▶

Find broadcast messages

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:broadcast-messages:read

Authorize

Use Your Own Token

Query Parameters

> startDate string<date>

> endDate string<date>

> status string

> text string

> dashboard boolean

> viewAll boolean

mm/dd/yyyy

mm/dd/yyyy

Responses

▼ ● 200 Success

▼ BroadcastMessage[]

> broadcastMessageId integer<int64>

> message string

> title string

> type string *

> startDate string<date> *

> endDate string<date> *

> status string *

> rawExpression string

▼ ● 400 Bad request



▼ ErrorResponse
 > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

Curl

```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/broadcast-messages?dashboard=boolean&viewAll=boolean' \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "broadcastMessageId": 97432,  
    "message": "",  
    "title": "",  
    "type": "",  
    "startDate": "2024-03-20",  
    "endDate": "2024-03-20",  
    "status": "",  
    "rawExpression": ""  
  }  
]
```



14.30.3. Update Existing Broadcast Message

PUT

<https://api.b.preview.loyaltyservices.io/api/v3/broadcast-messages/{broadcastMessageId}>

Try it ▶

Update Existing Broadcast Message

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:broadcast-message:modify

Authorize

Use Your Own Token

Path Parameters

> broadcastMessageId integer<int64> *

Format: int64

Body Parameters

▼ body BroadcastMessage

Description

Broadcast Message

> broadcastMessageId integer<int64> *

Format: int64

> message string

> title string

> type string *

general

> startDate string<date> *

mm/dd/yyyy 📅

> endDate string<date> *

mm/dd/yyyy 📅

> status string *

active

> rawExpression string

Responses

▼ ● 200 Success



▼ BroadcastMessage

- > **broadcastMessageId** integer<int64>
- > **message** string
- > **title** string
- > **type** string *
- > **startDate** string<date> *
- > **endDate** string<date> *
- > **status** string *
- > **rawExpression** string

▼ ● 400 Bad request

▼ ErrorResponse

- > **errors** ValidationError[]

- 401 Unauthorized

- 403 Forbidden

- 500 Internal error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/broadcast-messages/{broadcastMessageId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "type": "general",
    "startDate": "2024-03-20",
    "endDate": "2024-03-20",
    "status": "active"
  }'
```

Responses

● 200

```
// Success
{
  "broadcastMessageId": 97432,
  "message": "",
  "title": "",
  "type": "",
  "startDate": "2024-03-20",
  "endDate": "2024-03-20",
  "status": "",
  "rawExpression": ""
}
```






14.30.4. Get Broadcast Message

GET

`https://api.b.preview.loyaltyservices.io/api/v3/broadcast-messages/{id}`

Try it

Get a broadcast message by id

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:broadcast-messages:read

Authorize

Use Your Own Token

Path Parameters

> id integer<int64> *

Format: int64

Responses

✓ 200 Success

BroadcastMessage

> broadcastMessageId integer<int64>

> message string

> title string

> type string *

> startDate string<date> *

> endDate string<date> *

> status string *

> rawExpression string

✓ 400 Bad request

ErrorResponse

> errors ValidationError[]

• 401 Unauthorized

• 403 Forbidden

• 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/broadcast-messages/{id} \  
  --header 'accept: application/json'
```

Responses

200

```
// Success  
{  
  "broadcastMessageId": 97432,  
  "message": "",  
  "title": "",  
  "type": "",  
  "startDate": "2024-03-20",  
  "endDate": "2024-03-20",  
  "status": "",  
  "rawExpression": ""  
}
```



14.31. member-linked-account

14.31.1. Links individual member account to an another member account

POST

https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/linked-accounts

Try it ▶

Create a Member linked account

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:member-linked-account:modify

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Body Parameters

▼ body MemberLinkedAccountRequest *

> linkedMemberIdentifier string *

> startDateTime string

> endDateTime string

> linkedType string *

> administrator boolean *

Responses

▼ ● 200 Success



▼ MemberLinkedAccountResponse

- > memberLinkedAccountId integer<int64>
- > linkedMemberIdentifier string
- > startDateTime string<date-time>
- > endDateTime string<date-time>
- > linkedType string
- > administrator boolean
- > result string
- > updateDateTime string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

[Curl](#)

```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/linked-accounts \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "linkedMemberIdentifier": 123456,  
    "linkedType": "CORP",  
    "administrator": true  
  }'
```

Responses

[200](#)

```
// Success  
{  
  "memberLinkedAccountId": "",  
  "linkedMemberIdentifier": "",  
  "startDateTime": "2025-06-11T00:00:00Z",  
  "endDateTime": "2025-06-21T00:00:00Z",  
  "linkedType": "CORP",  
  "administrator": true,  
  "result": "Added or Moved",  
  "updateDateTime": "2023-06-18T09:09:58.067Z"  
}
```





14.31.2. Retrieve member accounts linked to another member

GET

`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/linked-accounts`

Try it

Returns a list of member accounts associated with the specified member account, where the link is currently active (EndDate is null or in the future).

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes

Required Scopes: api:member-linked-account:read

Authorize

Use Your Own Token

Path Parameters

> memberIdentifier string *

Query Parameters

> linkedType string *

Responses

200 List of linked member accounts

LinkedMemberAccount[]

> memberLinkedAccountId integer<int64>

> memberIdentifier integer<int64>

accountNumber string

firstName string

lastName string

> accountStatusCode string

> startDateTime string<date-time>

> endDateTime string<date-time>

> administrator boolean

400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/linked-accounts?linkedType=CORP' \
  --header 'accept: application/json'
```

Responses

 200

```
// List of linked member accounts
[
  {
    "memberLinkedAccountId": "",
    "memberIdentifier": "",
    "accountNumber": "",
    "firstName": "",
    "lastName": "",
    "accountStatusCode": "",
    "startDateTime": "",
    "endDateTime": "",
    "administrator": false
  }
]
```




14.31.3. Unlink Member Linked account

PUT`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/linked-accounts/{linkedMemberIdentifier}`**Try it** ▶

Unlink individual member account from a linked account

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>Flow: **Client Credentials**

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:member-linked-account:modify

Authorize

Use Your Own Token

Path Parameters

> **memberIdentifier** string *> **linkedMemberIdentifier** string *

Query Parameters

> **linkedType** string *> **unlink** boolean> **admin** boolean

Responses

✓ ● 200 Success

▾ LinkedMemberAccount

> **memberLinkedAccountId** integer<int64>> **memberIdentifier** integer<int64>**accountNumber** string**firstName** string**lastName** string> **accountStatusCode** string> **startDateTime** string<date-time>> **endDateTime** string<date-time>> **administrator** boolean

▾ ● 400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request PUT \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/linked-accounts/{linkedMemberIdentifier}?link  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "memberLinkedAccountId": "",  
  "memberIdentifier": "",  
  "accountNumber": "",  
  "firstName": "",  
  "lastName": "",  
  "accountStatusCode": "",  
  "startDateTime": "",  
  "endDateTime": "",  
  "administrator": false  
}
```



14.32. event-rule

14.32.1. Create Event Rule

POST

Try it ▶

Create a new event rule.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: **Client Credentials**

Scopes 1 ▼

Required Scopes: api:event-rule:modify

Authorize

Use Your Own Token



Body Parameters

▼

body EventRule *

> eventRuleId integer<int64>

Format: int64

> eventRuleCode string

> eventRuleName string

> eventRuleDescription string

> eventType string

> startDate string<date>

mm/dd/yyyy 📅

> endDate string<date>

mm/dd/yyyy 📅

> status string

> dataRelease integer<int64>

Format: int64

> dataReleaseName string

> dataReleaseItemStatus string

> workflowStatus string

> rank integer

> rawExpression string

> results EventRuleResult[]

Responses

▼

● 200 Success



▼ EventRule

```
> eventRuleId integer<int64>
> eventRuleCode string
> eventRuleName string
> eventRuleDescription string
> eventType string
> startDate string<date>
> endDate string<date>
> status string
> dataRelease integer<int64>
> dataReleaseName string
> dataReleaseItemStatus string
> workflowStatus string
> rank integer
> rawExpression string
> results EventRuleResult[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal server error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/event-rules \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "results": [
      {
        "resultTypeCode": "Bonus",
        "parameters": "{\"bonusRuleId\":1234,\"bonusRuleCode\":\"C234\"}"
      }
    ]
  }'
```



Responses

● 200

```
// Success
{
  "eventRuleId": 97432,
  "eventRuleCode": "Event233",
  "eventRuleName": "TestEvent123",
  "eventRuleDescription": "",
  "eventType": "",
  "startDate": "2024-03-20",
  "endDate": "2024-03-20",
  "status": "",
  "dataRelease": "",
  "dataReleaseName": "",
  "dataReleaseItemStatus": "",
  "workflowStatus": "",
  "rank": 0,
  "rawExpression": "",
  "results": [
    {
      "resultTypeCode": "Bonus",
      "parameters": {}
    }
  ]
}
```



14.32.2. Find Event Rules

GET

https://api.b.preview.loyaltyservices.io/api/v3/event-rules

Try it ▶

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:event-rule:read

Authorize

Use Your Own Token

Query Parameters

eventRuleCode string

eventRuleName string

> eventType string

> startDate string<date>

> endDate string<date>

> resultTypeCode string

> status string

dataRelease integer

> dataReleaseItemStatus string

mm/dd/yyyy 📅

mm/dd/yyyy 📅

Responses

▼ ● 200 Success



▼ EventRule[]

- > eventRuleId integer<int64>
- > eventRuleCode string
- > eventRuleName string
- > eventRuleDescription string
- > eventType string
- > startDate string<date>
- > endDate string<date>
- > status string
- > dataRelease integer<int64>
- > dataReleaseName string
- > dataReleaseItemStatus string
- > workflowStatus string
- > rank integer
- > rawExpression string
- > results EventRuleResult[]

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/event-rules \  
  --header 'accept: application/json'
```




Responses

● 200

```
// Success
[
  {
    "eventRuleId": 97432,
    "eventRuleCode": "Event233",
    "eventRuleName": "TestEvent123",
    "eventRuleDescription": "",
    "eventType": "",
    "startDate": "2024-03-20",
    "endDate": "2024-03-20",
    "status": "",
    "dataRelease": "",
    "dataReleaseName": "",
    "dataReleaseItemStatus": "",
    "workflowStatus": "",
    "rank": 0,
    "rawExpression": "",
    "results": [
      {
        "resultTypeCode": "Bonus",
        "parameters": {}
      }
    ]
  }
]
```



14.32.3. Update event rule

PUT

<https://api.b.preview.loyaltyservices.io/api/v3/event-rules/{eventRuleCode}>

Try it ▶

Update an event rule. The event rule must currently be in Working or Ready status. Updating an event rule in Approved, Published, or Live status requires the "api:event-rule

" scope.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:event-rule:modify

Authorize

Use Your Own Token

Path Parameters

> eventRuleCode string *



Body Parameters

▼

body EventRule *

> eventRuleId integer<int64>

Format: int64

> eventRuleCode string

> eventRuleName string

> eventRuleDescription string

> eventType string

> startDate string<date>

mm/dd/yyyy 📅

> endDate string<date>

mm/dd/yyyy 📅

> status string

> dataRelease integer<int64>

Format: int64

> dataReleaseName string

> dataReleaseItemStatus string

> workflowStatus string

> rank integer

> rawExpression string

> results EventRuleResult[]

Responses

▼

● 200 Success



▼ EventRule

