



API Reference

1. Archbee Public API

1.1. Overview

1.2. Open Api

1.2.1. Sync Open Api document

1.2.2. Info Open Api document

1.3. Doc

1.3.1. Get document

1.3.2. Update / Create document

1.3.3. Delete document

1.3.4. Search document

1.3.5. Import Content

1.4. DocSpace

1.4.1. Create Space

1.4.2. Update Space

1.4.3. Publish Space

1.4.4. Clone Space

1.5. Organization

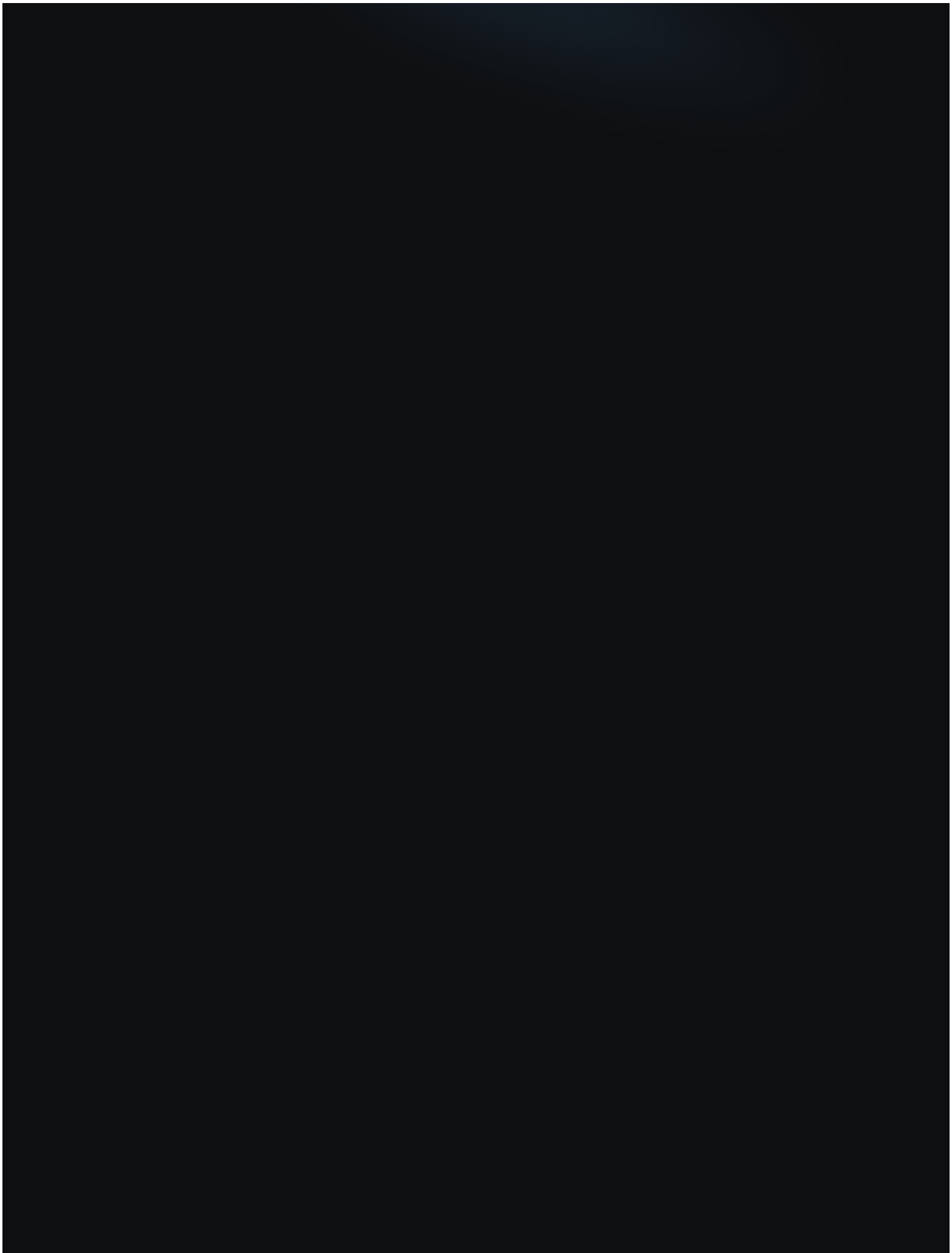
1.5.1. Organization Export

1.6. Download OpenAPI file

1.7. Models

1.7.1. Model: DocResponse

1.7.2. Model: Response400



1. Archbee Public API

1.1. Overview

Welcome to the documentation for our Application Programming Interface (API). Our API is designed to provide developers with seamless access to our services, enabling the creation of powerful and innovative applications.



