

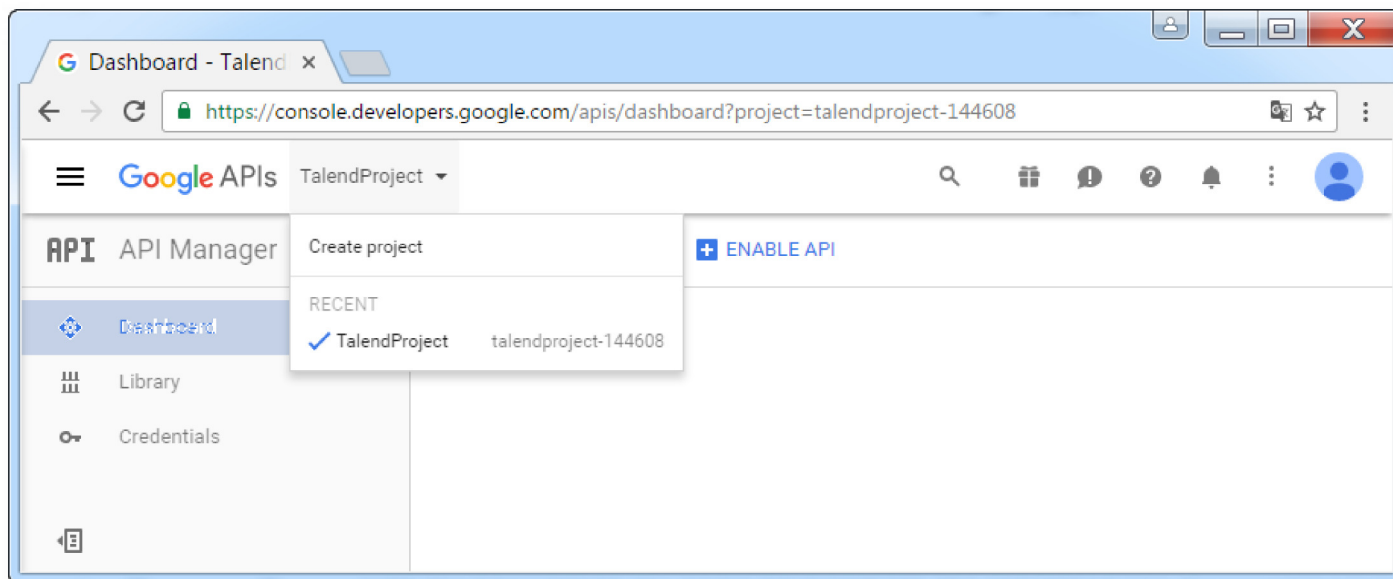
Google Drive related topic

Obtaining an access token for Google Drive

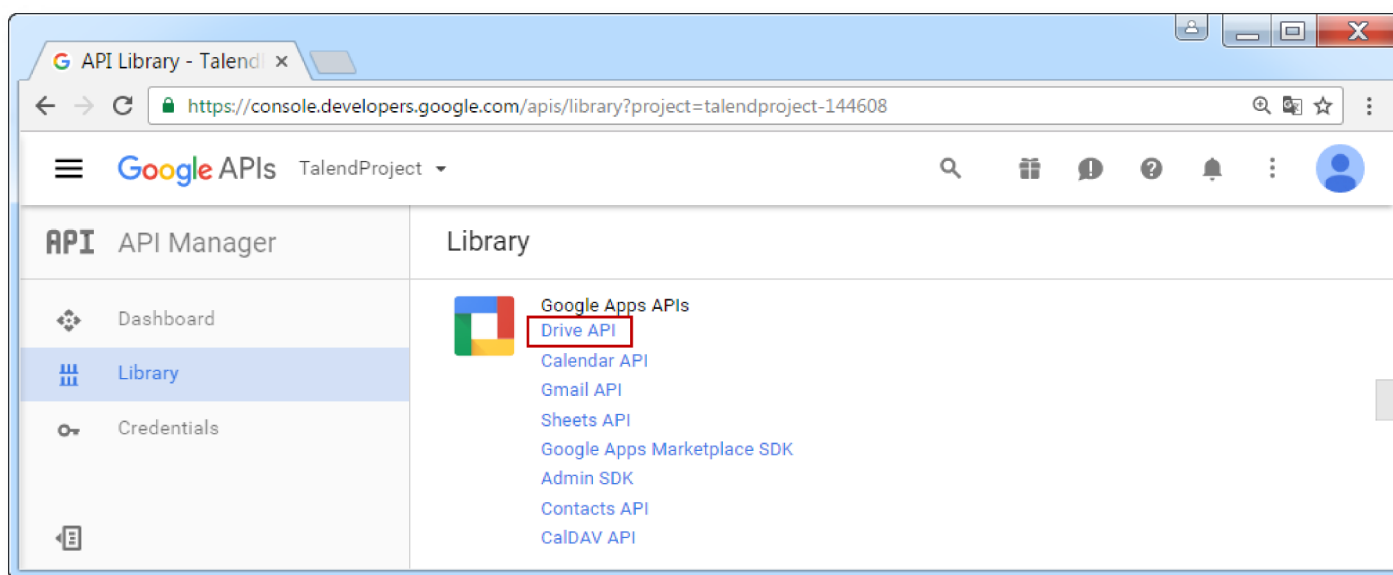
To use Google Drive API, you need first to obtain the access token by completing the following steps using Google Chrome:

For more technologies supported by Talend, see [Talend components](#).