```
> eventRuleId integer<int64>
> eventRuleCode string
> eventRuleName string
> eventRuleDescription string
> eventType string
> startDate string<date>
> endDate string<date>
> status string
> dataRelease integer<int64>
> dataReleaseName string
> dataReleaseItemStatus string
> workflowStatus string
> rank integer
> rawExpression string
> results EventRuleResult[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal server error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/event-rules/{eventRuleCode} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "results": [
      {
        "resultTypeCode": "Bonus",
        "parameters": "{\"bonusRuleId\":1234,\"bonusRuleCode\":\"C234\"}"
      }
    ]
  }'
```



Responses

● 200

```
// Success
{
  "eventRuleId": 97432,
  "eventRuleCode": "Event233",
  "eventRuleName": "TestEvent123",
  "eventRuleDescription": "",
  "eventType": "",
  "startDate": "2024-03-20",
  "endDate": "2024-03-20",
  "status": "",
  "dataRelease": "",
  "dataReleaseName": "",
  "dataReleaseItemStatus": "",
  "workflowStatus": "",
  "rank": 0,
  "rawExpression": "",
  "results": [
    {
      "resultTypeCode": "Bonus",
      "parameters": {}
    }
  ]
}
```



14.32.4. Patch event rule

PATCH`https://api.b.preview.loyaltyservices.io/api/v3/event-rules/{eventRuleCode}`**Try it** ▶

Performs a partial update on an event rule. Only the name, description, status, and end date fields can be modified; all other fields remain unchanged. The event rule must currently be in Approved, Published, or Live status.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:event-rule:modify

Authorize

Use Your Own Token

Path Parameters

> **eventRuleCode** string *



Body Parameters

▼

body EventRule *

> eventRuleId integer<int64>

Format: int64

> eventRuleCode string

> eventRuleName string

> eventRuleDescription string

> eventType string

> startDate string<date>

mm/dd/yyyy 📅

> endDate string<date>

mm/dd/yyyy 📅

> status string

> dataRelease integer<int64>

Format: int64

> dataReleaseName string

> dataReleaseItemStatus string

> workflowStatus string

> rank integer

> rawExpression string

> results EventRuleResult[]

Responses

▼

● 200 Success



▼ EventRule

```
> eventRuleId integer<int64>
> eventRuleCode string
> eventRuleName string
> eventRuleDescription string
> eventType string
> startDate string<date>
> endDate string<date>
> status string
> dataRelease integer<int64>
> dataReleaseName string
> dataReleaseItemStatus string
> workflowStatus string
> rank integer
> rawExpression string
> results EventRuleResult[]
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal server error

Request



```
curl --request PATCH \
  --url https://api.b.preview.loyaltyservices.io/api/v3/event-rules/{eventRuleCode} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "results": [
      {
        "resultTypeCode": "Bonus",
        "parameters": "{\"bonusRuleId\":1234,\"bonusRuleCode\":\"C234\"}"
      }
    ]
  }'
```




Responses

● 200

```
// Success
{
  "eventRuleId": 97432,
  "eventRuleCode": "Event233",
  "eventRuleName": "TestEvent123",
  "eventRuleDescription": "",
  "eventType": "",
  "startDate": "2024-03-20",
  "endDate": "2024-03-20",
  "status": "",
  "dataRelease": "",
  "dataReleaseName": "",
  "dataReleaseItemStatus": "",
  "workflowStatus": "",
  "rank": 0,
  "rawExpression": "",
  "results": [
    {
      "resultTypeCode": "Bonus",
      "parameters": {}
    }
  ]
}
```



14.32.5. Retrieve event rule

GET

`https://api.b.preview.loyaltyservices.io/api/v3/event-rules/{eventRuleCode}`

Try it ▶

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:event-rule:read

Authorize

Use Your Own Token

Path Parameters

> eventRuleCode string *

Responses

▼ ● 200 Success

▼ EventRule

```
> eventRuleId integer<int64>
> eventRuleCode string
> eventRuleName string
> eventRuleDescription string
> eventType string
> startDate string<date>
> endDate string<date>
> status string
> dataRelease integer<int64>
> dataReleaseName string
> dataReleaseItemStatus string
> workflowStatus string
> rank integer
> rawExpression string
> results EventRuleResult[]
```

▼ ● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/event-rules/{eventRuleCode} \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
{
  "eventRuleId": 97432,
  "eventRuleCode": "Event233",
  "eventRuleName": "TestEvent123",
  "eventRuleDescription": "",
  "eventType": "",
  "startDate": "2024-03-20",
  "endDate": "2024-03-20",
  "status": "",
  "dataRelease": "",
  "dataReleaseName": "",
  "dataReleaseItemStatus": "",
  "workflowStatus": "",
  "rank": 0,
  "rawExpression": "",
  "results": [
    {
      "resultTypeCode": "Bonus",
      "parameters": {}
    }
  ]
}
```



14.32.6. Delete event rule

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/event-rules/{eventRuleCode}`

Try it ▶

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:event-rule:modify

Authorize

Use Your Own Token

Path Parameters

> `eventRuleCode` string *

Responses

● 200 Success

▼ ● 400 Bad request

▼ ErrorResponse

> `errors` ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/event-rules/{eventRuleCode} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.33. audit

14.33.1. Find Audit Logs

GET

<https://api.b.preview.loyaltyservices.io/api/v3/audit>

Try it ▶

Find audit trail for one or more tables. Filter by entityId using OID (managed entities) or numeric ID.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:audit:read

Authorize

Use Your Own Token

Query Parameters

> **tableName** string[] *> **entityId** string> **pageNumber** integer> **pageSize** integer

+

Responses

▼ ● 200 Success

▼ FindAuditLogsResponse

> **auditLogs** AuditLogEntry[]> **pageInfo** PageInfo

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/audit?tableName=array&pageNumber=1&pageSize=100' \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "auditLogs": [  
    {  
      "tableName": "",  
      "entityId": "",  
      "entityKey": "",  
      "columnName": "",  
      "oldValue": "",  
      "newValue": "",  
      "changeType": "",  
      "user": "",  
      "updateDateTime": ""  
    }  
  ],  
  "pageInfo": {  
    "number": 0,  
    "size": 0,  
    "totalItems": 0,  
    "hasNext": false  
  }  
}
```



14.34. audit-configuration

14.34.1. List audit configurations

GET

<https://api.b.preview.loyaltyservices.io/api/v3/audit/configurations>

Try it ▶

Retrieves all audit configurations with optional filtering by schema, table, and active status

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:audit-configuration:read

Authorize

Use Your Own Token

Query Parameters

- > **schemaName** `string`
- > **tableName** `string`
- > **includeInactive** `boolean`

☐

Responses

▼ ● 200 Success

- ▼ AuditConfiguration[]
- > **auditConfigurationId** `integer<int64>`
 - > **schemaName** `string`
 - > **tableName** `string` *
 - > **columnName** `string` *
 - > **tableDisplayName** `string`
 - > **columnDisplayName** `string`
 - > **isActive** `boolean`
 - > **createDate** `string<date-time>`
 - > **updateDate** `string<date-time>`

▼ ● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/audit/configurations?includeInactive=boolean' \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "auditConfigurationId": "",  
    "schemaName": "",  
    "tableName": "",  
    "columnName": "",  
    "tableDisplayName": "",  
    "columnDisplayName": "",  
    "isActive": false,  
    "createDate": "",  
    "updateDate": ""  
  }  
]
```




14.34.2. Create audit configuration

POST`https://api.b.preview.loyaltyservices.io/api/v3/audit/configurations`

Try it

Creates a new audit configuration for field-level auditing

Credentials

OAuth 2.0

See <https://docs.wearephaedon.com/docs/platform-api-authentication>Flow: **Client Credentials**

Client ID

Client Secret

Scopes **1** Required Scopes: `api:audit-configuration:modify`**Authorize**

Use Your Own Token

Query Parameters

> **async** `boolean`

Body Parameters

body `AuditConfiguration` *> **schemaName** `string`> **tableName** `string` *> **columnName** `string` *> **tableDisplayName** `string`> **columnDisplayName** `string`> **isActive** `boolean`

dbo

Responses

201 Created



▼ AuditConfiguration

- > auditConfigurationId integer<int64>
- > schemaName string
- > tableName string *
- > columnName string *
- > tableDisplayName string
- > columnDisplayName string
- > isActive boolean
- > createDate string<date-time>
- > updateDate string<date-time>

▼ ● 202 Accepted

▼ AsyncTask

- > asyncTaskId integer<int64>
- > taskTypeCode string
- > taskCode string
- > requestDate string<date-time>
- > requestData object
- > statusCode string
- > responseDate string<date-time>
- > responseData object
- > fileName string
- > lineNumber integer
- > fileData string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

▼ ● 409 Conflict - configuration already exists



▼ ErrorResponse

> errors ValidationError[]

● 500 Internal error

Request

Curl

```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/audit/configurations?async=boolean' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "schemaName": "dbo",
    "tableName": "",
    "columnName": "",
    "isActive": true
  }'
```

Responses

● 201

```
// Created
{
  "auditConfigurationId": "",
  "schemaName": "",
  "tableName": "",
  "columnName": "",
  "tableDisplayName": "",
  "columnDisplayName": "",
  "isActive": false,
  "createDate": "",
  "updateDate": ""
}
```



14.34.3. Patch audit configuration

PATCH`https://api.b.preview.loyaltyservices.io/api/v3/audit/configurations/{schemaName}/{tableName}/{columnName}`**Try it** ▶

Perform a partial update (PATCH) on an audit configuration. Only `isActive`, `tableDisplayName`, and `columnDisplayName` fields can be modified. Fields not provided in the request will remain unchanged. To clear a field, provide an empty string.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:audit-configuration:modify`

Authorize

Use Your Own Token

Path Parameters

> `schemaName` string *> `tableName` string *> `columnName` string *

Body Parameters

▼ `body` AuditConfiguration *> `schemaName` string> `tableName` string *> `columnName` string *> `tableDisplayName` string> `columnDisplayName` string> `isActive` boolean

dbo

Responses

▼ ● 200 Success



▼ AuditConfiguration

- > auditConfigurationId integer<int64>
- > schemaName string
- > tableName string *
- > columnName string *
- > tableDisplayName string
- > columnDisplayName string
- > isActive boolean
- > createDate string<date-time>
- > updateDate string<date-time>

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

▼ ● 404 Not found

▼ ErrorResponse

- > errors ValidationError[]

● 500 Internal error

Request



```
curl --request PATCH \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/audit/configurations/{schemaName}/{tableName}/{columnName} \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "schemaName": "dbo",  
    "tableName": "",  
    "columnName": "",  
    "isActive": true  
  }'
```



Responses

● 200

```
// Success
{
  "auditConfigurationId": "",
  "schemaName": "",
  "tableName": "",
  "columnName": "",
  "tableDisplayName": "",
  "columnDisplayName": "",
  "isActive": false,
  "createDate": "",
  "updateDate": ""
}
```



14.34.4. Delete Audit Configuration

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/audit/configurations/{auditConfigurationId}`

Try it

Delete an audit configuration

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:audit-configuration:modify

Authorize

Use Your Own Token

Path Parameters

> **auditConfigurationId** `integer<int64>` *

Format: int64

Responses

● 200 Success

▾ ● 400 Bad request

▾ ErrorResponse

> **errors** `ValidationError[]`

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/audit/configurations/{auditConfigurationId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.35. support

14.35.1. Submit Contact Us Inquiry

POST

https://api.b.preview.loyaltyservices.io/api/v3/support/contact-us

Try it ▶

Submit a contact form inquiry to the client operations team

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:contact-us:modify

Authorize

Use Your Own Token

Body Parameters

body ContactUsRequest *

> details string *

> phoneNumber string

Responses

- 200 Success
- ▼ ● 400 Bad request
 - ▼ ErrorResponse
 - > errors ValidationError[]
- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

Curl

