ivanti

msp-api

ivanti

CONTENTS

1. **Overview** **3**
   - 1.1. Preface 4
   - 1.2. End User Agreement 7
2. **Retrieving the Authentication DSID** **8**
3. **API Calls** **10**
4. **MSP Portal REST API** **21**

Copyright © 2026, Ivanti, Inc. All rights reserved.

1. Overview

The MSP Portal enables a Managed Service Provider (MSP) to create and administer the tenants for which a subscription is required. The MSP Admin can create tenants, provide service entitlements, monitor usage, and schedule frequent usage reports for billing purposes.

This guide describes the MSP Portal REST API and includes the full list of supported API calls.

Copyright © 2026, Ivanti, Inc. All rights reserved.

1.1. Preface

- Document conventions
- Requesting Technical Support
- Reporting Documentation Issues

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Ivanti technical documentation.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used in the flow of the text to highlight specific words or phrases.

Format	Description
bold text	Identifies command names
	Identifies keywords and operands
	Identifies the names of user-manipulated GUI elements
	Identifies text to enter at the GUI
italic text	Identifies emphasis
	Identifies variables
	Identifies document titles
courier font	Identifies command output
	Identifies command syntax example

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Copyright © 2026, Ivanti, Inc. All rights reserved.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
[]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ x | y | z }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x|y	A vertical bar separates mutually exclusive elements.
< >	Non-printing characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, member [member ...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Code Block

Following is an example of Python based code block in the html documentation:

```
defsome_function():interesting=Falseprint'This line is highlighted.'print'This one is not...'print'...but this one is.'
```

Notes and Warnings

Note, Attention, and Caution statements might be used in this document.

> This is an example of a note. A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

Attention

This is an example of an attention statement. An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

Requesting Technical Support

Technical product support is available through the Pulse Secure Global Support Center (PSGSC). If you have a support contract, file a ticket with PSGSC.

Product warranties—For product warranty information, visit https://forums.ivanti.com/

Self-Help Online Tools and Resources

For quick and easy problem resolution, Pulse Secure provides an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

Copyright © 2026, Ivanti, Inc. All rights reserved.

- Find CSC offerings: https://forums.ivanti.com/s/contactsupport/
- Search for known bugs: https://forums.ivanti.com/
- Find product documentation: https://forums.ivanti.com/s/product-downloads
- Download the latest versions of software and review release notes: https://help.ivanti.com/
- Open a case online in the CSC Case Management tool: https://forums.ivanti.com/s/contactsupport/
- To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://forums.ivanti.com/

Opening a Case with PSGSC

You can open a case with PSGSC on the Web or by telephone.

Use the Case Management tool in the PSGSC at https://forums.ivanti.com/

Call 1-844 751 7629 (Toll Free, US).

For international or direct-dial options in countries without toll-free numbers, see https://forums.ivanti.com/s/contactsupport/

Reporting Documentation Issues

To report any errors or inaccuracies in Pulse Secure technical documentation, or to make suggestions for future improvement, contact Pulse Secure Technical Support (https://forums.ivanti.com/s/contactsupport?language=en_US). Include a full description of your issue or suggestion and the document(s) to which it relates.

Copyright © 2026, Ivanti, Inc. All rights reserved.

1.2. End User Agreement

The Ivanti product that is the subject of this technical documentation consists of (or is intended for use with) Ivanti software. Use of such software is subject to the terms and conditions of the End User License Agreement (“EULA”) posted at **https://www.ivanti.com/company/legal/eula**. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Copyright © 2026, Ivanti, Inc. All rights reserved.

2. Retrieving the Authentication DSID

The Data Set Identification (DSID) is required for API use.

The following CURL command format uses the DSID to query the REST API server:

```bash
curl -v --cookie "DSID=<value>" <api_request_url>
```

The DSID can be retrieved in two ways:

- Using the API, see **Retrieving the DSID Using the API**.
- Using a browser, see **Retrieving the DSID Using a Browser**.

Retrieving the DSID Using the API

You can use the following code to get a DSID token to use across all API calls:

```bash
def login(url,username,password): tenant_url = url return_dict = {'status': 0} global user_session, dsidlogin_URL = tenant_url + '
user_session = requests.session() r = user_session.get(url=login_URL, verify=False) dssignin = user_session.cookies.get('DSSIGNIN'
except IndexError: print 'Error: unable to get xsauth value' data = {'FormDataStr': formdatastr, 'xsauth': xsauth, 'btnContinue':
```

After the DSID is set in the cookies (refer to code `dsid = user_session.cookies.get('DSID')`) use that session for all other API Calls.