Get document



**Update / Create
document**



Delete document



Search document



Import Content

This documentation will guide you through the various functionalities, endpoints, and best practices associated with our API, ensuring that you have all the resources needed to integrate it effectively into your projects.

With comprehensive support, detailed examples, and robust security measures, our API aims to facilitate a smooth development experience while empowering you to harness the full potential of our platform.

We are committed to providing you with a reliable and efficient service. Thank you for choosing our API, and we look forward to supporting you in your development journey.

1.2. Open Api

1.2.1. Sync Open Api document

POST

https://app.archbee.com/api/public-api/sync-api-reference

Try it ▶

Archbee App Server

Sync Open Api document with a new or existing Open Api tree

Credentials

HTTP Bearer bearerAuth

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> Content-Type `string` *

multipart/form-data

Form Parameters

> docTreeId `string`

21-character_string0

> file `file` *

Choose file

> type `string<openapi | postman>`

Select... ▼

> openApiTryIt `string<true | false>`

Select... ▼

> showDownloadOpenApiFile `string<true | false>`

Select... ▼

- > **shouldCreateSchemaCategory** `string<true | fals...`
- > **shouldCreateIntro** `string<true | false>`
- > **openApiLanguageExamples** `list`

Select...



Select...



Responses

- ▼ ● **200** Import OK status

▼ **object**

Description

Import OK status

- > **status** `string<OK>`
- > **data** `object`

- ▼ ● **400** Invalid request

▼ **Response400**

- > **status** `string<OK | Not OK>`
- > **messages** `string[]`

```
curl --request POST \  
  --url https://app.archbee.com/api/public-api/sync-api-reference \  
  --header 'accept: application/json' \  
  --header 'content-type: multipart/form-data' \  
  --form 'docTreeId="21-character__string0"' \  
  --form 'file="@swagger.json"' \  
  --form 'type="openapi"' \  
  --form 'openApiTryIt="true"' \  
  --form 'showDownloadOpenApiFile="true"' \  
  --form 'shouldCreateSchemaCategory="true"' \  
  --form 'shouldCreateIntro="true"'
```

```
// Import OK status
{
  "status": "OK",
  "data": {
    "docTreeId": "21-character__string0"
  }
}
```

1.2.2. Info Open Api document

GET

https://app.archbee.com/api/public-api/info-api-reference

Try it ▶

Archbee App Server

Get info of an existing Open Api tree

Credentials

HTTP Bearer bearerAuth

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Form Parameters

> **docTreeId** `string`

Responses

▼ ● 200 Status

▼ `object`

Description

Status

> **status** `string<OK>`

> **data** `object`

▼ ● 400 Invalid request

▼ `Response400`

> **status** `string<OK | Not OK>`

> **messages** `string[]`

```
curl --request GET \  
  --url https://app.archbee.com/api/public-api/info-api-reference \  
  --header 'accept: application/json'
```

```
// Status  
{  
  "status": "OK",  
  "data": {  
    "result": "",  
    "importedContent": ""  
  }  
}
```

1.3. Doc

1.3.1. Get document

GET

https://app.archbee.com/api/public-api/doc

Try it ▶

Archbee App Server

Retrieve a document in `markdown`, `html`, `json`, or `source` format.

Credentials

HTTP Bearer `bearerAuth`

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> **Content-Type** `string` *

Body Parameters

▼ **body** `object` *

> **docId** `string` *

> **format** `string` `<markdown | html | json | so...`

Select...

Responses

▼ ● **200** Successful response containing the document content.



object

> content string

> format string



● 400 Invalid request parameters.