```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/support/contact-us \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "details": ""
  }'
```




Responses

● 200

// Success



14.36. data-release

14.36.1. Search Data Releases

GET

<https://api.b.preview.loyaltyservices.io/api/v3/data-releases>

Try it ▶

Search for data releases by status, text, and publish date range

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:data-release:read

Authorize

Use Your Own Token

Query Parameters

> text string

> statusCode string

> publishStartDate string<date>

> publishEndDate string<date>

> includeItemSummary boolean

> includeItems boolean

> includeStatusHistories boolean

mm/dd/yyyy 📅

mm/dd/yyyy 📅

Responses

▼ ● 200 Success



▼ DataRelease[]

Description

A data release record, optionally including item summary, items, and status history

- > **id** integer<int64>
- > **name** string
- > **description** string
- > **statusCode** string
- > **publishDate** string
- > **approvalsNeeded** boolean
- > **approvers** string[]
- > **cancelable** boolean
- > **editable** boolean
- > **deletable** boolean
- > **createDate** string<date-time>
- > **createUserName** string
- > **updateDate** string<date-time>
- > **updateUserName** string
- > **itemSummary** DataReleaseItemSummary[]
- > **items** DataReleaseItem[]
- > **statusHistories** DataReleaseHistory[]

▼ ● 400 Bad request

▼ ErrorResponse

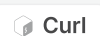
- > **errors** ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/data-releases \  
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "id": "",
    "name": "",
    "description": "",
    "statusCode": "",
    "publishDate": "",
    "approvalsNeeded": false,
    "approvers": [
      ""
    ],
    "cancelable": false,
    "editable": false,
    "deletable": false,
    "createDate": "",
    "createUserName": "",
    "updateDate": "",
    "updateUserName": "",
    "itemSummary": [
      {
        "typeCode": "",
        "count": 0
      }
    ],
    "items": [
      {
        "id": "",
        "itemId": "",
        "typeCode": "",
        "statusCode": "",
        "name": "",
        "workflowStatusCode": "",
        "startDate": "",
        "endDate": "",
        "dataReleaseItemStatus": "",
        "isEditable": false,
        "isReapprovable": false,
        "isBumpable": false,
        "isUsed": false
      }
    ],
    "statusHistories": [
      {
        "prevStatusCode": "",
        "statusCode": "",
        "comments": "",
        "createDate": "",
        "createUserName": "",
        "updateDate": "",
        "updateUserName": ""
      }
    ]
  }
]
```





14.37. segment

14.37.1. Find segments

GET

Try it ▶

Retrieve available segments

Credentials

OAuth 2.0

 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow:

Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:segment:read

Authorize

Use Your Own Token

Query Parameters

> segmentName string

> segmentCode string

> excludeTypeCode string[]

> statusCode string

> startDate string<date>

> endDate string<date>

> usedInType string

> usedInName string

+

mm/dd/yyyy

📅

mm/dd/yyyy

📅

Responses

▼ ● 200 Success



▼ Segment[]

Description

Defines a segment (list of members)

- > **segmentId** integer<int64>
- > **code** string
- > **name** string
- > **description** string
- > **statusCode** string
- > **typeCode** string
- > **expirationDays** integer
- > **ownerName** string
- > **displayInCC** boolean

▼ ● 400 Bad request

▼ ErrorResponse

- > **errors** ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

Curl

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/segments \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "segmentId": "",  
    "code": "",  
    "name": "",  
    "description": "",  
    "statusCode": "",  
    "typeCode": "",  
    "expirationDays": 0,  
    "ownerName": "",  
    "displayInCC": false  
  }  
]
```





14.37.2. Create segment

POST

<https://api.b.preview.loyaltyservices.io/api/v3/segments>

Try it ▶

Creates a segment, which is a collection of members. This is the first step in defining a segment.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:segment:modify

Authorize

Use Your Own Token

Body Parameters

▼ body Segment *

Description

Defines a segment (list of members)

> code string

> name string

> description string

> statusCode string

> typeCode string

> expirationDays integer

> ownerName string

> displayInCC boolean

Responses

▼ ● 200 Success



▼ Segment

Description

Defines a segment (list of members)

- > **segmentId** `integer<int64>`
- > **code** `string`
- > **name** `string`
- > **description** `string`
- > **statusCode** `string`
- > **typeCode** `string`
- > **expirationDays** `integer`
- > **ownerName** `string`
- > **displayInCC** `boolean`

▼ ● 400 Bad request

▼ ErrorResponse

- > **errors** `ValidationError[]`

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/segments \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "segmentId": "",  
  "code": "",  
  "name": "",  
  "description": "",  
  "statusCode": "",  
  "typeCode": "",  
  "expirationDays": 0,  
  "ownerName": "",  
  "displayInCC": false  
}
```



14.37.3. Get segment

GET

`https://api.b.preview.loyaltyservices.io/api/v3/segments/{segmentCode}`

Try it ▶

Retrieve a segment by its code

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:segment:read

Authorize

Use Your Own Token

Path Parameters

> `segmentCode` `string` *

Responses

▼ ● 200 Success

▼ Segment

Description

Defines a segment (list of members)

```
> segmentId integer<int64>
> code string
> name string
> description string
> statusCode string
> typeCode string
> expirationDays integer
> ownerName string
> displayInCC boolean
```

▼ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 404 Not found



● 500 Internal error

Request

[Curl](#)

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/segments/{segmentCode} \
  --header 'accept: application/json'
```

Responses

[200](#)

```
// Success
{
  "segmentId": "",
  "code": "",
  "name": "",
  "description": "",
  "statusCode": "",
  "typeCode": "",
  "expirationDays": 0,
  "ownerName": "",
  "displayInCC": false
}
```



14.37.4. Update segment

PUT

`https://api.b.preview.loyaltyservices.io/api/v3/segments/{segmentCode}`

Try it ▶

Updates an existing segment by its code. This endpoint modifies segment properties such as name and description.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:segment:modify

Authorize

Use Your Own Token

Path Parameters

> `segmentCode string *`

Body Parameters

▼ `body Segment *`

Description

Defines a segment (list of members)

> `code string`> `name string`> `description string`> `statusCode string`> `typeCode string`> `expirationDays integer`> `ownerName string`> `displayInCC boolean`

Responses

▼ ● 200 Success



Segment

Description

Defines a segment (list of members)

- > **segmentId** `integer<int64>`
- > **code** `string`
- > **name** `string`
- > **description** `string`
- > **statusCode** `string`
- > **typeCode** `string`
- > **expirationDays** `integer`
- > **ownerName** `string`
- > **displayInCC** `boolean`

● 400 Bad request

ErrorResponse

- > **errors** `ValidationError[]`

- 401 Unauthorized
- 403 Forbidden
- 404 Not found
- 500 Internal error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/segments/{segmentCode} \
  --header 'accept: application/json'
```

Responses



```
// Success
{
  "segmentId": "",
  "code": "",
  "name": "",
  "description": "",
  "statusCode": "",
  "typeCode": "",
  "expirationDays": 0,
  "ownerName": "",
  "displayInCC": false
}
```





14.37.5. Delete segment

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/segments/{segmentCode}`

Try it ▶

Delete an existing segment by its code

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:segment:modify

Authorize

Use Your Own Token

Path Parameters

> `segmentCode string *`

Responses

● 200 Success

▾ ● 400 Bad request

▾ ErrorResponse

> `errors ValidationError[]`

● 401 Unauthorized

● 403 Forbidden

● 404 Not found

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/segments/{segmentCode} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.38. communication-type

14.38.1. Find communication types

GET

<https://api.b.preview.loyaltyservices.io/api/v3/communication-types>

Try it ▶

Retrieve available communication types. A communication type can be created for each email/SMS campaign in your ESP that accepts triggered events.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:communication-type:read

Authorize

Use Your Own Token

Query Parameters

- > **name** *string*
- > **channel** *string*
- > **statusCode** *string*
- > **pageNumber** *integer*
- > **pageSize** *integer*

 ▼

Responses

- ▼ ● 200 Success
 - ▼ FindCommunicationTypesResponse
 - > **communicationTypes** *CommunicationType[]* *
 - > **pageInfo** *PageInfo* *
- ▼ ● 400 Bad request
 - ▼ ErrorResponse
 - > **errors** *ValidationError[]*
- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/communication-types?pageNumber=1&pageSize=50' \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "communicationTypes": [  
    {  
      "code": "WELCOME_EMAIL",  
      "name": "Welcome Email",  
      "description": "Sent when a member enrolls",  
      "channel": "Email",  
      "providerCode": "Braze",  
      "externalCode": "d-abc123",  
      "statusCode": "Active"  
    }  
  ],  
  "pageInfo": {  
    "number": 0,  
    "size": 0,  
    "totalItems": 0,  
    "hasNext": false  
  }  
}
```



14.38.2. Create communication type

POST

<https://api.b.preview.loyaltyservices.io/api/v3/communication-types>

Try it

Create a new communication type

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:communication-type:modify

Authorize

Use Your Own Token

Body Parameters

body CommunicationType *

- > code string *
- > name string *
- > description string
- > channel string<Email | SMS> *
- > providerCode string *
- > externalCode string *
- > statusCode string *

Select...

Responses

200 Success

CommunicationType

- > code string *
- > name string *
- > description string
- > channel string<Email | SMS> *
- > providerCode string *
- > externalCode string *
- > statusCode string *

400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/communication-types \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "code": "WELCOME_EMAIL",
    "name": "Welcome Email",
    "channel": "Email",
    "providerCode": "Braze",
    "externalCode": "d-abc123",
    "statusCode": "Active"
  }'
```

Responses

● 200

```
// Success
{
  "code": "WELCOME_EMAIL",
  "name": "Welcome Email",
  "description": "Sent when a member enrolls",
  "channel": "Email",
  "providerCode": "Braze",
  "externalCode": "d-abc123",
  "statusCode": "Active"
}
```



14.38.3. Get communication type

GET

`https://api.b.preview.loyaltyservices.io/api/v3/communication-types/{communicationCode}`

Try it

Retrieve a communication type by its code

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:communication-type:read

Authorize

Use Your Own Token

Path Parameters

> **communicationCode** string *

Responses

✓ ● 200 Success

CommunicationType

> **code** string *

> **name** string *

> **description** string

> **channel** string<Email | SMS> *

> **providerCode** string *

> **externalCode** string *

> **statusCode** string *

✓ ● 400 Bad request

ErrorResponse

> **errors** ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 404 Not found
- 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/communication-types/{communicationCode} \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "code": "WELCOME_EMAIL",  
  "name": "Welcome Email",  
  "description": "Sent when a member enrolls",  
  "channel": "Email",  
  "providerCode": "Braze",  
  "externalCode": "d-abc123",  
  "statusCode": "Active"  
}
```



14.38.4. Update communication type

PUT

<https://api.b.preview.loyaltyservices.io/api/v3/communication-types/{communicationCode}>

Try it ▶

Update a communication type. The channel (Email or SMS) cannot be changed.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:communication-type:modify

Authorize

Use Your Own Token

Path Parameters

> communicationCode string *

Body Parameters

▼ body CommunicationType *

> code string *

> name string *

> description string

> channel string<Email | SMS> *

> providerCode string *

> externalCode string *

> statusCode string *

Select... ▾

Responses

▼ ● 200 Success



CommunicationType

```
> code string *
> name string *
> description string
> channel string<Email | SMS> *
> providerCode string *
> externalCode string *
> statusCode string *
```

● 400 Bad request

ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 404 Not found
- 500 Internal error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/communication-types/{communicationCode} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "code": "WELCOME_EMAIL",
    "name": "Welcome Email",
    "channel": "Email",
    "providerCode": "Braze",
    "externalCode": "d-abc123",
    "statusCode": "Active"
  }'
```

