

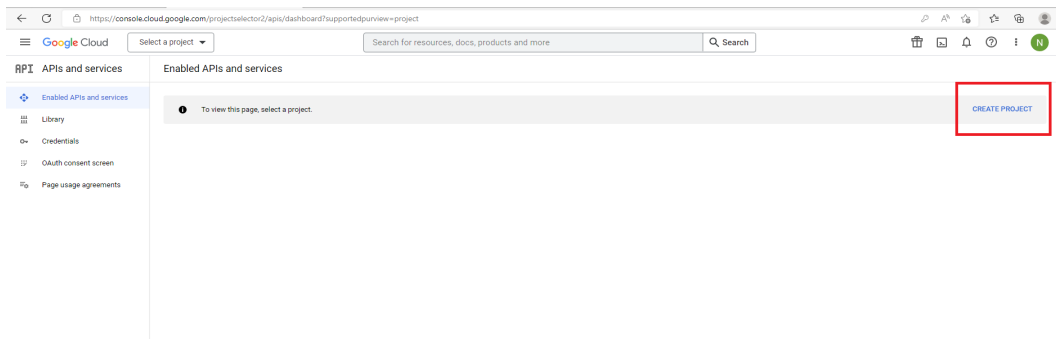
G-Link Google Drive Client Id & Client Secret Setup

In order to upload your documents in Google Drive from your Salesforce via G-Link, you must have Google Client Id, Google Secret Id and Google Refresh Token. For that you need to Sign Up for Google Drive. If you already have a Google Drive account just ignore it. Otherwise follow the **G-Link Google Drive Sign Up** Guide document to create a Google Account.

After that, you will have to generate Google Client Id, Google Secret Id and Google Refresh Token in that Google Account which will be used in your Salesforce to connect your Google account with your Salesforce. Follow below steps to generate Google Client Id, Google Secret Id and Google Refresh Token.

1. Create Google Project

- a. Go to <https://console.developers.google.com> and sign in your Google account
- b. Click on **CREATE PROJECT**



- c. Provide Project name and Location and click **CREAT**

New Project

 You have 12 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)
[MANAGE QUOTAS](#)

Project name *

My Project - 1

?

Project ID: tidy-set-370106. It cannot be changed later. [EDIT](#)

Location *

 No organisation

BROWSE

Parent organisation or folder

CREATE

CANCEL

2. Enable Google Drive API

- a. Open the newly created project
b. Go to the **Library** at the left side section

Google Cloud

My Project - 1

Search for resources, docs, products and more

Q Sea

APIs and services

APIs & Services

+ ENABLE APIS AND SERVICES

Enabled APIs and services

Library

Credentials

OAuth consent screen

Page usage agreements

Traffic

Errors

- c. Search for the **Google Drive API** in the search box and then select **Google Drive API**

API Library

drive

X

API Library > "drive"

Filter Type to filter

Visibility Public (11)

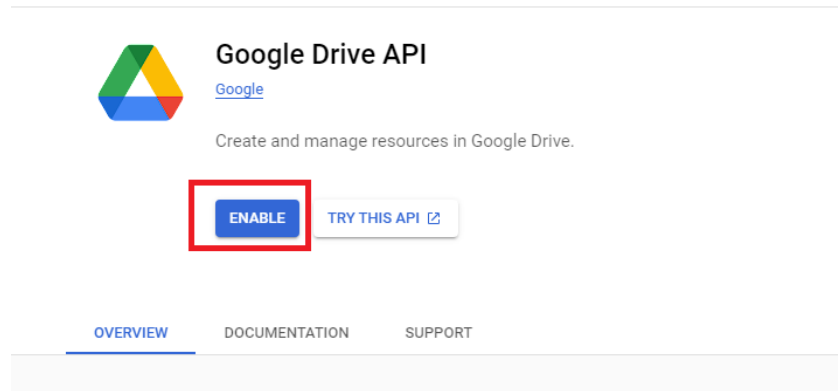
Category Big data (3) Analytics (2) Databases (2)

11 results

 **Google Drive API**
Google
With the Google Drive API, you can access resources from Google Drive to create files, manage file sharing, search for files and folders, and more.