Response400

> status string<OK | Not OK>

> messages string[]

```
curl --request GET \  
  --url https://app.archbee.com/api/public-api/doc \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "docId": "21-character__string0"  
  }'
```

```
// Successful response containing the document content.  
{  
  "content": "",  
  "format": ""  
}
```

1.3.2. Update / Create document

POST

https://app.archbee.com/api/public-api/doc

Try it ▶

Archbee App Server

Create / Update Doc by `docId`

Credentials

HTTP Bearer `bearerAuth`

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> **Content-Type** `string` *

Body Parameters

▼ **body** `object` *

> **title** `string`

> **description** `string`

> **previewImgURL** `string`

> **slug** `string`

> **alias** `string`

> **content** `string` *

> **format** `string`<markdown | json>

> **docId** `string`

> **parentDocId** `string`

Select... ▼

Responses

▼ ● 200 Process status

▼ object

Description

Process status

> **status** `string<OK>`

> **data** `object`

▼ ● 400 Invalid request

▼ Response400

> **status** `string<OK | Not OK>`

> **messages** `string[]`

```
curl --request POST \  
  --url https://app.archbee.com/api/public-api/doc \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "content": "# this is a h1 title\nAnd this is a paragraph\n- list item 1\n- list item 2\n}'
```

```
// Process status  
{  
  "status": "",  
  "data": {  
    "docId": "",  
    "newRecord": false  
  }  
}
```

1.3.3. Delete document

DELETE

https://app.archbee.com/api/public-api/doc

Try it ▶

Archbee App Server

Delete document by `docId`

Credentials

HTTP Bearer bearerAuth

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> **Content-Type** `string` *

Body Parameters

▼ **body** `object` *

> **docId** `string` *

Responses

▼ ● **200** Delete status

▼ `object`

Description

Delete status

> **status** `string<OK>`

> **data** `object`

▼ ● **400** Invalid request



Response400

- > **status** string<OK | Not OK>
- > **messages** string[]

```
curl --request DELETE \  
  --url https://app.archbee.com/api/public-api/doc \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "docId": "21-character__string0"  
  }'
```

```
// Delete status  
{  
  "status": "",  
  "data": {  
    "status": false  
  }  
}
```

1.3.4. Search document

GET

https://app.archbee.com/api/public-api/docs/search

Try it ▶

Archbee App Server

Search Archbee documents in docSpace. Can perform one of the 3 types of search:

- **ai-chat** to return ai generative answer accompanied by source docs;
- **ai-retrieval** to return just similar docs with the query;
- **words** to perform normal search aka. "word-based"; use empty query to return all docs
- **single-doc** to return document info by id

Credentials

HTTP Bearer bearerAuth

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> Content-Type `string` *

Body Parameters

▼

body object *

>

query string *

>

searchOnlyTitle boolean

>

persistSearch boolean

>

searchSessionId string

>

docId string

>

dataTextFormat string<markdown | html>

Select... ▼

>

parentDocId string

>

type string<words | ai-chat | ai-retrieva...