Responses

● 200

```
// Success
{
  "code": "WELCOME_EMAIL",
  "name": "Welcome Email",
  "description": "Sent when a member enrolls",
  "channel": "Email",
  "providerCode": "Braze",
  "externalCode": "d-abc123",
  "statusCode": "Active"
}
```






14.39. points-pool

14.39.1. Find Points Pools

GET

<https://api.b.preview.loyaltyservices.io/api/v3/points-pools>

Try it ▶

Search for points pools.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:points-pool:read

Authorize

Use Your Own Token

Query Parameters

- > poolName string
- > poolStatusCode string
- > memberIdentifier string

Responses

✓ ● 200 Success

▾ PointsPool[]

- > poolId integer<int64>
- > name string
- > description string
- > statusCode string
- > members PointsPoolMember[]

▾ ● 400 Bad request

▾ ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/points-pools \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "poolId": "",  
    "name": "",  
    "description": "",  
    "statusCode": "",  
    "members": [  
      {  
        "memberIdentifier": "",  
        "statusCode": "",  
        "accountNumber": "",  
        "firstName": "",  
        "lastName": "",  
        "accountStatusCode": "",  
        "pointBalance": 12345  
      }  
    ]  
  }  
]
```



14.39.2. Create Points Pool

POST<https://api.b.preview.loyaltyservices.io/api/v3/points-pools>

Try it

Create a new points pool. At least two initial (active or invited) members are required. A member can be active in or invited to only one active points pool at a time.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:points-pool:modify

Authorize

Use Your Own Token

Body Parameters

▾ **body** `PointsPool` *

- > **name** `string`
- > **description** `string`
- > **members** `PointsPoolMember[]`

Responses

▾ ● **200** Success

▾ `PointsPool`

- > **poolId** `integer<int64>`
- > **name** `string`
- > **description** `string`
- > **statusCode** `string`
- > **members** `PointsPoolMember[]`

▾ ● **400** Bad request

▾ `ErrorResponse`

- > **errors** `ValidationError[]`

● **401** Unauthorized

● **403** Forbidden

● **500** Internal error



Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/points-pools \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "members": [
      {
        "memberIdentifier": "",
        "statusCode": "A"
      }
    ]
  }'
```

Responses

● 200

```
// Success
{
  "poolId": "",
  "name": "",
  "description": "",
  "statusCode": "",
  "members": [
    {
      "memberIdentifier": "",
      "statusCode": "",
      "accountNumber": "",
      "firstName": "",
      "lastName": "",
      "accountStatusCode": "",
      "pointBalance": 12345
    }
  ]
}
```



14.39.3. Get Points Pool

GET

`https://api.b.preview.loyaltyservices.io/api/v3/points-pools/{poolId}`

Try it ▶

Retrieve a points pool by its ID.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:points-pool:read

Authorize

Use Your Own Token

Path Parameters

> poolId integer<int64> *

Format: int64

Responses

▼ ● 200 Success

▼ PointsPool

> poolId integer<int64>

> name string

> description string

> statusCode string

> members PointsPoolMember[]

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/points-pools/{poolId} \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "poolId": "",
  "name": "",
  "description": "",
  "statusCode": "",
  "members": [
    {
      "memberIdentifier": "",
      "statusCode": "",
      "accountNumber": "",
      "firstName": "",
      "lastName": "",
      "accountStatusCode": "",
      "pointBalance": 12345
    }
  ]
}
```



14.39.4. Delete Points Pool

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/points-pools/{poolId}`

Try it

Disband a points pool.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:points-pool:modify

Authorize

Use Your Own Token

Path Parameters

> poolId integer<int64> *

Format: int64

Responses

● 200 Success

▾ ● 400 Bad request

▾ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request DELETE \
  --url https://api.b.preview.loyaltyservices.io/api/v3/points-pools/{poolId} \
  --header 'accept: application/json'
```

Responses

● 200

// Success



14.39.5. Add or Update Pool Member

PUT

https://api.b.preview.loyaltyservices.io/api/v3/points-pools/{poolId}/members/{memberIdentifier}

Try it ▶

Add a member to a points pool, or update their status if they are already in the pool. A member can be active in or invited to only one active points pool at a time. To remove a member from the pool, set their status to Retired. If fewer than two members remain in Active or Invited status, the pool is automatically disbanded.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:points-pool:modify

Authorize

Use Your Own Token

Path Parameters

> poolId integer<int64> *

Format: int64

> memberIdentifier string *

Body Parameters

body PointsPoolMember *

memberIdentifier string *

> statusCode string *

A

> pointBalance number

Responses

200 Success

PointsPool

> poolId integer<int64>

> name string

> description string

> statusCode string

> members PointsPoolMember[]

400 Bad request

ErrorResponse

> errors ValidationError[]



- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/points-pools/{poolId}/members/{memberIdentifier} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "memberIdentifier": "",
    "statusCode": "A"
  }'
```

Responses



```
// Success
{
  "poolId": "",
  "name": "",
  "description": "",
  "statusCode": "",
  "members": [
    {
      "memberIdentifier": "",
      "statusCode": "",
      "accountNumber": "",
      "firstName": "",
      "lastName": "",
      "accountStatusCode": "",
      "pointBalance": 12345
    }
  ]
}
```



14.40. customData

14.40.1. Get All Custom Data Metadata

GET

https://api.b.preview.loyaltyservices.io/api/v3/custom-data

Try it ▶

Gets metadata describing available custom data

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼ Required Scopes: api:custom-data:read

Authorize

Use Your Own Token

Responses

- ▼ ● 200 Success
- ▼ CustomDataMetadata[]
- > customDataId integer<int64> *
- > type string *
- > name string *
- > description string *
- > usageType string<Earnings | Redemptions | Both> *
- > metadata object
- > updateDateTime string
- ▼ ● 400 Bad request
- ▼ ErrorResponse
- > errors ValidationError[]
- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/custom-data \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "customDataId": 1,  
    "type": "Ace01",  
    "name": "Brand",  
    "description": "Brand",  
    "usageType": "",  
    "metadata": {},  
    "updateDateTime": "2026-06-02T00:00:00"  
  }  
]
```



14.40.2. Get Custom Data Metadata by ID

GET

`https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}`

Try it

Gets metadata describing a specific custom data type identified by ID.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:custom-data:read

Authorize

Use Your Own Token

Path Parameters

> **customDataId** `integer<int64>` *

Format: int64

Responses

▾ ● 200 Success

▾ CustomDataMetadata

```
> customDataId integer<int64> *
> type string *
> name string *
> description string *
> usageType string<Earnings | Redemptions | Both> *
> metadata object
> updateDateTime string
```

▾ ● 400 Bad request

▾ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId} \  
  --header 'accept: application/json'
```

Responses

200

```
// Success  
{  
  "customDataId": 1,  
  "type": "Ace01",  
  "name": "Brand",  
  "description": "Brand",  
  "usageType": "",  
  "metadata": {},  
  "updateDateTime": "2026-06-02T00:00:00"  
}
```



14.40.3. Gets Custom Data Detail Items

GET

`https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details`

Try it

Gets custom data detail items for a custom data type.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:custom-data:read

Authorize

Use Your Own Token

Path Parameters

> **customDataId** `integer<int64>` *

Format: int64

Query Parameters

> **includeInactive** `boolean`

Responses

✓ 200 Success

CustomDataDetail[]

```
> detailId integer<int64>
> externalKey string *
> description string *
> status string
> additionalProperties map<string, string>
> updateDateTime string
> createUserName string
> updateUserName string
```

✓ 400 Bad request

ErrorResponse

> **errors** `ValidationError[]`

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details?includeInactive=boolean' \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "detailId": 1,
    "externalKey": "ABC123",
    "description": "ABC 123",
    "status": "Active",
    "additionalProperties": "",
    "updateDateTime": "2026-06-02T00:00:00",
    "createUserName": "",
    "updateUserName": ""
  }
]
```




14.40.4. Create records under individual Ace Values

POST

`https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details`

Try it

This will be used to create records under individual Ace values.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:custom-data:modify

Authorize

Use Your Own Token

Path Parameters

> **customDataId** `integer<int64>` *

Format: int64

Query Parameters

> **async** `boolean`

Body Parameters

▼ **body** `CustomDataDetail` *

Description

Create Custom Data Details Request Object

> **detaiId** `integer<int64>`

Format: int64

> **externalKey** `string` *> **description** `string` *> **status** `string`> **additionalProperties** `map<string, string>`

Responses

▼ ● 200 Success



▼ CustomDataDetail[]

- > detailId integer<int64>
- > externalKey string *
- > description string *
- > status string
- > additionalProperties map<string, string>
- > updateDateTime string
- > createUserName string
- > updateUserName string

▼ ● 202 Accepted

▼ AsyncTask

- > asyncTaskId integer<int64>
- > taskTypeCode string
- > taskCode string
- > requestDate string<date-time>
- > requestData object
- > statusCode string
- > responseDate string<date-time>
- > responseData object
- > fileName string
- > lineNumber integer
- > fileData string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details?async=boolean' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "externalKey": "ABC123",
    "description": "ABC 123"
  }'
```

Responses



```
// Success
[
  {
    "detailId": 1,
    "externalKey": "ABC123",
    "description": "ABC 123",
    "status": "Active",
    "additionalProperties": "",
    "updateDateTime": "2026-06-02T00:00:00",
    "createUserName": "",
    "updateUserName": ""
  }
]
```



14.40.5. Get Custom Data Detail Item by ID

GET

`https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details/{customDataDetailId}`

Try it ▶

Gets a custom data detail item by ID

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:custom-data:read

Authorize

Use Your Own Token

Path Parameters

> **customDataId** `integer<int64>` *

Format: int64

> **customDataDetailId** `integer<int64>` *

Format: int64

Responses

✓ ● 200 Success

CustomDataDetail

```
> detailId integer<int64>
> externalKey string *
> description string *
> status string
> additionalProperties map<string, string>
> updateDateTime string
> createUserName string
> updateUserName string
```

✓ ● 400 Bad request

ErrorResponse

> errors `ValidationError[]`

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details/{customDataDetailId} \  
  --header 'accept: application/json'
```

Responses



```
// Success  
{  
  "detailId": 1,  
  "externalKey": "ABC123",  
  "description": "ABC 123",  
  "status": "Active",  
  "additionalProperties": "",  
  "updateDateTime": "2026-06-02T00:00:00",  
  "createUserName": "",  
  "updateUserName": ""  
}
```



14.40.6. Update records under individual Ace values

PUT`https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details/{customDataDetailId}`

Try it

This will be used to update records under individual Ace values

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:custom-data:modify

Authorize

Use Your Own Token

Path Parameters

> **customDataId** integer<int64> *

Format: int64

> **customDataDetailId** integer<int64> *

Format: int64

Body Parameters

body CustomDataDetail *

Description

Update Custom Data Details Request Object

> **detailId** integer<int64>

Format: int64

> **externalKey** string *> **description** string *> **status** string> **additionalProperties** map<string, string>

Responses

200 Success



CustomDataDetail

```
> detailId integer<int64>
> externalKey string *
> description string *
> status string
> additionalProperties map<string, string>
> updateDateTime string
> createUserName string
> updateUserName string
```

● 400 Bad request

ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details/{customDataDetailId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "externalKey": "ABC123",
    "description": "ABC 123"
  }'
```

Responses

● 200

```
// Success
{
  "detailId": 1,
  "externalKey": "ABC123",
  "description": "ABC 123",
  "status": "Active",
  "additionalProperties": "",
  "updateDateTime": "2026-06-02T00:00:00",
  "createUserName": "",
  "updateUserName": ""
}
```



14.40.7. Get Custom Data Detail Item by External Key

GET

`https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details/externalKey;{externalKey}`

Try it ▶

Gets a custom data detail item by external key.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:custom-data:read

Authorize

Use Your Own Token

Path Parameters

> **customDataId** `integer<int64>` *

Format: int64

> **externalKey** `string` *

Responses

✓ ● 200 Success

CustomDataDetail