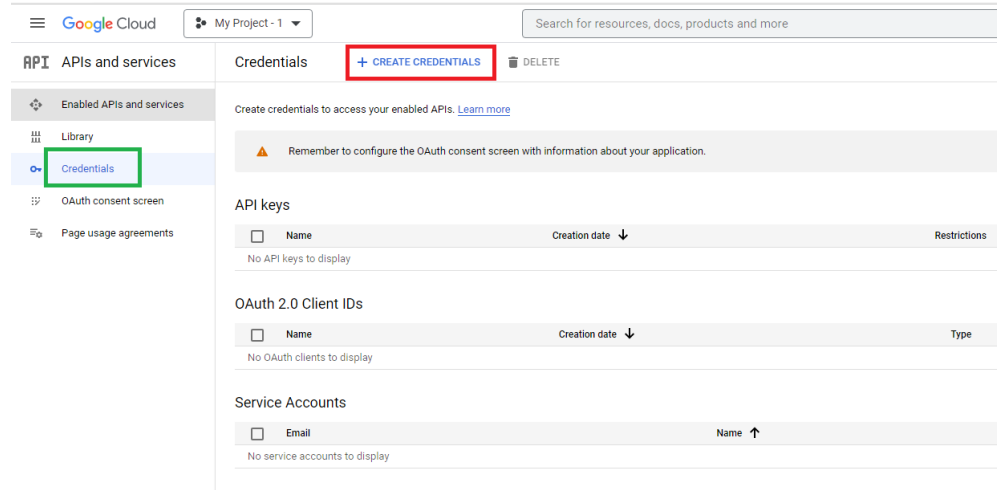
 **Google Drive Activity API**
Google
With the Google Drive Activity API you can retrieve information about a user's Google Drive activity. This provides additional functionality on top the Google Drive API for your app to display activity on a user's files, track changes to specific files or folders, or alert a user to new comments or changes to files.

- d. After selecting it, click **ENABLE**

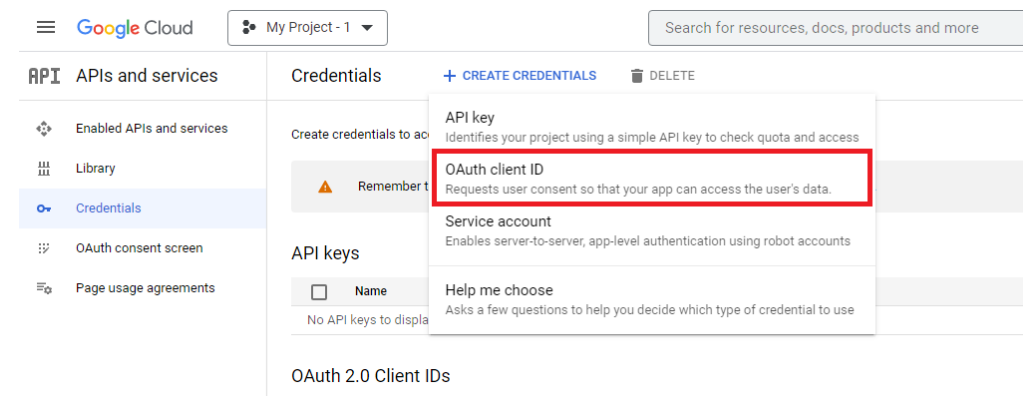


3. Configure Consent Screen

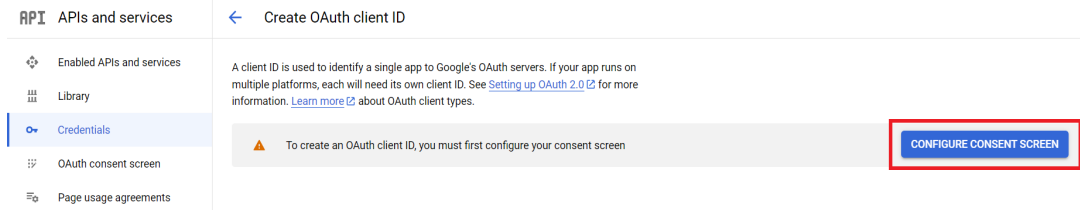
- a. Go to Credentials tab at left side corner and click **+ CREATE CREDENTIAL**



