



v3.0

CONTENTS

1. Archbee Public API	3
1.1. 概述	3
1.1.1. 获取订单	4
1.2. Open Api	6
1.2.1. 同步开放 API 文档	6
1.2.2. 信息开放API文档	8
1.3. Doc	10
1.3.1. 获取文档	10
1.3.2. 更新 / 创建文档	12
1.3.3. 删除文档	14
1.3.4. 搜索文档	16
1.3.5. 导入内容	18
1.4. DocSpace	20
1.4.1. 创建空间	20
1.4.2. 更新空间	22
1.4.3. 发布空间	24
1.4.4. 克隆空间	26
1.5. Organization	28
1.5.1. 组织导出	28
1.6. Download OpenAPI file	
1.7. Models	30
1.7.1. 模型：文档响应	30
1.7.2. 模型：响应400	31

1. Archbee Public API

1.1. 概述

欢迎阅读我们应用程序编程接口（API）的文档。我们的 API 旨在为开发者提供无缝访问我们的服务的能力，从而构建功能强大且创新的应用程序。

选项 1	选项 2	



获取文档

Will be generated on publish for the public website from document's SEO Meta Controls.



更新 / 创建文档

Will be generated on publish for the public website from document's SEO Meta Controls.



删除文档

Will be generated on publish for the public website from document's SEO Meta Controls.



搜索文档

Will be generated on publish for the public website from document's SEO Meta Controls.



导入内容

Will be generated on publish for the public website from document's SEO Meta Controls.

本文件将引导您了解我们 API 的各种功能、端点以及最佳实践，确保您拥有将其有效集成到项目中所需的全部资源。

通过全面的支持、详细的示例和强大的安全措施，我们的 API 致力于为您提供顺畅的开发体验，同时帮助您充分发挥平台的全部潜力。

我们致力于为您提供可靠且高效的服务。感谢您选择我们的 API，期待在您的开发旅程中持续为您提供支持。

1.1.1. 获取订单

概述

此端点可帮助您获取所有已下的订单。

Get Cakes

Get a cake by its ID

GET

https://api.cakes.com

Request

BODY PARAMETERS

id	String	required
要获取的蛋糕的ID		
Parameter name	String	optional
Parameter name	String	optional

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",
}
```

1.2. Open Api

1.2.1. 同步开放 API 文档

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

> docTreeld 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 | false>

Select... ▾

> shouldCreateIntro string<true | false>

Select... ▾

> openApiLanguageExamples list

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[]

Request



```
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"'
```

Responses

● 200

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

1.2.2. 信息开放API文档

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

> docTreeld string

Responses

- ▼

● 200

Status

▼

object

Description

Status

>

status string<OK>

>

data object
- ▼

● 400

Invalid request

▼

Response400

>

status string<OK | Not OK>

>

messages string[]

Request

Curl

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

Responses

200

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

1.3. Doc

1.3.1. 获取文档

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 | source>

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`[]

Request



```
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"  
  }'
```

Responses



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

1.3.2. 更新 / 创建文档

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>

Select... ▼

> docId string

> parentDocId 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[]
```

Request



```
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"
  }'
```

Responses

 200

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

1.3.3. 删除文档

DELETE

Try it ▶

Archbee App Server

Delete document by

Credentials

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

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**[]

Request

 Curl

```
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"
  }'
```

Responses

 200

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

1.3.4. 搜索文档

GEThttps://app.archbee.com/api/public-api/docs/searchTry 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 BearerbearerAuth

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-retrieval>

Select...▼

Responses

▼ ● 200 Process status

▼ object

DescriptionProcess status

> status string<OK>

> data object

▼ ● 400 Invalid request

16 / 31

Created at Oct 01, 2026

 Response400

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

> messages string[]
```

Request



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

Responses

 200

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

1.3.5. 导入内容

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.

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[]`

Request

 Curl

```
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"'
```

Responses

● 200

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

1.4. DocSpace

1.4.1. 创建空间

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[]

Request



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

Responses



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

1.4.2. 更新空间

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[]

Request

Curl

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

22 / 31

Created at Oct 01, 2026

Responses

 200

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

1.4.3. 发布空间

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[]

Request

Curl

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

24 / 31

Created at Oct 01, 2026

Responses

 200

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

1.4.4. 克隆空间

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[]`

Request

 Curl

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

Responses

 200

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

1.5. Organization

1.5.1. 组织导出

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.

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`[]

Request

Curl

```
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"  
  }'
```

Responses

● 200

// Successful export

1.7. Models

1.7.1. 模型：文档响应

Model Parameters



DocResponse

- > id string
- > name string
- > urlKey string
- > privacy string<private | shared with team | public-via-link | public>
- > highlight object

Request

JSON

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

1.7.2. 模型：响应400

Model Parameters

- ▼ Response400
- > status string<OK | Not OK>

> messages string[]

Request

JSON

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