You can use the following CURL command format uses the DSID to query the REST API server:

```bash
curl -v --cookie "DSID=<value>" <api_request_url>
```

The following code demonstrates how to get a list of tenants using the API. It updates the cookie information for DSID from the above code.

```bash
def get_tenants(): input_payload = {"type": "application"}request_uri = host_url + api_version + "msp/tenants" output = requests.g
```

Retrieving the DSID Using a Browser

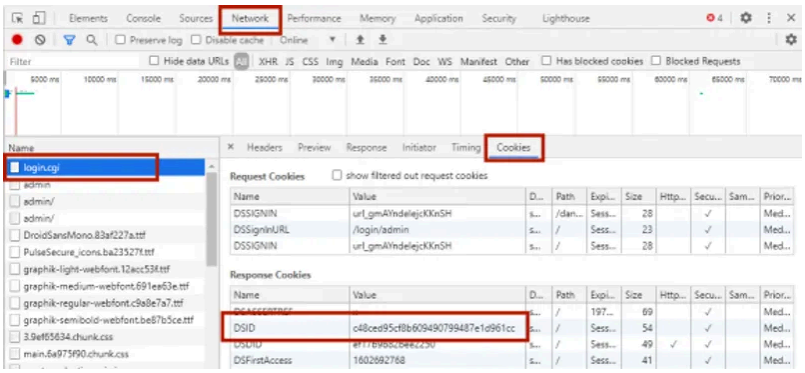
You can use a browser to access the DSID. The procedure below describes the process for the *Chrome* browser, but any browser that offers similar tools can also be used.

Copyright © 2026, Ivanti, Inc. All rights reserved.

1. Log in to the MSP Portal interface in the *Chrome* browser.
   The home page appears.
2. Right-click the main map, and select **Inspect** from the context menu.
   The screen divides horizontally and the element view appears to the right of the screen.

> In the *Edge* browser, you click the **...** control, and then click **Other Tools > Developer Tools**. In the *Firefox* browser, you right-click and select **Inspect Element** from the context menu.

1. In the element view, select the **Network** tab.
2. Select an element from the list of elements (on the left of the tab). For example, *login*, *admin* or *subscriptions*.
   A tab bar appears (on the right of the tab).
3. Select the **Cookies** tab.
   A list of cookies for the page appears.
4. For the *DSID* entry, copy the DSID value and retain this value for future use. For example:



You can use the following CURL command format uses the DSID to query the REST API server:

```bash
curl -v --cookie "DSID=<value>" <api_request_url>
```


Copyright © 2026, Ivanti, Inc. All rights reserved.

3. API Calls

This chapter describes the MSP REST entities and the API calls that can be made to them.

For all API calls, the following CURL command format uses the DSID cookie to query the REST API server:

```bash
curl -v --cookie "DSID=<value>" <api_request_url>
```

MSP API Schemas

The *msp* API contains the following schema entities:

- TenantAdminUserEntity, see **TenantAdminUserEntity**.
- SubscriptionEntity, see **SubscriptionEntity**.
- TenantCreateEntity, see **TenantCreateEntity**.
- TenantUpdateEntity, see **TenantUpdateEntity**.
- TenantEntity, see **TenantEntity**.
- TenantCollectionEntity, see **TenantCollectionEntity**.
- MspTenantUsageReportEntity, see **MspTenantUsageReportEntity**.
- UsageReportCollectionEntity, see **UsageReportCollectionEntity**.
- DeleteErrorReportEntity, see **DeleteErrorReportEntity**.
- DeleteErrorReportCollectionEntity, see **DeleteErrorReportCollectionEntity**.
- UsageReportDownloadURL, see **UsageReportDownloadURL**.
- UsageReportScheduleEntity, see **UsageReportScheduleEntity**.

TenantAdminUserEntity Schema

This schema contains the following structure and properties:

```bash
{ username string First admin username of the tenant email string($email) Admin login email address }
```

SubscriptionEntity Schema

This schema contains the following structure and properties:

Copyright © 2026, Ivanti, Inc. All rights reserved.