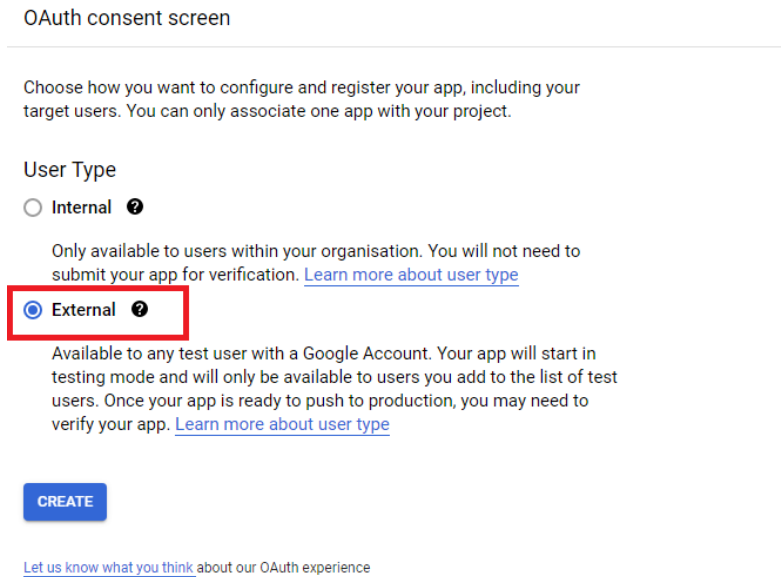
- b. Choose **OAuth Client ID**



- c. Click on the **CONFIGURE CONSENT SCREEN**. This will be displayed only if the consent screen is not configured in the past.



- d. Select **External** as User Type and click on **CREATE**



- e. There will be 4 steps. Provide all the required information in the fields like App name, User support email etc in step 1 of OAuth consent screen.
- f. You can skip the 2nd step of Scope and 3rd step of Test users and click on **SAVE AND CONTINUE**

- g. At the end you can click on **BACK TO DASHBOARD** in step 4 of Summary.

Edit app registration

1 OAuth consent screen — 2 Scopes — 3 Test users — 4 Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *
The name of the app asking for consent

User support email *
For users to contact you with questions about their consent

App logo [BROWSE](#)
Upload an image, not larger than 1 MB on the consent screen that will help users recognise your app. Allowed image formats are JPG, PNG and BMP. Logos should be square and 120px by 120px for the best results.

App domain

To protect you and your users, Google only allows apps using OAuth to use Authorised Domains. The following information will be shown to your users on the consent screen.

Application home page
Provide users a link to your home page

Application privacy policy link
Provide users a link to your public privacy policy

Application Terms of Service link
Provide users a link to your public Terms of Service

4. Create Client Id and Client Secret

After Consent screen configuration, now you can create Client Id and the secret.

- Click again on the Credentials tab at the left side corner and then click on **+CREATE CREDENTIALS** and choose **OAuth Client Id**
- Select **Web Application** as Application Type
- Click on **ADD URI** in Authorised redirect URIs section and add <https://developers.google.com/oauthplayground> in the field and then click on **CREATE**

Application type *
Web application

Name *
Web client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorised domains](#).

Authorised JavaScript origins ⓘ

For use with requests from a browser

+ ADD URI

Authorised redirect URIs ⓘ

For use with requests from a web server

URIs 1 *
<https://developers.google.com/oauthplayground>

+ ADD URI

Note: It may take five minutes to a few hours for settings to take effect

CREATE CANCEL

- d. Now, Client Id and Client Secret will be shown immediately in a popup.

OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

OAuth access is restricted to the [test users](#) listed on your [OAuth consent screen](#)

Your Client ID

Your Client Secret

DOWNLOAD JSON

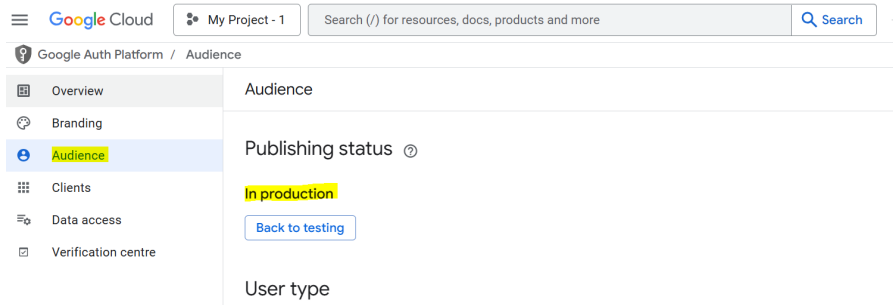
OK

5. Publish Application

You have to publish your app and set your app Publishing status as **In production** so that there will be no verification required.

