

Google Calendar API Provider Setup

Last Modified on 12/30/2019 4:23 pm EST

To authenticate a Google Calendar connector instance you must create a project with a web application in the Google API console, enable the Google Calendar API, and establish OAuth 2.0 credentials. When you authenticate, use the OAuth 2.0 credentials **Client ID**, **Client secret**, and **Authorized redirect URI** as the **API Key**, **API Secret**, and **Callback URL**.

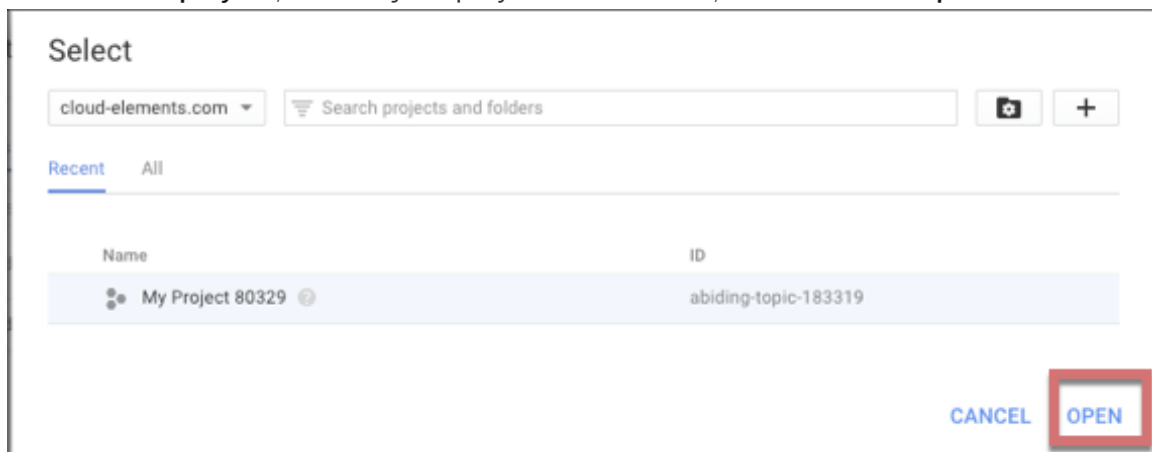
See the latest setup instructions in the [Google API documentation](#).

Locate Credentials for Authentication

If you already created a project and application, follow the steps below to locate the **Client ID**, **Client secret**, and **Authorized redirect URI**. If you have not created an app, see [Create an Application](#).

To find your OAuth 2.0 credentials:

1. Log in to your account at [Google](#).
2. Click **Select a project**, choose your project from the list, and then click **Open**.



Google displays your apps and associated **Client ID**.

3. Click the pencil icon to see the **Client ID**, **Client secret**, and **Authorized redirect URI**.
4. Record the **Client ID** and **Client secret**.

5. Record the **Authorized redirect URI** for your app.

Client ID for Web application

DOWNLOAD JSON RESET SECRET DELETE

Client ID	2574: [redacted]
Client secret	T2 [redacted]
Creation date	Oct 18, 2017, 3:39:59 PM

Name

Awesome Web App

Restrictions

Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

https://www.example.com

Authorized redirect URIs
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

https://httpbin.org/get

https://www.example.com/oauth2callback

Save Cancel

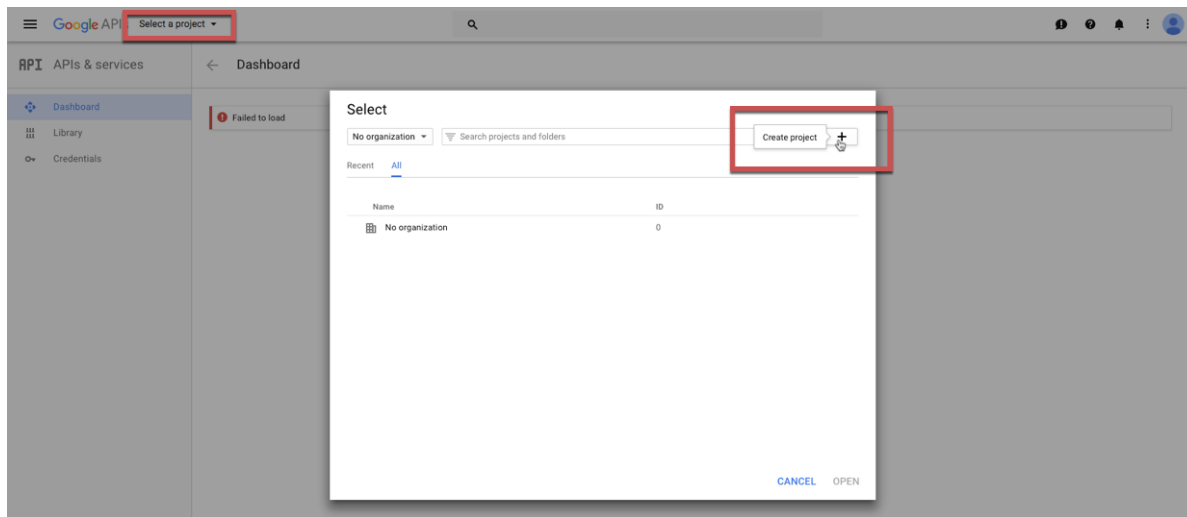
You must also enable the Google Calendar API. For the steps see [Enable Google APIs](#).