Bash

```
{ id string($uuid) Unique identifier of the subscription
display_name string
The display name for the subscription category string Enum:
[ ZTA, PCS ] total_users integer Maximum users count for the subscription display_category string The product category displayed i
}
```

TenantCreateEntity Schema

This schema contains the following structure and properties:

Bash

```
{ sub_domain string
Subdomain name of the tenant user $ref TenantAdminUserEntity The admin user instance for the tenant name string Tenant nam createc
string($datetime)Datetime of Tenant creation subscriptions array
[ SubscriptionEntity ] A list of subscription entitlements for the tenant }
```

- See also: TenantAdminUserEntity Schema
- See also: SubscriptionEntity Schema

TenantUpdateEntity Schema

This schema contains the following structure and properties:

Bash

```
{name string
Tenant name
subscriptions array [ SubscriptionEntity ]
A list of subscription entitlements for the tenant
}
```

- See also: SubscriptionEntity Schema

TenantEntity Schema

This schema contains the following structure and properties:

Bash

```
{ sub_domain string Subdomain name of the tenant user $ref TenantAdminUserEntity The admin user instance for the tenant name strir
Date-time of Tenant creation subscriptions array SubscriptionEntity ] A list of subscription entitlements for the tenant login_url
Login URL of the Tenant Admin Portal
status string
Status of the tenant
}
```


Copyright © 2026, Ivanti, Inc. All rights reserved.

- See also: TenantAdminUserEntity Schema
- See also: SubscriptionEntity Schema

TenantCollectionEntity Schema

This schema contains the following structure and properties:

```
{ total integer Number of MSP owned tenants tenants array
[ TenantEntity ] A list of tenants users { ZTA integer The active users count for the ZTA product PCS integer The active users cou
} tenant_stats {active integer The number of active tenants blocked integer The number of blocked tenants initializing integer The
The maximum number of tenants allowed for this MSP }
```

- See also: TenantEntity Schema

MspTenantUsageReportEntity Schema

This schema contains the following structure and properties:

```
{ id string($uuid)
Unique identifier of the Usage Report
file_name string
Usage Report file name created string($datetime)
Date-time of Usage Report creation}
```

UsageReportCollectionEntity Schema

This schema contains the following structure and properties:

```
{
total integer
Total number of Tenant Usage Reports items array MspTenantUsageReportEntity ] A list of Tenant Usage Reports }
```

See also: MspTenantUsageReportEntity Schema

DeleteErrorReportEntity Schema

This schema contains the following structure and properties:

```
{ id string($uuid) Unique identifier of the Usage Report file_name string
Usage Report file name
}
```


Copyright © 2026, Ivanti, Inc. All rights reserved.

DeleteErrorReportCollectionEntity Schema

This schema contains the following structure and properties:

```
Bash

{ message string
List of file name which are failed to be deleted total integer
Total number of Tenant Usage Reports failed to be deleted error array [ DeleteErrorReportEntity ]
A list of reports which are failed to be deleted }
```

- See also: DeleteErrorReportEntity Schema

UsageReportDownloadURL Schema

This schema contains the following structure and properties:

```
Bash

{ url string($url)
The Usage Report download URL }
```

UsageReportScheduleEntity Schema

This schema contains the following structure and properties:

```
Bash

{ frequency string ( default: daily ) Frequency of the schedule (one of 'daily', 'weekly', 'monthly') start_day integer ( default:
2. start_day must be between 0 (Mon) and 6 (Sun) if frequency is 'weekly' 3. start_day must be between 1 and 28 if frequency is 'm
booleanDisable the schedule
}
```

MSP API Methods

The *msp* API supports the following activities/methods:

- Adding a tenant, see Adding a tenant.
- Retrieving all tenants, see Retrieving all tenants.
- Requesting an increase to the maximum tenant cap, see Creating a Request to Increase the Tenant Limit.
- Updating a tenant's details, see Editing a Tenant Configuration.
- Generating a new usage summary report, see Generating a new usage summary report.
- Retrieving the list of usage summary reports, see Retrieving the list of usage summary reports.
- Deleting a usage summary report, see Deleting a usage summary report.
- Obtaining the URL to download a usage summary report, see Obtaining the URL to download a usage summary report.
- Creating a new usage summary report schedule, see Creating a new usage summary report schedule.
- Retrieving the usage summary report schedule, see Retrieving the usage summary report schedule.

Copyright © 2026, Ivanti, Inc. All rights reserved.

Adding a Tenant

To add a new tenant, use the following REST API call:

- **Method:** POST /api/msp/tenants
- **Resource:** Path
- **JSON Data:** JSON dictionary representing a new **TenantCreateEntity Schema** entity.

> You can only continue to use this method while the total number of tenants for this MSP is below the maximum tenant limit. If you hit the limit, consider creating a request to increase the limit (see).