- Go to <https://console.developers.google.com>
- Select your app

- c. Click **OAuth Consent Screen** at the left side panel and select **Audience**
- d. Click **PUBLISH APP**



6. Create Refresh Token

There are several ways to get Refresh Token in Google. But we will be using the OAuth 2.0 Playground app to get the Refresh Token. OAuth 2.0 Playground is a public tool provided by Google to facilitate the development process. If you do not want to use OAuth 2.0 Playground app, you can also use other Google options to get the Refresh Token using Client Id and Client Secret.

Follow below steps to get the Refresh Token using OAuth 2.0 Playground app.

- a. Open <https://developers.google.com/oauthplayground/> in incognito mode of your browser
- b. Click on gear icon on the right top corner and check the **Use your own OAuth credentials** checkbox and then provide the Client Id and Client Secret provided in step 4.d

OAuth 2.0 configuration

OAuth flow: Server-side

OAuth endpoints: Google

Authorization endpoint: <https://accounts.google.com/o/oauth2/v2/auth>

Token endpoint: <https://oauth2.googleapis.com/token>

Access token location: Authorization header w/ Bearer prefix

Access type: Offline

Force prompt: Consent Screen

☒ Use your own OAuth credentials

You will need to list the URL <https://developers.google.com/oauthplayground> as a valid redirect URI in your [Google APIs Console](#)'s project. Then enter the client ID and secret assigned to a web application on your project below:

OAuth Client ID: [\[REDACTED\]](#)

OAuth Client secret: [\[REDACTED\]](#)

Note: Your credentials will be sent to our server as we need to proxy the request. Your credentials will not be logged.

[Close](#)

- c. Click on **Drive API v3** from the left side panel and check all the items under Drive API v3



OAuth 2.0 Playground ×

Step 1 Select & authorize APIs

Select the **scope** for the APIs you would like to access or input your own OAuth scopes below. Then click the "Authorize APIs" button.

Cloud Vision API v1

Compute Engine API v1

Contacts v3

Content API for Shopping v2.1

Dataflow API v1b3

Drive API v3

✓ https://www.googleapis.com/auth/drive

✓ https://www.googleapis.com/auth/drive.appdata

✓ https://www.googleapis.com/auth/drive.file

✓ https://www.googleapis.com/auth/drive.metadata

✓ https://www.googleapis.com/auth/drive.metadata.readonly

✓ https://www.googleapis.com/auth/drive.photos.readonly

✓ https://www.googleapis.com/auth/drive.readonly

✓ https://www.googleapis.com/auth/drive.scripts

Drive Activity API v2

Enterprise License Manager API v1

Error Reporting API v1beta1

Fact Check Tools API v1alpha1

Firebase Cloud Messaging API v1

Input your own scopes

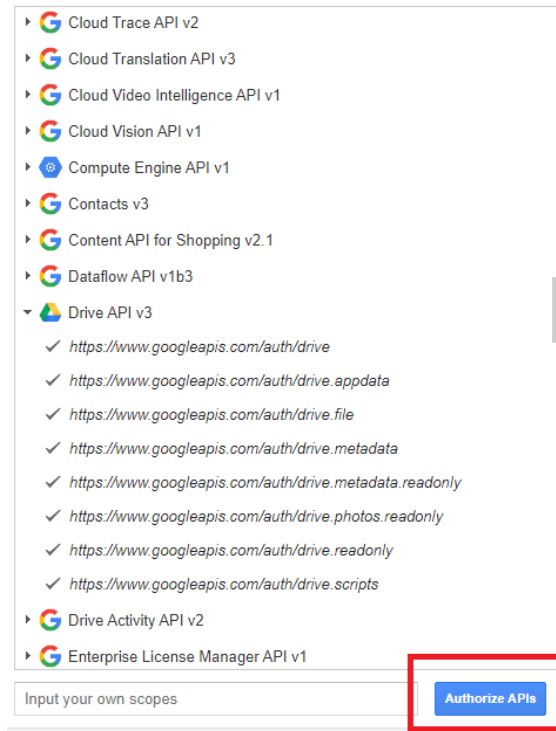
Authorize APIs

Request / Response

No request.

d. Click **Authorize APIs**

© Neilon Technologies LLP. All rights reserved.



