PLEX
DOCUMENTATION

Plex APIs

Copyright Notice

Note: This document is provided for print purposes only. Once printed, please compare the last updated date on the cover page with the Plex Knowledge Center to ensure you have the latest content.

Contents

Plex Developer Portal

Use the **Plex Developer Portal** at https://developers.plex.com/ to access the API documentation. To use the Developer Portal, complete the following steps:

1. Request a developer account	Submit a case to Plex Service Cloud to request one or more developer accounts. Include the email address that you want associated with the account. Plex will create your account and notify you via email.
2. Access the Developer Portal	Log in to developers.plex.com using the link from the account confirmation email.
3. Obtain API credentials	In the Developer Portal, create an App to generate your API keys.
4. Create your apps	Perform a test call to confirm your account access. After confirmation, you're ready to use the API. Update your case in the Service Cloud to confirm that you were able to access the API.

Getting Started with the Plex Developer Portal

Request a Plex API Developer Account

A Plex Developer Account is required to access the Developer Portal and call the API.

1. To request developer accounts, your Plex Champion must submit a case to the Plex support **Service Cloud.** Include a name and email address for each account that you need created.
2. Plex will review your request and send a confirmation email to the email account(s) that you requested.
3. You will be notified via email when your account as been created. Follow the instructions in the email to log in to the **Developer Portal**.
4. After you have successfully accessed the **Developer Portal**, you must create an App in the portal to get your API keys.
5. See Obtain API credentials for next steps.

Obtain API credentials

To obtain your API credentials (Consumer Key and Consumer Secret), create an application in the Plex Developer Portal.

1. Access the Plex Developer Portal at developers.plex.com.
2. Click on your account name and select **My Apps.**
3. Click **+Add a new App**.
4. On the **Add App** screen, type a name for your App.
5. For **Product**, select the API products to which you want access.
6. Click **Create App**.

The Consumer Key and Consumer Secret is generated for your application and is displayed on the screen.

Call the API

If you have a Plex API developer account, you can test Plex APIs directly within the Plex Developer Portal.

1. In the **Developer Portal**, go to the APIs page.
2. Click the API operation that you would like to test.
3. Click a resource to view its details, including the resource URL and the request body.
4. After the Request Body, in the **API Key** field, click **Set**.

    a. For Name, type **X-Plex-Connect-Api-Key**.
    b. For Value, type your Customer Key. (You can retrieve this from the App that you created. See Obtain API credentials.)
    c. Select **Header.**
    d. Click **Ok**.

5. Click **Send this request.**
6. The API request is sent. The page will update with the response.
7. If the request is unsuccessful, see Error Codes for additional information.

Error Codes

400 - Malformed request

400 errors generally indicate that the body of the request does not match the resource being requested.

400 - Validation failure

One or more of the parameters use an incorrect format or are omitted from the request.

401 - Request not authenticated

A 401 error can occur when you try to access the system using an expired api key, an invalid api key, or without an api key at all.

If you receive this error, verify that the X-Plex-Connect-Api-Key header is specified on the request and that the key matches the value in the Plex Developer Portal.

404 - Resource not found

A 404 error can occur when your application attempts to access Plex functionality using an incorrect request URL or if no data exists to return.

500 - Request processing error

500 errors are unexpected. If you can reproduce the error, submit a support ticket to Plex. Include the steps to duplicate the issue, but do not include client secrets, passwords, or subscription keys.

Making API Calls

Request Headers

Subscription Key header

All Plex API requests require a valid subscription key header. The request header `X-Plex-Connect-Api-Key` is a subscription key which provides access to the API. You can obtain your API key through the Plex Developer Portal.

If you do not include the subscription key header in an API request, or provide an invalid subscription key, you will receive the following response:

```
{    "code": "REQUEST_NOT_AUTHENTICATED",
    "message": "The request could not be authenticated."
}
```

Tenant ID header

Plex API requests are performed in the context of a single default tenant (also called an entity or PCN). The request header `X-Plex-Connect-Tenant-Id` can be used to execute the request for a tenant other than the default one. The header value to use must be a tenant ID.

Customer ID header

Plex API requests are performed in the context of a single default tenant (also called an entity or PCN). The request header `X-Plex-Connect-Customer-Id` can be used to execute the request for a different tenant other than the default one. The header value to use must be a PCN value.

If you do not have access to the requested tenant you will receive the following response:

```
{    "code": "REQUEST_NOT_AUTHENTICATED",
    "message": "The request could not be authenticated."
}
```

URL Structure

Managed endpoints have the following form:

```
https://[tier.]connect.plex.com/{collection}/{version}/{resource}
[/{resource id}]?[filters]
```

Example: `https://connect.plex.com/mdm/v1/employees?lastName=smith`

UTC time

Plex API responses return all dates in Coordinated Universal Time (UTC). Client applications must take this into account.

List Query String Parameters

Some APIs have methods that can accept a list for a query string parameter.

To use multiple values for a list query string parameter, construct the URL with multiple instances of the query string parameter and its value pair.

For example, for a query string parameter `Id` that can accept a list of values, the URL for the request could look similar to this example:

```
https://<base URL>?Id=value1&Id=value2&Id=value3.
```

In this example, three different values are provided for the `Id`query string parameter, and the request will use the three parameters that are listed.

Dates and Times

Coordinated Universal Time (UTC)

Use UTC or UTC+Offset standard data and time formats.

Format String

Use the following format for any date or time input parameters:

```
YYYY-MM-DDThh:mm:ss.fffffffZ
```

Symbols

```
YYYY      Four-digit year (required)
MM        Two-digit month (01 through 12, required)
DD        Two-digit day of month (01 through 28/29/30/31, required)
T         Constant char separator between date and time components
(required)
hh        Two-digit hour (00 through 23, required)
mm        Two-digit minute (00 through 59, required)
ss        Two-digit second (00 through 59, optional)
fffffff   Fractional digits of second (0 to 7 digits, optional)
Z         Constant char designation for UTC a.k.a. "Zulu Time"
(required)
```

The follow also applies:

- The *seconds* component may be excluded.
- Fractional digits of seconds may be excluded, but when included may not exceed 7 digits. This applies

to both inputs and outputs.

- Trailing zeros are allowed, but are not required

The following are not supported:

- Dates without times
- Times without a time zone
- Fractional digits for anything except seconds
- Using a blank space instead of the 'T' character for date time separator

Range of accuracy

```
1999-12-31T12:34Z
1999-12-31T12:34:56Z
1999-12-31T12:34:56.1Z
1999-12-31T12:34:56.12Z
1999-12-31T12:34:56.123Z
1999-12-31T12:34:56.1234Z
1999-12-31T12:34:56.12345Z
1999-12-31T12:34:56.123456Z
1999-12-31T12:34:56.1234567Z
```