```
> detailId integer<int64>
> externalKey string *
> description string *
> status string
> additionalProperties map<string, string>
> updateDateTime string
> createUserName string
> updateUserName string
```

✓ ● 400 Bad request

ErrorResponse

> errors `ValidationError[]`

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details/externalKey;{externalKey} \  
  --header 'accept: application/json'
```

Responses

200

```
// Success  
{  
  "detailId": 1,  
  "externalKey": "ABC123",  
  "description": "ABC 123",  
  "status": "Active",  
  "additionalProperties": "",  
  "updateDateTime": "2026-06-02T00:00:00",  
  "createUserName": "",  
  "updateUserName": ""  
}
```



14.40.8. Update records under individual Ace values using externalKey

PUT`https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details/externalKey/{externalKey}`

Try it

This will be used to update records under individual Ace values using externalKey

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:custom-data:modify

Authorize

Use Your Own Token

Path Parameters

> **customDataId** integer<int64> *

Format: int64

> **externalKey** string *

Query Parameters

> **async** boolean

Body Parameters

▼ **body** CustomDataDetail *

Description

Update Custom Data Details Request Object

> **detailId** integer<int64>

Format: int64

> **externalKey** string *> **description** string *> **status** string> **additionalProperties** map<string, string>

Responses

▼ ● 200 Success



CustomDataDetail

- > detailId integer<int64>
- > externalKey string *
- > description string *
- > status string
- > additionalProperties map<string, string>
- > updateDateTime string
- > createUserName string
- > updateUserName string

202 Accepted

AsyncTask

- > asyncTaskId integer<int64>
- > taskTypeCode string
- > taskCode string
- > requestDate string<date-time>
- > requestData object
- > statusCode string
- > responseDate string<date-time>
- > responseData object
- > fileName string
- > lineNumber integer
- > fileData string

400 Bad request

ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request PUT \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/custom-data/{customDataId}/details/externalKey;{externalKey}?async=boolean' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "externalKey": "ABC123",
    "description": "ABC 123"
  }'
```

Responses



```
// Success
{
  "detailId": 1,
  "externalKey": "ABC123",
  "description": "ABC 123",
  "status": "Active",
  "additionalProperties": "",
  "updateDateTime": "2026-06-02T00:00:00",
  "createUserName": "",
  "updateUserName": ""
}
```



14.41. data-codes

14.41.1. Get data codes

GET

<https://api.b.preview.loyaltyservices.io/api/v3/data-codes>

Try it ▶

Gets data codes

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:data-codes:read`

Authorize

Use Your Own Token

Query Parameters

> `includeInactive` `boolean`

Responses

✓ ● 200 Success

Code[]

- > `codeId` `integer<int64>`
- > `type` `string` *
- > `code` `string` *
- > `description` `string` *
- > `notes` `string`
- > `sortOrder` `integer`
- > `otherCode` `string`
- > `display` `boolean`
- > `active` `boolean`
- > `children` `Code[]`
- > `parent` `Code`

✓ ● 400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/data-codes?includeInactive=boolean' \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "codeId": 1,
    "type": "AccrualType",
    "code": "B",
    "description": "Base",
    "notes": "",
    "sortOrder": 1,
    "otherCode": "",
    "display": false,
    "active": false,
    "children": [
      {
        "": ""
      }
    ],
    "parent": {
      "": ""
    }
  }
]
```



14.41.2. Create Data Code

POST

<https://api.b.preview.loyaltyservices.io/api/v3/data-codes>

Try it ▶

Create a new data code

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:data-codes:modify

Authorize

Use Your Own Token

Body Parameters

▼ body Code *

> type string *

> code string *

> description string *

> notes string

> sortOrder integer

> otherCode string

> display boolean

> active boolean

> children Code[]

> parent Code

Responses

▼ ● 200 Success



Code

```
> codeId integer<int64>
> type string *
> code string *
> description string *
> notes string
> sortOrder integer
> otherCode string
> display boolean
> active boolean
> children Code[]
> parent Code
```

● 400 Bad request

ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/data-codes \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "type": "AccrualType",
    "code": "B",
    "description": "Base"
  }'
```




Responses

● 200

```
// Success
{
  "codeId": 1,
  "type": "AccrualType",
  "code": "B",
  "description": "Base",
  "notes": "",
  "sortOrder": 1,
  "otherCode": "",
  "display": false,
  "active": false,
  "children": [
    {
      "": ""
    }
  ],
  "parent": {
    "": ""
  }
}
```



14.41.3. Get data codes by code type

GET

`https://api.b.preview.loyaltyservices.io/api/v3/data-codes/{codeType}`

Try it ▶

Gets data codes by code type

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:data-codes:read`

Authorize

Use Your Own Token

Path Parameters

> `codeType` `string` *

Query Parameters

> `includeInactive` `boolean`

Responses

▼ ● 200 Success

▼ `Code[]`

> `codeId` `integer<int64>`

> `type` `string` *

> `code` `string` *

> `description` `string` *

> `notes` `string`

> `sortOrder` `integer`

> `otherCode` `string`

> `display` `boolean`

> `active` `boolean`

> `children` `Code[]`

> `parent` `Code`

▼ ● 400 Bad request

 ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/data-codes/{codeType} \
  --header 'accept: application/json'
```

Responses

 200

```
// Success
[
  {
    "codeId": 1,
    "type": "AccrualType",
    "code": "B",
    "description": "Base",
    "notes": "",
    "sortOrder": 1,
    "otherCode": "",
    "display": false,
    "active": false,
    "children": [
      {
        "": ""
      }
    ],
    "parent": {
      "": ""
    }
  }
]
```



14.41.4. Gets data code by type and code

GET

`https://api.b.preview.loyaltyservices.io/api/v3/data-codes/{codeType}/codes/{code}`

Try it

Get data code by type and code

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:data-codes:read`

Authorize

Use Your Own Token

Path Parameters

> `codeType` `string` *

> `code` `string` *

Query Parameters

> `includeInactive` `boolean`

Responses

✓ ● 200 Success

▾ Code

> `codeId` `integer<int64>`

> `type` `string` *

> `code` `string` *

> `description` `string` *

> `notes` `string`

> `sortOrder` `integer`

> `otherCode` `string`

> `display` `boolean`

> `active` `boolean`

> `children` `Code[]`

> `parent` `Code`

✓ ● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/data-codes/{codeType}/codes/{code} \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
{
  "codeId": 1,
  "type": "AccrualType",
  "code": "B",
  "description": "Base",
  "notes": "",
  "sortOrder": 1,
  "otherCode": "",
  "display": false,
  "active": false,
  "children": [
    {
      "": ""
    }
  ],
  "parent": {
    "": ""
  }
}
```



14.41.5. Update datacode by codeId

PUT`https://api.b.preview.loyaltyservices.io/api/v3/data-codes/{codeId}`

Try it

Update datacode - to 'delete' a datacode, active and display fields on request object must both be set to false.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:data-codes:modify

Authorize

Use Your Own Token

Path Parameters

> **codeId** integer<int64> *

Format: int64

Body Parameters

▼ **body** Code *> **type** string *> **code** string *> **description** string *> **notes** string> **sortOrder** integer> **otherCode** string> **display** boolean> **active** boolean> **children** Code[]> **parent** Code

Responses

▼ ● 200 Success



Code

```
> codeId integer<int64>
> type string *
> code string *
> description string *
> notes string
> sortOrder integer
> otherCode string
> display boolean
> active boolean
> children Code[]
> parent Code
```

● 400 Bad request

ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/data-codes/{codeId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "type": "AccrualType",
    "code": "B",
    "description": "Base"
  }'
```



Responses

● 200

```
// Success
{
  "codeId": 1,
  "type": "AccrualType",
  "code": "B",
  "description": "Base",
  "notes": "",
  "sortOrder": 1,
  "otherCode": "",
  "display": false,
  "active": false,
  "children": [
    {
      "": ""
    }
  ],
  "parent": {
    "": ""
  }
}
```




14.41.6. Gets all data code types

GET

`https://api.b.preview.loyaltyservices.io/api/v3/data-code-types`

Try it ▶

Returns all client-managed code types

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:data-codes:read`

Authorize

Use Your Own Token

Responses

▼ ● 200 Success

▼ CodeType[]

> name string

> description string

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/data-code-types \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "name": "CloseReason",
    "description": "Defines a Closed Reason code that is shown when a member account is closed"
  }
]
```





14.42. managed-entity

14.42.1. Retrieve managed entity options.

GET

`https://api.b.preview.loyaltyservices.io/api/v3/managed-entities/{entityType}/options`

Try it ▶

Retrieves the available options for a managed entity.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:managed-entity:read`

Authorize

Use Your Own Token

Path Parameters

> `entityType` string *

Responses

✓ ● 200 Success

▾ ManagedEntityOption[]

> `id` integer<int64>

`code` string

`description` string

▾ ● 400 Bad request

▾ ErrorResponse

> `errors` ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

📄 Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/managed-entities/{entityType}/options \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "id": "",
    "code": "",
    "description": ""
  }
]
```



14.43. expressions

14.43.1. Find expression builder configuration

GET

<https://api.b.preview.loyaltyservices.io/api/v3/expression-builder/config>

Try it ▶

Returns a list of field configurations for the expression builder based on the provided types

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:expression-config:read

Authorize

Use Your Own Token

Query Parameters

> type string *

> context string *

PERCOLATE

Responses

▼ ● 200 Success

▼ EntityConfigurationData[]

> fieldName string

> dataType string

> dataHint string

searchable boolean

oid boolean

collection boolean

> updatable boolean

> contentKey string

> groupName string

> displayOrder integer

▼ ● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/expression-builder/config?type=string&context=PERCOLATE' \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "fieldName": "member.memberPreference.preferenceValue.raw",
    "dataType": "TEXT",
    "dataHint": "",
    "searchable": false,
    "oid": false,
    "collection": false,
    "updatable": true,
    "contentKey": "Preference Value",
    "groupName": "member.memberPreference",
    "displayOrder": 0
  }
]
```



14.44. report

14.44.1. Under Investigation Report

GET

<https://api.b.preview.loyaltyservices.io/api/v3/reports/under-investigation>

Try it ▶

Get Under Investigation Report

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:reports:under-investigation

Authorize

Use Your Own Token

Responses

▼ ● 200 Success

▼ UnderInvestigationReport[]

- > createDate string<date-time>
- > memberIdentifier integer<int64>
- > accountNumber string
- > fullName string
- > country string
- > reasonCodes string
- > pointBalance number
- > suspendedSource string
- > instances integer

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request

 Curl

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/reports/under-investigation \  
  --header 'accept: application/json'
```

Responses

 200

```
// Success  
[  
  {  
    "createDate": "",  
    "memberIdentifier": "",  
    "accountNumber": "123456789",  
    "fullName": "John Smith",  
    "country": "US",  
    "reasonCodes": "Stay > Total Points Threshold",  
    "pointBalance": 12345,  
    "suspendedSource": "System",  
    "instances": 1  
  }  
]
```




14.45. hotel-daily-summaries

14.45.1. Get hotel daily summaries

GET`https://api.b.preview.loyaltyservices.io/api/v3/hotel-daily-summaries/{locationCode}/{yearMonth}`

Try it

Retrieve hotel daily summaries for a location and month.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:compensation:read

Authorize

Use Your Own Token

Path Parameters

> **locationCode** string *> **yearMonth** string *

Responses

✓ ● 200 Success

▼ HotelDailySummary[]