If processed correctly, a HTTP 202 response is returned. Otherwise, a JSON body containing an error is returned.

Request

The following is an example request:

Bash

```
Authorization: Content-Type: application/json
Request Body
{
"sub_domain": "string", "user": { "username": "string", "email": "user@example.com" },
"name": "string",
"created": "string", "subscriptions": [
{ "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "display_name": "string",
"category": "ZTA",
"total_users": 0,
"display_category": "string"}]
}
```

Response

Bash

```
HTTP/1.1 202 OK
```

Retrieving a list of Tenants

To retrieve the list of tenant configurations, use the following REST API call:

- **Method:** GET /api/msp/tenants
- **Resource:** Path

If processed correctly, a JSON body containing a **TenantCollectionEntity Schema** entity is returned. Otherwise, a JSON body containing an error is returned.

Parameters

Copyright © 2026, Ivanti, Inc. All rights reserved.

- `start` : (integer - in: query)
  The number of initial results that should be skipped. (Default 0)
- `limit` : (integer - in: query)
  The number of results that should be returned. (Default 20)

Request

The following is an example request:

Bash

```
GET /api/msp/tenants?start=0&limit=20 Authorization:
Content-Type: application/json
```

Response

The following is an example response:

Bash

```
HTTP/1.1 200 OK
Content-Type: application/json
Response Body{ "total": 0, "tenants": [ {
"sub_domain": "string", "user": { "username": "string", "email": "user@example.com" }, "name": "string", "created": "string", "sub
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "display_name": "string", "category": "ZTA", "total_users": 0,
"display_category": "string"
}
], "login_url": "string",
"status": "string" }
],
"users": { "ZTA": 0, "PCS": 0
}, "tenant_stats": { "active": 0,
"blocked": 0,
"initializing": 0 },
"tenant_cap_limit": 0
}
```

Creating a Request to Increase the Tenant Limit

To create a request to increase the maximum tenant creation limit, use the following REST API call:

- **Method:** POST /api/msp/tenants/requests/tenant-cap-limit
- **Resource:** Path

If processed correctly, a JSON body containing the request ID is returned. Otherwise, a JSON body containing an error is returned.

Request

The following is an example request:

Copyright © 2026, Ivanti, Inc. All rights reserved.

```
Bash

POST /api/msp/tenants/requests/tenant-cap-limit Authorization:
Content-Type: application/json
```

Response

The following is an example response:

```
Bash

HTTP/1.1 200 OK Content-Type: application/json
Response Body { "id": "abcdef1ad9424389b788d97337c894a6"
}
```

Editing a Tenant Configuration

To edit a tenant's configuration, use the following REST API call:

- **Method:** PUT /api/msp/tenants/
- **Resource:** Path
- **JSON Data:** JSON dictionary representing a **TenantUpdateEntity Schema** entity.

If processed correctly, a JSON body containing an updated instance of a **TenantUpdateEntity Schema** is returned. Otherwise, a JSON body containing an error is returned.

Parameters

- `tenant_id` : (string UUID - in: path)
  The ID of the MSP-owned tenant.

Request

The following is an example request:

```
Bash

PUT /api/msp/tenants/627df11ad9424389b788d97337c894a6
Authorization: Content-Type: application/json Request Body { "name": "string", "subscriptions": [ { "id": "3fa85f64-5717-4562-b3fc
} ]
}
```

Response

The following is an example response:

Copyright © 2026, Ivanti, Inc. All rights reserved.

Bash

```
HTTP/1.1 200 OK Content-Type: application/json Response Body
{
"name": "string", "subscriptions": [ {
"id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"display_name": "string", "category": "ZTA", "total_users": 0,
"display_category": "string"
}
]
}
```

Generating a new usage summary report

To generate a new usage summary report for all MSP-owned tenants, use the following REST API call:

- **Method:** POST /api/msp/reports/usage
- **Resource:** Path

If processed correctly, a JSON body containing a **MspTenantUsageReportEntity Schema** entity is returned. Otherwise, a JSON body containing an error is returned.

Request

The following is an example request:

Bash

```
POST /api/msp/reports/usage Authorization:
Content-Type: application/json
```

Response

The following is an example response:

Bash

```
HTTP/1.1 200 OK
Content-Type: application/json Response Body { "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "file_name": "string", "created": "st
```

Retrieving the list of Usage Reports

To retrieve the list of usage reports already generated for this MSP, use the following REST API call:

- **Method:** GET /api/msp/reports/usage
- **Resource:** Path

If processed correctly, a JSON body containing a **UsageReportCollectionEntity Schema** entity is returned. Otherwise, a JSON body containing an error is returned.