Select... ▼

Responses

▼

● 200

Process status

▼

object

Description

Process status

>

status string<OK>

>

data object

▼

● 400

Invalid request

▼

Response400

>

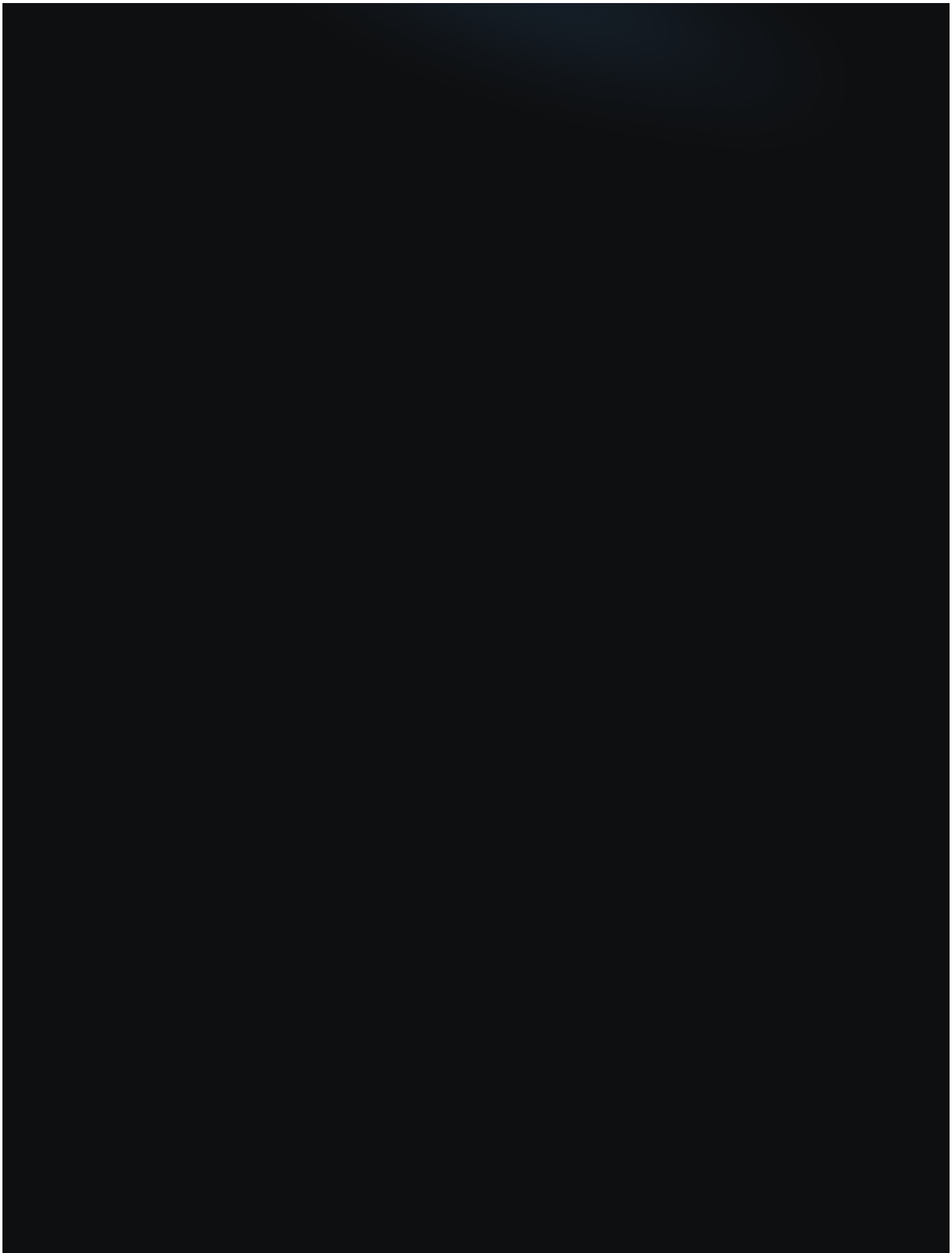
status string<OK | Not OK>

>

messages string[]

```
curl --request GET \  
  --url https://app.archbee.com/api/public-api/docs/search \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "query": ""  
  }'
```

```
// Process status  
{  
  "status": "",  
  "data": {  
    "searchSessionId": "",  
    "docs": [  
      {  
        "id": "",  
        "name": "",  
        "urlKey": "",  
        "privacy": "",  
        "highlight": {}  
      }  
    ]  
  }  
}
```



1.3.5. Import Content

POST

https://app.archbee.com/api/public-api/import-content

Try it ▶

Archbee App Server

Create new Doc from imported markdown file. In case of zip file, create a new docTree and keeps archived tree structure.

Credentials

HTTP Bearer bearerAuth

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> **Content-Type** `string` *

multipart/form-data

Form Parameters

> **file** `file` *

Choose file

> **type** `string<markdown>`

markdown

Responses

▼ ● 200 Import OK status

▼ `object`

Description

Import OK status

> **status** `string<OK>`

> **data** `object`

▼ ● 400 Invalid request



Response400

> **status** string<OK | Not OK>

> **messages** string[]

```
curl --request POST \  
  --url https://app.archbee.com/api/public-api/import-content \  
  --header 'accept: application/json' \  
  --header 'content-type: multipart/form-data' \  
  --form 'file="@markdowns.zip"' \  
  --form 'type="markdown"'
```

```
// Import OK status  
{  
  "status": "OK",  
  "data": {  
    "docId": "21-character__string0"  
  }  
}
```

1.4. DocSpace

1.4.1. Create Space

POST

https://app.archbee.com/api/public-api/space/create

Try it ▶

Archbee App Server

Create a new space with desired fields, return its id. New space will inherit apiKey.

Credentials

HTTP Bearer bearerAuth

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> Content-Type `string` *

Body Parameters

▼

body object *

> name string

> isLlmEnabled boolean

> isReviewSystemEnabled boolean

> docSpaceGroupId string

Responses

▼

● 200 Process status

▼

object

Description

Process status

> status string<OK>

> data object

▼

● 400 Invalid request

▼

Response400

> status string<OK | Not OK>

> messages string[]

```
curl --request POST \
  --url https://app.archbee.com/api/public-api/space/create \
  --header 'accept: application/json'
```

```
// Process status
{
  "status": "",
  "data": {
    "newDocSpaceId": ""
  }
}
```

1.4.2. Update Space

POST

https://app.archbee.com/api/public-api/space/update

Try it ▶

Archbee App Server

Update space fields.