1. Go to the [Google API Console](#) and select an existing project or create a new one. In this example, we create a new project *TalendProject*.



2. Go to the **Library** page and in the right panel, click **Drive API** and then click **ENABLE** to enable the Google Drive API that allows clients to access resources from Google Drive.



3. Go to the **Credentials** page, click **OAuth consent screen** in the right panel and set a product name in the **Product name shown to users** field. In this example, it is *TalendProduct*. When done, click **Save** to save the change.

Consent screen - TalendProject

Google APIs TalendProject

API Manager

Credentials

Credentials OAuth consent screen Domain verification

Email address

lena567@gmail.com

Product name shown to users

TalendProduct

Homepage URL (Optional)

https:// or http://

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.

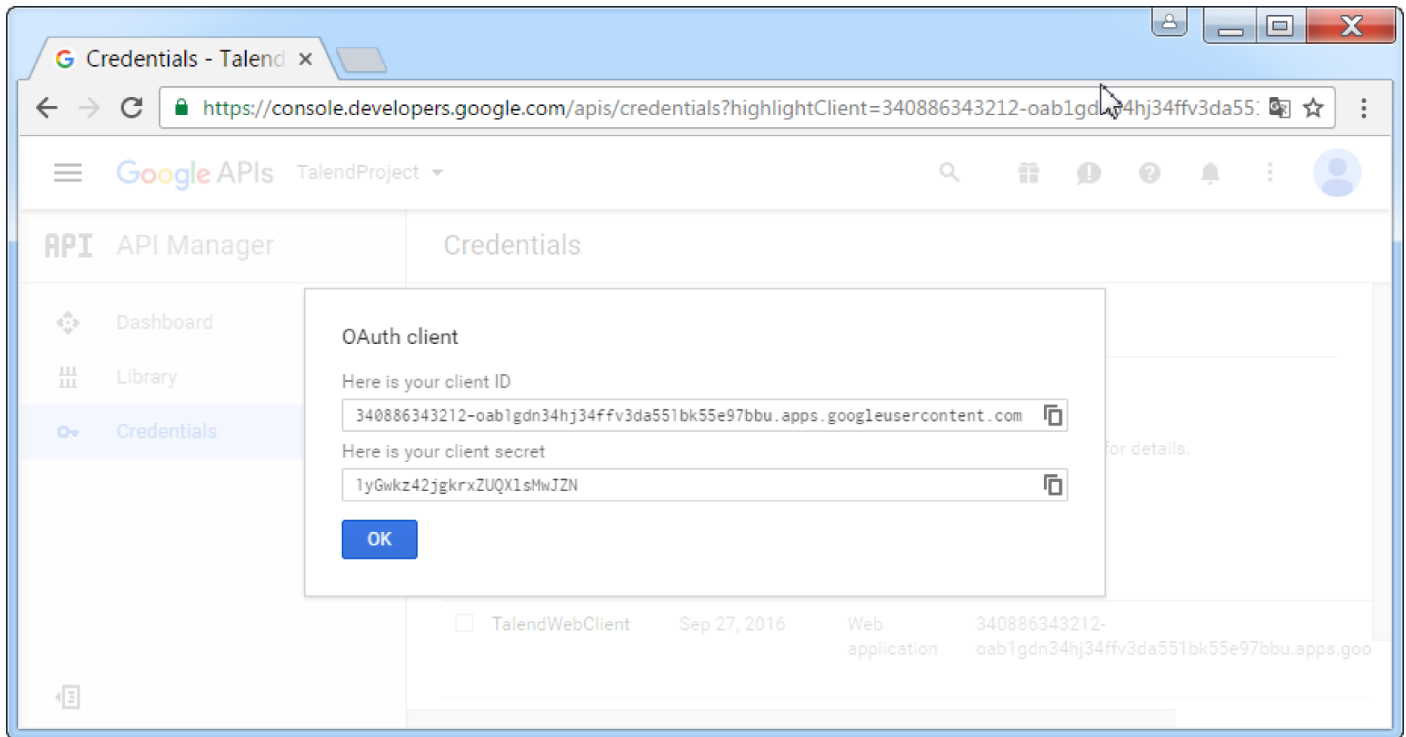
You must provide an email address and product name for OAuth to work.

- Click **Create credentials > OAuth client ID**, and in the **Create client ID** page, create a new client ID *TalendWebClient* with **Application type** set to **Web application** and **Authorized redirect URIs** set to <https://developers.google.com/oauthplayground>.