- > **date** string<date> *
- > **currencyTypeCode** string
- > **currencyExchangeRate** number
- > **availableRoomCount** integer
- > **occupiedRoomCount** integer *
- > **averageDailyRateUSD** number *
- > **averageDailyRateIntl** number

▼ ● 400 Bad request

▼ ErrorResponse

- > **errors** ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/hotel-daily-summaries/{locationCode}/{yearMonth} \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "date": "",
    "currencyTypeCode": "",
    "currencyExchangeRate": 1.5,
    "availableRoomCount": 100,
    "occupiedRoomCount": 10,
    "averageDailyRateUSD": 10.24,
    "averageDailyRateIntl": 11.21
  }
]
```



14.45.2. Update hotel daily summaries

PUT

https://api.b.preview.loyaltyservices.io/api/v3/hotel-daily-summaries/{locationCode}/{yearMonth}

Try it ▶

Update hotel daily summaries for a location and month. All of the days in the month should be sent in the request when calling this service. If the caller does not want to update the data for one of the days the following fields should be left blank- availableRoomCount,occupiedRoomCount,averageDailyRateIntl. It is also expected that all ADR values are entered in using the averageDailyRateIntl field.

Credentials

OAuth 2.0

OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow:

Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:compensation:modify

Authorize

Use Your Own Token

Path Parameters

> locationCode string *

> yearMonth string *

Query Parameters

> verified boolean

> async boolean

Body Parameters

body HotelDailySummary[] *

> date string<date> *

> currencyTypeCode string

> availableRoomCount integer

> occupiedRoomCount integer *

> averageDailyRateUSD number *

> averageDailyRateIntl number

mm/dd/yyyy

USD

+

Responses

200 Success

999 / 1062

Created at Oct 02, 2026



▼ HotelDailySummary[]

- > date string<date> *
- > currencyTypeCode string
- > currencyExchangeRate number
- > availableRoomCount integer
- > occupiedRoomCount integer *
- > averageDailyRateUSD number *
- > averageDailyRateIntl number

▼ ● 202 Accepted

▼ AsyncTask

- > asyncTaskId integer<int64>
- > taskTypeCode string
- > taskCode string
- > requestDate string<date-time>
- > requestData object
- > statusCode string
- > responseDate string<date-time>
- > responseData object
- > fileName string
- > lineNumber integer
- > fileData string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request PUT \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/hotel-daily-summaries/{locationCode}/{yearMonth}?verified=boolean&async=' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '[
  {
    "date": "",
    "currencyTypeCode": "USD",
    "occupiedRoomCount": 10,
    "averageDailyRateUSD": 10.24
  }
]'
```

Responses

● 200

```
// Success
[
  {
    "date": "",
    "currencyTypeCode": "",
    "currencyExchangeRate": 1.5,
    "availableRoomCount": 100,
    "occupiedRoomCount": 10,
    "averageDailyRateUSD": 10.24,
    "averageDailyRateIntl": 11.21
  }
]
```



14.46. user

14.46.1. Get user context

GET

<https://api.b.preview.loyaltyservices.io/api/v3/user>

Try it ▶

Get a user and its entitlements

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:user:read

Authorize

Use Your Own Token

Responses

- ▼

● 200

Success

▼

User

> userId integer<int64>

> name string

> loginName string *

> firstName string *

> lastName string *

> emailAddress string

> active boolean

> typeCode string

userRoleNames string[]

> userEntitlements UserEntitlement[]
- ▼

● 400

Bad request

▼

ErrorResponse

> errors ValidationError[]
- 401

Unauthorized
- 403

Forbidden



● 500 Internal error

Request

Curl

```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/user \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "userId": "",  
  "name": "",  
  "loginName": "",  
  "firstName": "",  
  "lastName": "",  
  "emailAddress": "",  
  "active": false,  
  "typeCode": "CC (Call Center), P (Partner), API (platform-api-security client)",  
  "userRoleNames": [  
    ""  
  ],  
  "userEntitlements": [  
    {  
      "name": "twa:member:search",  
      "description": "Tally Call Center: Search Member"  
    }  
  ]  
}
```



POST <https://api.b.preview.loyaltyservices.io/api/v3/users> **Try it** ▶

Creates a new user for accessing the Tally frontend application or for granting API access

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:user:modify

Authorize

Use Your Own Token

Body Parameters

[illegible]

Responses

▼ ● 200 Success



▼ User

```
> userId integer<int64>

> name string

> loginName string *

> firstName string *

> lastName string *

> emailAddress string

> active boolean

> typeCode string

    userRoleNames string[]

> userEntitlements UserEntitlement[]
```

▼ ● 400 Bad Request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden

● 404 Not Found

▼ ● 409 Conflict

▼ ErrorResponse

```
> errors ValidationError[]
```

● 500 Internal Server Error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/users \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "loginName": "",
    "firstName": "",
    "lastName": ""
  }'
```



Responses

● 200

```
// Success
{
  "userId": "",
  "name": "",
  "loginName": "",
  "firstName": "",
  "lastName": "",
  "emailAddress": "",
  "active": false,
  "typeCode": "CC (Call Center), P (Partner), API (platform-api-security client)",
  "userRoleNames": [
    ""
  ],
  "userEntitlements": [
    {
      "name": "twa:member:search",
      "description": "Tally Call Center: Search Member"
    }
  ]
}
```



14.46.3. Update User

PUT

<https://api.b.preview.loyaltyservices.io/api/v3/users/{userId}>

Try it ▶

Updates an existing user

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: **Client Credentials**

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: **api:user:modify**

Authorize

Use Your Own Token

Path Parameters

```
> userId integer<int64> *
```

Format: int64

Body Parameters

body User *

```
> name string
```

```
> loginName string *
```

```
> firstName string *
```

```
> lastName string *
```

```
> emailAddress string
```

```
> active boolean
```

```
> typeCode string
```

```
> password string
```

```
> confirmPassword string
```

```
userRoleNames string[]
```

 $+$

Responses

▼ ● 200 Success



▼ User

```
> userId integer<int64>

> name string

> loginName string *

> firstName string *

> lastName string *

> emailAddress string

> active boolean

> typeCode string

    userRoleNames string[]

> userEntitlements UserEntitlement[]
```

▼ ● 400 Bad Request

▼ ErrorResponse

```
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal Server Error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/users/{userId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "loginName": "",
    "firstName": "",
    "lastName": ""
  }'
```



Responses

● 200

```
// Success
{
  "userId": "",
  "name": "",
  "loginName": "",
  "firstName": "",
  "lastName": "",
  "emailAddress": "",
  "active": false,
  "typeCode": "CC (Call Center), P (Partner), API (platform-api-security client)",
  "userRoleNames": [
    ""
  ],
  "userEntitlements": [
    {
      "name": "twa:member:search",
      "description": "Tally Call Center: Search Member"
    }
  ]
}
```



14.47. user-role

14.47.1. Create a User Role, including Entitlements when provided.

POST

https://api.b.preview.loyaltyservices.io/api/v3/user-roles

Try it ▶

Credentials

OAuth 2.0

OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow:

Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:user-role:modify

Authorize

Use Your Own Token

Body Parameters

body UserRole *

> roleName string *

> roleDescription string *

> active boolean

> rank integer *

> channelCode string

> entitlements UserEntitlement[]

Responses

▼

● 200 Success



▼ UserRole

- > `roleId` integer<int64>
- > `roleName` string *
- > `roleDescription` string *
- > `active` boolean
- > `rank` integer *
- > `channelCode` string
- > `entitlements` UserEntitlement[]

▼ ● 400 Bad request

▼ ErrorResponse

- > `errors` ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request

Curl

```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/user-roles \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "roleName": "",
    "roleDescription": "",
    "rank": ""
  }'
```

Responses

● 200

```
// Success
{
  "roleId": "",
  "roleName": "",
  "roleDescription": "",
  "active": false,
  "rank": 0,
  "channelCode": "API",
  "entitlements": [
    {
      "name": "twa:member:search",
      "description": "Tally Call Center: Search Member"
    }
  ]
}
```





14.47.2. Retrieve all user roles. Include entitlements if specified.

GET

<https://api.b.preview.loyaltyservices.io/api/v3/user-roles>

Try it ▶

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:user-role:read

Authorize

Use Your Own Token

Query Parameters

> includeEntitlements boolean

☐

Responses

▼ ● 200 Success

```
▼ UserRole[]
> userRoleId integer<int64>
> roleName string *
> roleDescription string *
> active boolean
> rank integer *
> channelCode string
> entitlements UserEntitlement[]
```

▼ ● 400 Bad request

```
▼ ErrorResponse
> errors ValidationError[]
```

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/user-roles?includeEntitlements=boolean' \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "userRoleId": "",
    "roleName": "",
    "roleDescription": "",
    "active": false,
    "rank": 0,
    "channelCode": "API",
    "entitlements": [
      {
        "name": "twa:member:search",
        "description": "Tally Call Center: Search Member"
      }
    ]
  }
]
```



14.47.3. Update User Role and associate Entitlement if provided and active

PUT

https://api.b.preview.loyaltyservices.io/api/v3/user-roles/{userRoleId}

Try it ▶

Credentials

OAuth 2.0

OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow:

Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:user-role:modify

Authorize

Use Your Own Token

Path Parameters

> userRoleId integer<int64> *

Format: int64

Body Parameters

▼ body UserRole *

> roleName string *

> roleDescription string *

> active boolean

> rank integer *

> channelCode string

> entitlements UserEntitlement[]

Responses

▼ ● 200 Success

▼ UserRole

> userRoleId integer<int64>

> roleName string *

> roleDescription string *

> active boolean

> rank integer *

> channelCode string

> entitlements UserEntitlement[]



▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/user-roles/{userRoleId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "roleName": "",
    "roleDescription": "",
    "rank": ""
  }'
```

Responses

● 200

```
// Success
{
  "userRoleId": "",
  "roleName": "",
  "roleDescription": "",
  "active": false,
  "rank": 0,
  "channelCode": "API",
  "entitlements": [
    {
      "name": "twa:member:search",
      "description": "Tally Call Center: Search Member"
    }
  ]
}
```



14.48. user-task

14.48.1. Find user tasks

GET

<https://api.b.preview.loyaltyservices.io/api/v3/user-tasks>

Try it ▶

Retrieve user tasks

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:user-task:read

Authorize

Use Your Own Token

Query Parameters

> startDate string<date>

mm/dd/yyyy 📅

> endDate string<date>

mm/dd/yyyy 📅

> includeClosed boolean

☐

> assignee string

> statuses string[]

+

Responses

▼ ● 200 Success



▼ UserTask[]

- > userTaskId integer<int64>
- > title string
- > status string
- > followUpFlag boolean
- > followUpDate string<date>
- > assignedTo string
- > locked boolean
- > lockerId integer<int64>
- > lockerName string
- > lockerRank integer<int32>
- > comments UserTaskComment[]

▼ ● 400 Bad Request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal Server Error

Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/user-tasks?includeClosed=boolean' \  
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "userTaskId": "",
    "title": "",
    "status": "",
    "followUpFlag": false,
    "followUpDate": "",
    "assignedTo": "",
    "locked": false,
    "lockerId": "",
    "lockerName": "",
    "lockerRank": "",
    "comments": [
      {
        "createDate": "",
        "createdBy": "",
        "comment": "",
        "assignedTo": "",
        "assignedFrom": ""
      }
    ]
  }
]
```