Credentials

HTTP Bearer bearerAuth

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> **Content-Type** `string` *

Body Parameters

▼ **body** `object` *

> **publicAccessControlPart** `object`

> **hostnamePart** `object`

> **spaceLinks** `object[]`

Responses

▼ ● **200** Process status

▼ `object`

Description

Process status

> **status** `string<OK>`

> **data** `object`

▼ ● **400** Invalid request



Response400

- > **status** string<OK | Not OK>
- > **messages** string[]

```
curl --request POST \  
  --url https://app.archbee.com/api/public-api/space/update \  
  --header 'accept: application/json'
```

```
// Process status  
{  
  "status": "",  
  "data": {  
    "id": ""  
  }  
}
```

1.4.3. Publish Space

POST

https://app.archbee.com/api/public-api/space/publish

Try it ▶

Archbee App Server

Publish Archbee documents from given `docSpaceId`

Credentials

HTTP Bearer bearerAuth

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> **Content-Type** `string` *

Body Parameters

▼ **body** `object` *

> **environment** `string` *

Responses

▼ ● **200** Process status

▼ `object`

Description

Process status

> **status** `string<OK>`

> **data** `object`

▼ ● **400** Invalid request



Response400

> **status** string<OK | Not OK>

> **messages** string[]

```
curl --request POST \  
  --url https://app.archbee.com/api/public-api/space/publish \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "environment": ""  
  }'
```

```
// Process status  
{  
  "status": "",  
  "data": {  
    "id": ""  
  }  
}
```

1.4.4. Clone Space

POST

https://app.archbee.com/api/public-api/space/clone

Try it ▶

Archbee App Server

Clone Archbee space from given docSpaceId

Credentials

HTTP Bearer bearerAuth

Requires the Authorization header in the form Authorization: Bearer {base64(docSpaceId~apiKey)} Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> Content-Type string *

Body Parameters

body object *

> targetSpaceGroupId string

Responses

200 Process status

object

Description

Process status

> status string<OK>

> data object

400 Invalid request



Response400

> **status** string<OK | Not OK>

> **messages** string[]

```
curl --request POST \  
  --url https://app.archbee.com/api/public-api/space/clone \  
  --header 'accept: application/json'
```

```
// Process status  
{  
  "status": "",  
  "data": {  
    "id": ""  
  }  
}
```

1.5. Organization

1.5.1. Organization Export

GET

https://app.archbee.com/api/public-api/team/export

Try it ▶

Archbee App Server

Export assets, spaces with documents, team and user data from Archbee given `teamId`.
Use `--output <export_file_path>`

Credentials

HTTP Bearer bearerAuth

Requires the `Authorization` header in the form `Authorization: Bearer {base64(docSpaceId~apiKey)}` Where the token is base64 encoding of docSpaceId, tilde, and apiKey.

Enter http bearer value

Header Parameters

> Content-Type `string` *

Body Parameters

▼ body `object` *

> teamId `string` *

> exportThisSpaceOnly `boolean`

> exportAsLink `boolean`

Responses

● **200** Successful export

▼ ● **400** Invalid request

▼ Response400

> **status** string<OK | Not OK>

> **messages** string[]

```
curl --request GET \  
  --url https://app.archbee.com/api/public-api/team/export \  
  --header 'accept: application/json' \  
  --header 'content-type: application/json' \  
  --data '{  
    "teamId": "21-character__string0"  
  }'
```

```
// Successful export
```

1.6. Download OpenAPI file

1.7. Models

1.7.1. Model: DocResponse

Model Parameters



DocResponse

- > **id** `string`
- > **name** `string`
- > **urlKey** `string`
- > **privacy** `string<private | shared with team...`
- > **highlight** `object`

```
{
  "id": "",
  "name": "",
  "urlKey": "",
  "privacy": "",
  "highlight": {}
}
```

1.7.2. Model: Response400

Model Parameters



Response400

- > **status** `string<OK | Not OK>`
- > **messages** `string[]`

```
{
  "status": "",
  "messages": [
    ""
  ]
}
```