Copyright © 2026, Ivanti, Inc. All rights reserved.

Parameters

- `start` : (integer - in: query)
  The number of initial results that should be skipped. (Default 0)
- `limit` : (integer - in: query)
  The number of results that should be returned. (Default 20)

Request

The following is an example request:

```Bash
GET /api/msp/reports/usage?start=0&limit=20 Authorization:
Content-Type: application/json
```

Response

The following is an example response:

```Bash
HTTP/1.1 200 OK Content-Type: application/json Response Body { "total": 0, "items": [ { "id": "3fa85f64-5717-4562-b3fc-2c963f66afa
"file_name": "string", "created": "string" } ]
}
```

Deleting a list of Usage Reports

To delete a collection of usage report entities, use the following REST API call:

- **Method:** DELETE /api/msp/reports/usage
- **Resource:** Path

If processed correctly, a HTTP 204 confirmation is returned. Otherwise, a HTTP 200 response is returned, with a JSON response body containing a **DeleteErrorReportCollectionEntity Schema** entity (the list of report IDs that were not deleted).

Parameters

- `report_ids` : (array[string] - in: query)
  A list of the report IDs to delete

Request

The following is an example request:

```Bash
DELETE /api/msp/reports/usage?report_ids=e274bf3ebe3841a88ade1630515624c6,e362bf3ebe3841a88ade5823915684c2 Authorization:
Content-Type: application/json
```


Copyright © 2026, Ivanti, Inc. All rights reserved.

Response

The following is an example response:

```
Bash

HTTP/1.1 204 Successfully deleted usage reports
Content-Type: application/json
```

Obtaining the URL to Download a Usage Report

To obtain the URL with which you can download a usage report to your local workstation, use the following REST API call:

- **Method:** GET /api/msp/reports/usage//download-url
- **Resource:** Path

If processed correctly, a JSON body containing a **UsageReportDownloadURL Schema** entity is returned. Otherwise, a JSON body containing an error is returned.

Parameters

- `report_id` : (string UUID - in: path)
  The report ID.

Request

The following is an example request:

```
Bash

GET /api/msp/reports/usage/827df11ad9424389b788d97337c894a8/download-url Authorization:
Content-Type: application/json
```

Response

The following is an example response:

```
Bash

HTTP/1.1 200 OK Content-Type: application/json
Response Body
{ "url": "string" }
```

Creating a Usage Summary Report Schedule

To create a usage summary report schedule for all MSP-owned tenants, use the following REST API call:

- **Method:** POST /api/msp/reports/usage/schedule
- **Resource:** Path
- **JSON Data:** JSON dictionary representing a new **UsageReportScheduleEntity Schema** entity.

Copyright © 2026, Ivanti, Inc. All rights reserved.

If processed correctly, a JSON body containing an updated instance of a **UsageReportScheduleEntity Schema** entity is returned. Otherwise, a JSON body containing an error is returned.

Request

The following is an example request:

Response

The following is an example response:

```
Bash
```

```
HTTP/1.1 200 OK Content-Type: application/json
Response Body
{
"frequency": "weekly", "start_day": 3,
"disabled": true
}
```

Retrieving the Usage Report Schedule

To retrieve the usage report schedule for this MSP, use the following REST API call:

- **Method:** GET /api/msp/reports/usage/schedule
- **Resource:** Path

If processed correctly, a JSON body containing the **UsageReportScheduleEntity Schema** entity is returned. Otherwise, a JSON body containing an error is returned.

Request

The following is an example request:

```
Bash
```

```
GET /api/msp/reports/usage/schedule Authorization:
Content-Type: application/json
```

Response

The following is an example response:

```
Bash
```

```
HTTP/1.1 200 OK Content-Type: application/json Response Body {
"frequency": "weekly", "start_day": 3, "disabled": true }
```


Copyright © 2026, Ivanti, Inc. All rights reserved.

4. MSP Portal REST API

Ivanti Neurons for Secure Access (nSA) is a cloud-based SaaS (software as a service) application that provides fully-managed zero-trust authentication and access control for an organization’s application infrastructure. enables administrators to define end-to-end authorization and authentication policies that control application visibility, access, and security for all users and their devices.MSP Portal REST API

View as PDF

Copyright © 2026, Ivanti, Inc. All rights reserved.