14.48.2. Create user task

POST

https://api.b.preview.loyaltyservices.io/api/v3/user-tasks

Try it ▶

Create a new user task

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:user-task:modify

Authorize

Use Your Own Token

Body Parameters

body UserTaskRequest *

> title string *

> status string *

> followUpFlag boolean *

> followUpDate string<date>

> assignedTo integer<int64>

> comment string *

mm/dd/yyyy 📅

Format: int64

Responses

200 Success



▼ UserTask

> **taskId** integer<int64>

> **title** string

> **status** string

> **followUpFlag** boolean

> **followUpDate** string<date>

> **assignedTo** string

> **locked** boolean

> **lockerId** integer<int64>

> **lockerName** string

> **lockerRank** integer<int32>

> **comments** UserTaskComment[]

▼ ● 400 Bad request

▼ ErrorResponse

> **errors** ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request



```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/user-tasks \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "title": "",
    "status": "",
    "followUpFlag": "",
    "comment": ""
  }'
```



Responses

● 200

```
// Success
{
  "userTaskId": "",
  "title": "",
  "status": "",
  "followUpFlag": false,
  "followUpDate": "",
  "assignedTo": "",
  "locked": false,
  "lockerId": "",
  "lockerName": "",
  "lockerRank": "",
  "comments": [
    {
      "createDate": "",
      "createdBy": "",
      "comment": "",
      "assignedTo": "",
      "assignedFrom": ""
    }
  ]
}
```



14.48.3. Get user task

GET

`https://api.b.preview.loyaltyservices.io/api/v3/user-tasks/{taskId}`

Try it

Retrieve a user task

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:user-task:read

Authorize

Use Your Own Token

Path Parameters

> `taskId integer<int64> *`

Format: int64

Responses

✓ ● 200 Success

▼ UserTask

```
> taskId integer<int64>
> title string
> status string
> followUpFlag boolean
> followUpDate string<date>
> assignedTo string
> locked boolean
> lockerId integer<int64>
> lockerName string
> lockerRank integer<int32>
> comments UserTaskComment[]
```

✓ ● 400 Bad request

▼ ErrorResponse

```
> errors ValidationError[]
```

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/user-tasks/{userTaskId} \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
{
  "userTaskId": "",
  "title": "",
  "status": "",
  "followUpFlag": false,
  "followUpDate": "",
  "assignedTo": "",
  "locked": false,
  "lockerId": "",
  "lockerName": "",
  "lockerRank": "",
  "comments": [
    {
      "createDate": "",
      "createdBy": "",
      "comment": "",
      "assignedTo": "",
      "assignedFrom": ""
    }
  ]
}
```



14.48.4. Update user task

PUT

<https://api.b.preview.loyaltyservices.io/api/v3/user-tasks/{userTaskId}>

Try it ▶

Update a user task

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:user-task:modify

Authorize

Use Your Own Token

Path Parameters

> userTaskId integer<int64> *

Format: int64

Body Parameters

body UserTaskRequest *

> title string *

> status string *

> followUpFlag boolean *

> followUpDate string<date>

> assignedTo integer<int64>

> comment string *

mm/dd/yyyy 📅

Format: int64

Responses

200 Success



▼ UserTask

- > **taskId** integer<int64>
- > **title** string
- > **status** string
- > **followUpFlag** boolean
- > **followUpDate** string<date>
- > **assignedTo** string
- > **locked** boolean
- > **lockerId** integer<int64>
- > **lockerName** string
- > **lockerRank** integer<int32>
- > **comments** UserTaskComment[]

▼ ● 400 Bad request

▼ ErrorResponse

- > **errors** ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request PUT \
  --url https://api.b.preview.loyaltyservices.io/api/v3/user-tasks/{taskId} \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "title": "",
    "status": "",
    "followUpFlag": "",
    "comment": ""
  }'
```



Responses

● 200

```
// Success
{
  "userTaskId": "",
  "title": "",
  "status": "",
  "followUpFlag": false,
  "followUpDate": "",
  "assignedTo": "",
  "locked": false,
  "lockerId": "",
  "lockerName": "",
  "lockerRank": "",
  "comments": [
    {
      "createDate": "",
      "createdBy": "",
      "comment": "",
      "assignedTo": "",
      "assignedFrom": ""
    }
  ]
}
```



14.48.5. Lock user task

POST`https://api.b.preview.loyaltyservices.io/api/v3/user-tasks/{taskId}/locks`

Try it

Lock a user task

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:user-task:modify

Authorize

Use Your Own Token

Path Parameters

> `taskId integer<int64> *`

Format: int64

Responses

▾ ● 200 Success

▾ UserTask

```
> taskId integer<int64>
> title string
> status string
> followUpFlag boolean
> followUpDate string<date>
> assignedTo string
> locked boolean
> lockerId integer<int64>
> lockerName string
> lockerRank integer<int32>
> comments UserTaskComment[]
```

▾ ● 400 Bad request

▾ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/user-tasks/{userTaskId}/locks \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "userTaskId": "",  
  "title": "",  
  "status": "",  
  "followUpFlag": false,  
  "followUpDate": "",  
  "assignedTo": "",  
  "locked": false,  
  "lockerId": "",  
  "lockerName": "",  
  "lockerRank": "",  
  "comments": [  
    {  
      "createDate": "",  
      "createdBy": "",  
      "comment": "",  
      "assignedTo": "",  
      "assignedFrom": ""  
    }  
  ]  
}
```



14.48.6. Unlock user task

DELETE`https://api.b.preview.loyaltyservices.io/api/v3/user-tasks/{taskId}/locks`

Try it

Unlock a user task

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:user-task:modify

Authorize

Use Your Own Token

Path Parameters

> `taskId integer<int64> *`

Format: int64

Responses

▾ ● 200 Success

▾ UserTask

```
> taskId integer<int64>
> title string
> status string
> followUpFlag boolean
> followUpDate string<date>
> assignedTo string
> locked boolean
> lockerId integer<int64>
> lockerName string
> lockerRank integer<int32>
> comments UserTaskComment[]
```

▾ ● 400 Bad request

▾ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request DELETE \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/user-tasks/{userTaskId}/locks \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "userTaskId": "",  
  "title": "",  
  "status": "",  
  "followUpFlag": false,  
  "followUpDate": "",  
  "assignedTo": "",  
  "locked": false,  
  "lockerId": "",  
  "lockerName": "",  
  "lockerRank": "",  
  "comments": [  
    {  
      "createDate": "",  
      "createdBy": "",  
      "comment": "",  
      "assignedTo": "",  
      "assignedFrom": ""  
    }  
  ]  
}
```



14.49. badge

14.49.1. Get Badges

GET

https://api.b.preview.loyaltyservices.io/api/v3/badges

Try it ▶

Retrieve badges

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:badge:read

Authorize

Use Your Own Token

Responses

▼ ● 200 Success

▼ BadgeData[]

Description
Badge details

> badgeId integer<int64>

> badgeTypeCode string

> badgeName string

> badgeDescription string

> statusCode string

> expirationDate string<date>

> expirationType string

> expirationDayPeriod integer<int32>

> expirationTimePeriod integer<int32>

> logoKey string

> content GenericItem

▼ ● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/badges \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
[  
  {  
    "badgeId": 147258369,  
    "badgeTypeCode": "General",  
    "badgeName": "Event Organizer",  
    "badgeDescription": "An organizer who brings in over $100,000 year in business",  
    "statusCode": "Active",  
    "expirationDate": "4000-01-01",  
    "expirationType": "Exact",  
    "expirationDayPeriod": 60,  
    "expirationTimePeriod": 10,  
    "logoKey": "elite-explorer.png",  
    "content": {  
      "contentCode": "Bonus100",  
      "contentTypeCode": "HEADER",  
      "name": "Bonus 100 points",  
      "shortDescription": "Award 100 points",  
      "description": "Award 100 points on 100th stay",  
      "alternateDescription": "Award 100 points on 100th stay",  
      "termsAndConditions": "Terms and conditions for this promotion",  
      "title": "Award 100 points",  
      "label": "100 points awarded",  
      "logoKey": "100-points.png",  
      "displayTypeKey": "100 points",  
      "startDate": "",  
      "endDate": "",  
      "url": "/loyaltyprogram/bonus/100-points",  
      "tierCode": "B",  
      "displayForAll": true,  
      "descriptorCode": "LIMITED_TIME"  
    }  
  }  
]
```



14.50. coupon

14.50.1. Create a coupon request.

POST

<https://api.b.preview.loyaltyservices.io/api/v3/coupon-requests>

Try it ▶

Creates new coupon(s) that can be assigned to a member with the coupon-use service. This service will create the coupon request and generate the coupons.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:coupon-request:modify

Authorize

Use Your Own Token

Query Parameters

- > activateCoupons boolean

> useCoupons boolean

> async boolean
- ☐

☐

☐



Body Parameters

▼

body CouponRequest *

Description

Create coupon request object

>

offerCode string *

>

name string *

>

description string

>

expirationType string

Rolling

>

startDate string<date>

mm/dd/yyyy

📅

>

expirationDate string<date>

mm/dd/yyyy

📅

>

expirationDayPeriod integer<int32>

Format: int32

>

expirationTimePeriod integer<int32>

Format: int32

>

useTypeCode string

Single

>

couponRequestType string

Bulk

>

memberIdentifier string

>

couponCount integer<int32>

1

>

attributes CouponRequestAttribute[]

Responses

▼

●

200 Success



- ▼ Coupon[]
 - > couponId integer<int64>
 - > couponRequestId integer<int64>
 - > offer Offer
 - > code string
 - > statusCode string
 - > startDate string<date>
 - > endDate string<date>
 - > useTypeCode string
 - > useLimit integer<int32>
 - > currentUse integer<int32>
 - > couponUses CouponUse[]
 - > couponAttributes CouponAttribute[]

▼ ● 202 Accepted

- ▼ AsyncTask
 - > asyncTaskId integer<int64>
 - > taskTypeCode string
 - > taskCode string
 - > requestDate string<date-time>
 - > requestData object
 - > statusCode string
 - > responseDate string<date-time>
 - > responseData object
 - > fileName string
 - > lineNumber integer
 - > fileData string

▼ ● 400 Bad request

- ▼ ErrorResponse
 - > errors ValidationError[]



- **401** Unauthorized
- **403** Forbidden
- **500** Internal error

Request



```
curl --request POST \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/coupon-requests?activateCoupons=boolean&useCoupons=boolean&async=boolean' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "offerCode": "Offer12345",
    "name": "",
    "expirationType": "Rolling",
    "useTypeCode": "Single",
    "couponRequestType": "Bulk",
    "couponCount": 1,
    "attributes": [
      {
        "name": "site_tier",
        "value": 123
      }
    ]
  }'
```



Responses

● 200



```
// Success
[
  {
    "couponId": 147258369,
    "couponRequestId": 147258370,
    "offer": {
      "offerId": 147258369,
      "offerTypeCode": "Discount",
      "code": "ALON2020",
      "name": "Free night",
      "description": "Award double points",
      "featureVersion": 2,
      "statusCode": "Enabled",
      "startDateTime": "",
      "endDateTime": "",
      "offerAttributes": [
        {
          "offerAttributeId": 147258369,
          "attribute": {
            "attributeId": 147258369,
            "typeCode": "Validation",
            "statusCode": "Enabled",
            "name": "AccountNumber",
            "description": "Account Number",
            "dataTypeCode": "open"
          }
        }
      ],
      "defaultValue": "Standard",
    }
  }
]
```



14.51. adjustment-rules

14.51.1. Get adjustment rules

GET

<https://api.b.preview.loyaltyservices.io/api/v3/adjustment-rules>

Try it ▶

Gets all adjustment rules

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:adjustment-rules:read

Authorize

Use Your Own Token

Query Parameters

> includeInactive `boolean`

☐

> includeHidden `boolean`

☐

Responses

▼ ● 200 Success

▼ AdjustmentRule[]

> adjustmentCode `string`

> adjustmentDescription `string`

> points `number`

> pointType `string`

> pointCategory `string`

> billingCode `string`

> resultTierQualifier `boolean`

> status `string`

> display `boolean`

▼ ● 400 Bad request



▼ ErrorResponse

> errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/adjustment-rules?includeInactive=boolean&includeHidden=boolean' \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "adjustmentCode": "Employee_Reward",
    "adjustmentDescription": "Employee Reward",
    "points": 1000,
    "pointType": "POINTS",
    "pointCategory": "P",
    "billingCode": "BILLING123",
    "resultTierQualifier": false,
    "status": "A",
    "display": true
  }
]
```