- e. Provide your Google account credentials and login
- f. It will display **Google hasn't verified this app** screen, click **Advance**



Google hasn't verified this app

The app is requesting access to sensitive info in your Google Account. Until the developer (My First App) verifies this app with Google, you shouldn't use it.

If you're the developer, submit a verification request to remove this screen. [Learn more](#)

[Hide Advanced](#)

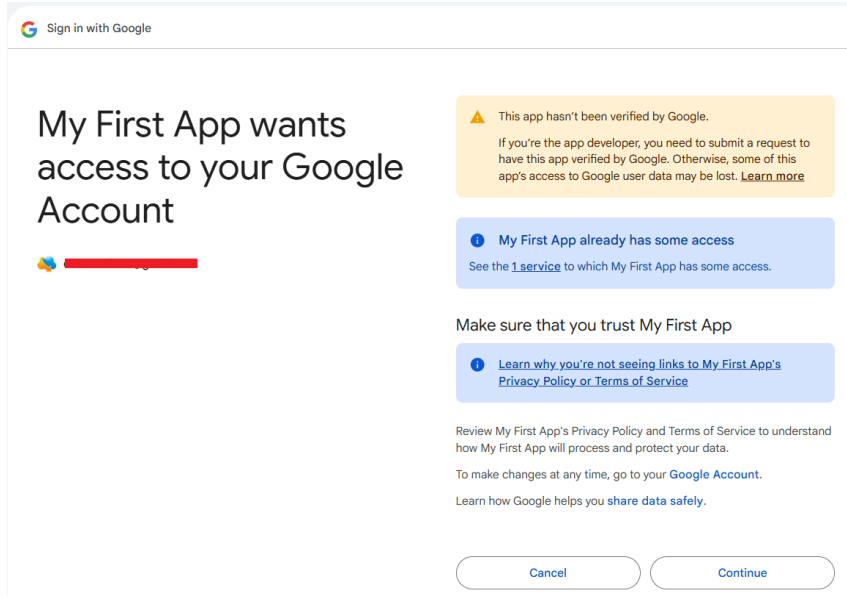
[BACK TO SAFETY](#)

Continue only if you understand the risks and trust the developer

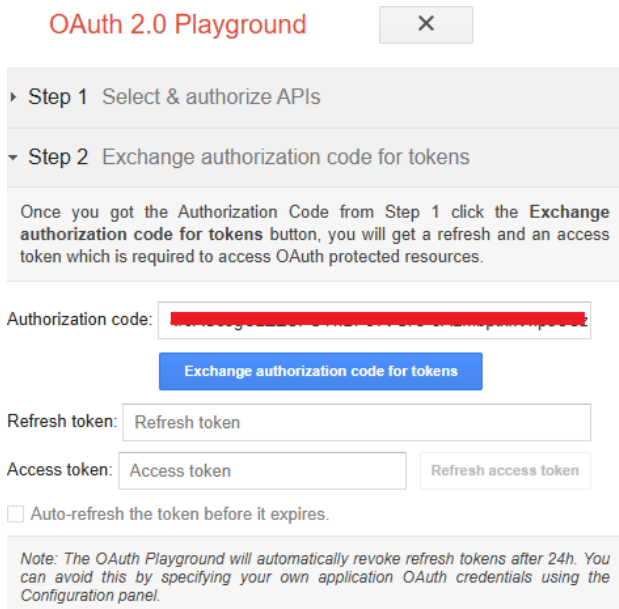
(My First App).

[Go to My First App \(unsafe\)](#)

- g. Click **Go to My First App**. Here My First App is a test OAuth 2 Playground app auto created in your Google account.
- h. It will ask you to give access to your test OAuth 2.0 playground app
- i. Click **Continue**



j. You will see the screen below



- k. Click **Exchange authorization code for tokens** button
- l. You will see Refresh token which you can use in your Salesforce along with Client Id and Client Secret



▶ Step 1 Select & authorize APIs

- ▼ Step 2 Exchange authorization code for tokens

Once you got the Authorization Code from Step 1 click the **Exchange authorization code for tokens** button, you will get a refresh and an access token which is required to access OAuth protected resources.

Exchange authorization code for tokens

Refresh access token

☐ Auto-refresh the token before it expires.

The access token will expire in 3574 seconds.

Note: The OAuth Playground will automatically revoke refresh tokens after 24h. You can avoid this by specifying your own application OAuth credentials using the Configuration panel.