Create an Application

If you have not already created a project and application, you need one to authenticate with Google Calendar.

To create a project and application:

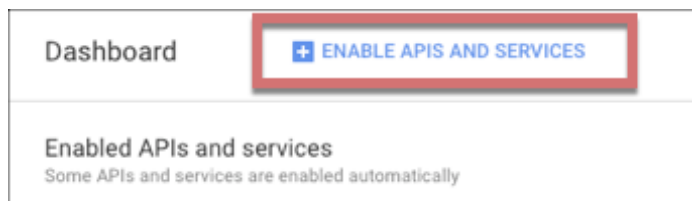
1. Log in to your account at [Google](#).
2. Click **Select a project** and click the **Create Project** button.



3. Accept the terms of service.
4. Complete the required information (Project Name, email updates, and the Terms of Service).

 A screenshot of the 'New Project' form in the Google API console. The form includes a quota warning, a 'Project name' field with the value 'My Project 80329', and an 'Organization' field with the value 'cloud-elements.com'. There are two sections for accepting terms and updates, each with 'Yes' and 'No' radio buttons. The 'Create' button is highlighted with a red box.

5. Click **Create**.
6. Select the project that you just created



7. Click **Enable APIs and Services**.
8. Search for and enable the Google Calendar API.
9. Click **Credentials** on the left menu.

10. Click the **OAuth consent screen** tab.
11. Enter a Product Name and add any optional information, and then click **Save**.

Credentials

Credentials **OAuth consent screen** Domain verification

Email address [?]

Product name shown to users [?]
Awesome Docs App

Homepage URL (Optional)
https://cloud-elements.com

Product logo URL (Optional) [?]
http://www.example.com/logo.png

This is how your logo will look to end users
Max size: 120x120 px

Privacy policy URL
Optional until you deploy your app
https:// or http://

Terms of service URL (Optional)
https:// or http://

Save Cancel

The consent screen will be shown to users whenever you request access to their private data using your client ID. It will be shown for all applications registered in this project.

You must provide an email address and product name for OAuth to work.

12. Click **Create Credentials**, and then select **OAuth Client ID**.

APIs
Credentials

You need credentials to access APIs. [Enable the APIs you plan to use](#) and then create the credentials they require. Depending on the API, you need an API key, a service account, or an OAuth 2.0 client ID. [Refer to the API documentation](#) for details.

Create credentials ▾

- API key
Identifies your project using a simple API key to check quota and access
- OAuth client ID**
Requests user consent so your app can access the user's data
- Service account key
Enables server-to-server, app-level authentication using robot accounts
- Help me choose
Asks a few questions to help you decide which type of credential to use

13. Select **Web application** as the Application type.
14. Enter a **Name** and the **Authorized redirect URI** for your app. Record this as the OAuth Callback URL that you will need to authenticate.

← Create client ID

Application type

- Web application
- Android [Learn more](#)
- Chrome App [Learn more](#)
- iOS [Learn more](#)
- PlayStation 4
- Other

Name

Web client 1

Restrictions
Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins
For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

https://www.example.com

Authorized redirect URIs
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

https://auth.cloud-elements.com/oauth ×

https://www.example.com/oauth2callback

Create Cancel

15. Click **Create**.

16. Record the **Client ID** and **Client secret** to use when you authenticate.

OAuth client

Here is your client ID

1359780821

Here is your client secret

Rc-D5l

OK

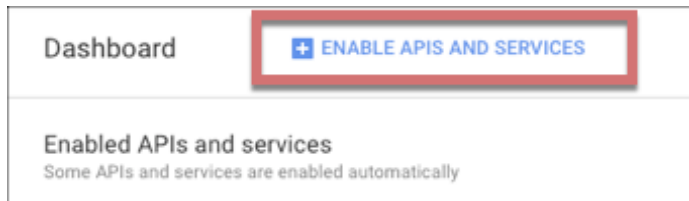
Enable Google APIs

To enable API requests from SAP Cloud Platform Open Connectors or your app to access Google Calendar, you must enable the Google Drive API and the Google Calendar API. If you do not enable the APIs, you might get 403 errors in your response with a `providerMessage` such as:

```
{
  "providerMessage": "error - {code=403, message=Google Calendar API has not
  been used in project .....}"
}
```

To enable the required Google APIs:

1. On the Dashboard page, click **Enable APIs and Services**.



2. Search for and enable the Google Calendar API.
-