14.52. outbound-notification

14.52.1. Get outbound notification types

GET`https://api.b.preview.loyaltyservices.io/api/v3/outbound-notification-types`**Try it** ▶

Returns a list of outbound notification types that are currently enabled/disabled in the system. Used for reconciliation and verification purposes.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>Flow: **Client Credentials**

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: `api:outbound-notification:read`**Authorize**

Use Your Own Token

Responses▼ ● **200** Success▼ `OutboundNotification[]`> `type` `string`> `enabled` `boolean`▼ ● **400** Bad request▼ `ErrorResponse`> `errors` `ValidationError[]`● **401** Unauthorized● **403** Forbidden● **500** Internal error**Request** **Curl**

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/outbound-notification-types \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "type": "",
    "enabled": false
  }
]
```



14.52.2. fetch all outbound notifications sent within a specified timeframe

GET

`https://api.b.preview.loyaltyservices.io/api/v3/outbound-notifications/{type}`

Try it ▶

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:outbound-notification:read

Authorize

Use Your Own Token

Path Parameters

> type string *

Select... ▾

Query Parameters

> startDate string<date>

mm/dd/yyyy



> endDate string<date>

mm/dd/yyyy



> pageNumber integer

1

> pageSize integer

500

Responses

✓ ● 200 Success

FindOutboundNotificationResponse

> outboundNotifications OutboundNotificationPage[]

> pageInfo PageInfo

✓ ● 400 Bad request

ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error



Request



```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/outbound-notifications/{type}?pageNumber=1&pageSize=500' \  
  --header 'accept: application/json'
```

Responses

● 200

```
// Success  
{  
  "outboundNotifications": [  
    {  
      "objectId": "",  
      "type": "",  
      "snapshotDate": ""  
    }  
  ],  
  "pageInfo": {  
    "number": 0,  
    "size": 0,  
    "totalItems": 0,  
    "hasNext": false  
  }  
}
```



14.52.3. Send an outbound notification

POST`https://api.b.preview.loyaltyservices.io/api/v3/outbound-notifications/{type}/{objectId}`

Try it ▶

Sends outbound notifications to the client's configured webhook service.

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:outbound-notification:modify

Authorize

Use Your Own Token

Path Parameters

> type string *

Select... ▼

> objectId integer<int64> *

Format: int64

Responses

▼ ● 200 Success

▼ OutboundNotificationResponse

> snapshotDate string<date-time>

> objectId integer<int64>

> accountNumber string

> additionalProperties object

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/outbound-notifications/{type}/{objectId} \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "snapshotDate": "",
  "objectId": "",
  "accountNumber": "",
  "additionalProperties": {}
}
```



14.53. standard-reports

14.53.1. fetch results for a standard report

GET

https://api.b.preview.loyaltyservices.io/api/v3/standard-reports/{standardReportName}

Try it ▶

Credentials

OAuth 2.0

OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow:

Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:standard-reports:read

Authorize

Use Your Own Token

Path Parameters

> standardReportName string *

Query Parameters

> startDate string<date>

mm/dd/yyyy

> endDate string<date>

mm/dd/yyyy

> pageNumber integer

1

> pageSize integer

500

Responses

▼ ● 200 Success

▼ StandardReportResponse

> resultRows object[]

> pageInfo PageInfo

▼ ● 400 Bad request

▼ ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

1048 / 1062

Created at Oct 02, 2026



Request

 Curl

```
curl --request GET \  
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/standard-reports/{standardReportName}?pageNumber=1&pageSize=500' \  
  --header 'accept: application/json'
```

Responses

 200

```
// Success  
{  
  "resultRows": [  
    {}  
  ],  
  "pageInfo": {  
    "number": 0,  
    "size": 0,  
    "totalItems": 0,  
    "hasNext": false  
  }  
}
```



14.54. enroll-codes

14.54.1. Get enroll codes

GET

<https://api.b.preview.loyaltyservices.io/api/v3/enroll-codes>

Try it

Gets all enroll codes

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1

Required Scopes: api:enroll-codes:read

Authorize

Use Your Own Token

Query Parameters

> includeInactive boolean



Responses

✓ ● 200 Success

EnrollCode[]

- > enrollCodeId integer<int64>
- > enrollCode string
- > description string
- > name string
- > status string
- > startDate string<date>
- > endDate string<date>
- > locationCode string

✓ ● 400 Bad request

ErrorResponse

- > errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden



● 500 Internal error

Request

Curl

```
curl --request GET \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/enroll-codes?includeInactive=boolean' \
  --header 'accept: application/json'
```

Responses

● 200

```
// Success
[
  {
    "enrollCodeId": 1,
    "enrollCode": "B",
    "description": "This is an example base type",
    "name": "",
    "status": "pending",
    "startDate": "2024-03-20",
    "endDate": "2024-03-21",
    "locationCode": "MSP123"
  }
]
```



14.55. customer-satisfaction

14.55.1. Customer Satisfaction

PUT

<https://api.b.preview.loyaltyservices.io/api/v3/customer-satisfaction>

Try it ▶

Updates an existing Customer Satisfaction Metric

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:customer-satisfaction:modify

Authorize

Use Your Own Token

Query Parameters

> async boolean

Body Parameters

body CustomerSatisfaction

Description

Customer Satisfaction Request

> metricType string *

> ace03 string *

> reportDate string<date> *

> override number<double>

> score number<double>

mm/dd/yyyy 📅

Format: double

Format: double

Responses

200 Success



▼ CustomerSatisfaction

- > metricType string *
- > ace03 string *
- > reportDate string<date> *
- > override number<double>
- > score number<double>

▼ ● 202 Accepted

▼ AsyncTask

- > asyncTaskId integer<int64>
- > taskTypeCode string
- > taskCode string
- > requestDate string<date-time>
- > requestData object
- > statusCode string
- > responseDate string<date-time>
- > responseData object
- > fileName string
- > lineNumber integer
- > fileData string

▼ ● 400 Bad request

▼ ErrorResponse

- > errors ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error



Request



```
curl --request PUT \
  --url 'https://api.b.preview.loyaltyservices.io/api/v3/customer-satisfaction?async=boolean' \
  --header 'accept: application/json' \
  --header 'content-type: application/json' \
  --data '{
    "metricType": "Overall Satisfaction",
    "ace03": 20763,
    "reportDate": ""
  }'
```

Responses

● 200

```
// Success
{
  "metricType": "Overall Satisfaction",
  "ace03": 20763,
  "reportDate": "",
  "override": 30,
  "score": ""
}
```



14.56. member-referral-code

14.56.1. Create Referral Code

POST`https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referral-codes`

Try it ▶

Generates a referral code for clients to use

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: `api:member-referral-codes:create`

Authorize

Use Your Own Token

Path Parameters

> `memberIdentifier` `string` *

Responses

✓ ● 200 Success

MemberReferralCodeResponse

> `memberReferralCodeId` `integer<int64>`

> `referralCode` `string`

> `active` `boolean`

> `updateDateTime` `string<date-time>`

✓ ● 400 Bad Request

ErrorResponse

> `errors` `ValidationError[]`

● 401 Unauthorized

● 500 Internal Error

Request

Curl

```
curl --request POST \
  --url https://api.b.preview.loyaltyservices.io/api/v3/members/{memberIdentifier}/referral-codes \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
{
  "memberReferralCodeId": 4,
  "referralCode": "F9DA265FD56D496C",
  "active": false,
  "updateDateTime": ""
}
```



14.57. preference

14.57.1. Get Preferences

GET

<https://api.b.preview.loyaltyservices.io/api/v3/preferences>

Try it ▶

Retrieve all available preferences that can be set on member profiles

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▾

Required Scopes: api:preference:read

Authorize

Use Your Own Token

Responses

✓ ● 200 Success

PreferenceRule[]

Description

A group of similar preferences

> code string

> description string

> sortOrder integer

> statusCode string

> details PreferenceRuleDetail[]

✓ ● 400 Bad request

ErrorResponse

> errors ValidationError[]

● 401 Unauthorized

● 403 Forbidden

● 500 Internal error

Request

Curl

```
curl --request GET \
  --url https://api.b.preview.loyaltyservices.io/api/v3/preferences \
  --header 'accept: application/json'
```



Responses

● 200

```
// Success
[
  {
    "code": "",
    "description": "",
    "sortOrder": 0,
    "statusCode": "",
    "details": [
      {
        "code": "",
        "description": "",
        "sortOrder": 0,
        "statusCode": "",
        "dataType": "",
        "optionSource": "",
        "options": [
          {
            "value": "",
            "sortOrder": 0,
            "statusCode": ""
          }
        ]
      }
    ]
  }
]
```



14.58. content

14.58.1. Create Content

POST

<https://api.b.preview.loyaltyservices.io/api/v3/content>

Try it ▶

Create a new content item

Credentials

OAuth 2.0 OAuth2

See <https://docs.wearephaedon.com/docs/platform-api-authentication>

Flow: Client Credentials

Client ID

Client Secret

Scopes 1 ▼

Required Scopes: api:generic-content:modify

Authorize

Use Your Own Token



Body Parameters

▼ body GenericItem *

> contentCode string *

> contentTypeCode string

> name string

> shortDescription string

> description string

> alternateDescription string

> termsAndConditions string

> title string

> label string

> logoKey string

> displayTypeKey string

> startDate string<date-time>

> endDate string<date-time>

> url string

> tierCode string

> displayForAll boolean

> descriptorCode string

mm/dd/yyyy, --:-- --📅

mm/dd/yyyy, --:-- --📅

Responses

▼ ● 200 Success



- GenericItem
- > **contentCode** string *
 - > **contentTypeCode** string
 - > **name** string
 - > **shortDescription** string
 - > **description** string
 - > **alternateDescription** string
 - > **termsAndConditions** string
 - > **title** string
 - > **label** string
 - > **logoKey** string
 - > **displayTypeKey** string
 - > **startDate** string<date-time>
 - > **endDate** string<date-time>
 - > **url** string
 - > **tierCode** string
 - > **displayForAll** boolean
 - > **descriptorCode** string

400 Bad request

- ErrorResponse
- > **errors** ValidationError[]

- 401 Unauthorized
- 403 Forbidden
- 500 Internal error

Request



```
curl --request POST \  
  --url https://api.b.preview.loyaltyservices.io/api/v3/content \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "contentCode": "Bonus100"  
  }'
```



Responses

● 200

```
// Success
{
  "contentCode": "Bonus100",
  "contentTypeCode": "HEADER",
  "name": "Bonus 100 points",
  "shortDescription": "Award 100 points",
  "description": "Award 100 points on 100th stay",
  "alternateDescription": "Award 100 points on 100th stay",
  "termsAndConditions": "Terms and conditions for this promotion",
  "title": "Award 100 points",
  "label": "100 points awarded",
  "logoKey": "100-points.png",
  "displayTypeKey": "100 points",
  "startDate": "",
  "endDate": "",
  "url": "/loyaltyprogram/bonus/100-points",
  "tierCode": "B",
  "displayForAll": true,
  "descriptorCode": "LIMITED_TIME"
}
```