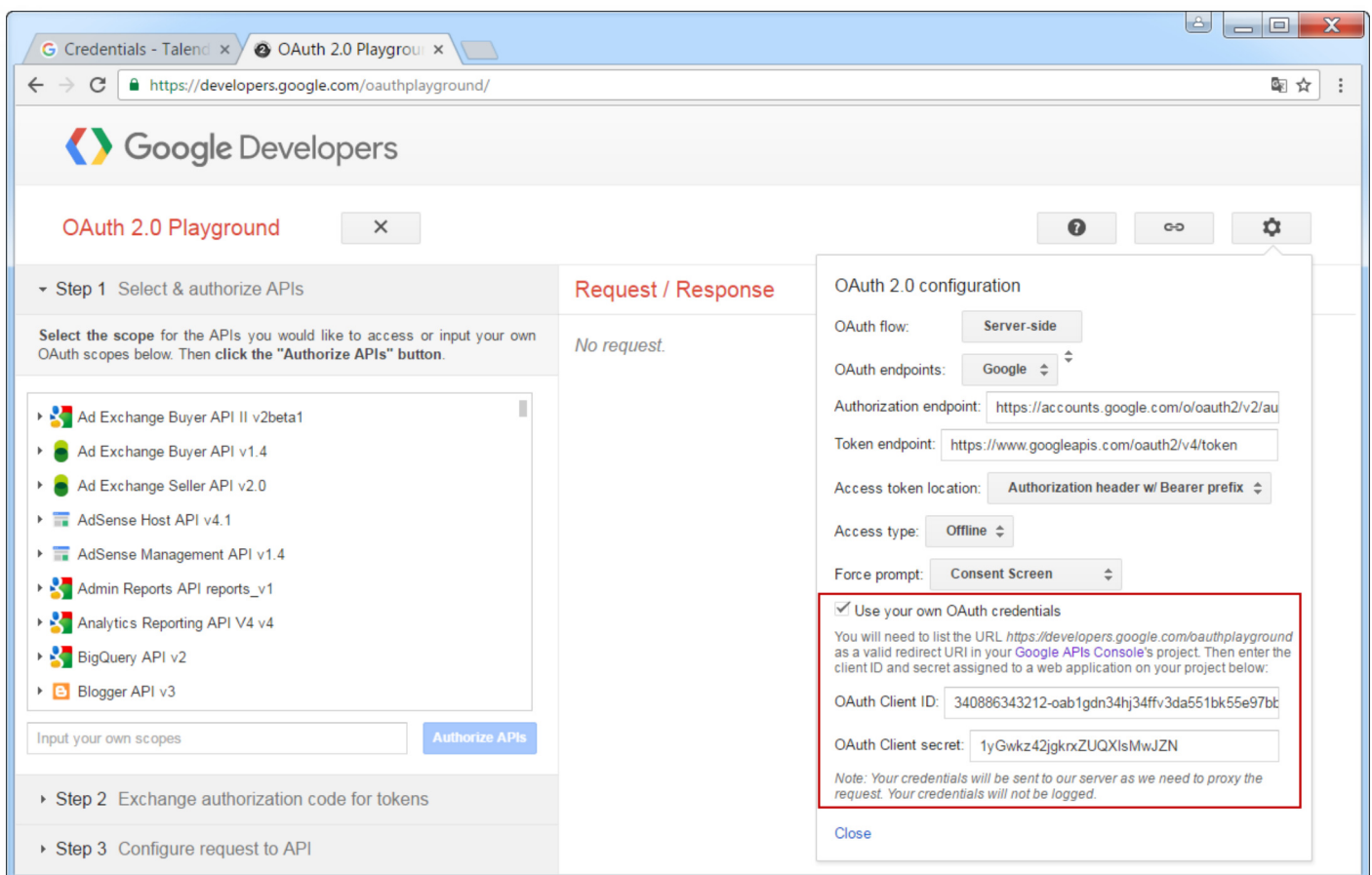
The screenshot shows the 'Create client ID' page in the Google Developers console. The left sidebar has a menu with 'Dashboard', 'Library', and 'Credentials'. The 'Credentials' section is active. The main content area is titled 'Create client ID' and includes the following fields:

- Application type:** Radio buttons for 'Web application' (selected), 'Android', 'Chrome App', 'iOS', 'PlayStation 4', and 'Other'. Each has a 'Learn more' link.
- Name:** A text input field containing 'TalendWebClient'.
- Restrictions:** A section with the instruction 'Enter JavaScript origins, redirect URIs, or both'. It contains two sub-sections:
 - Authorized JavaScript origins:** A text input field containing 'http://www.example.com'.
 - Authorized redirect URIs:** A text input field containing 'https://developers.google.com/oauthplayground'.
- Buttons:** 'Create' and 'Cancel' buttons at the bottom.

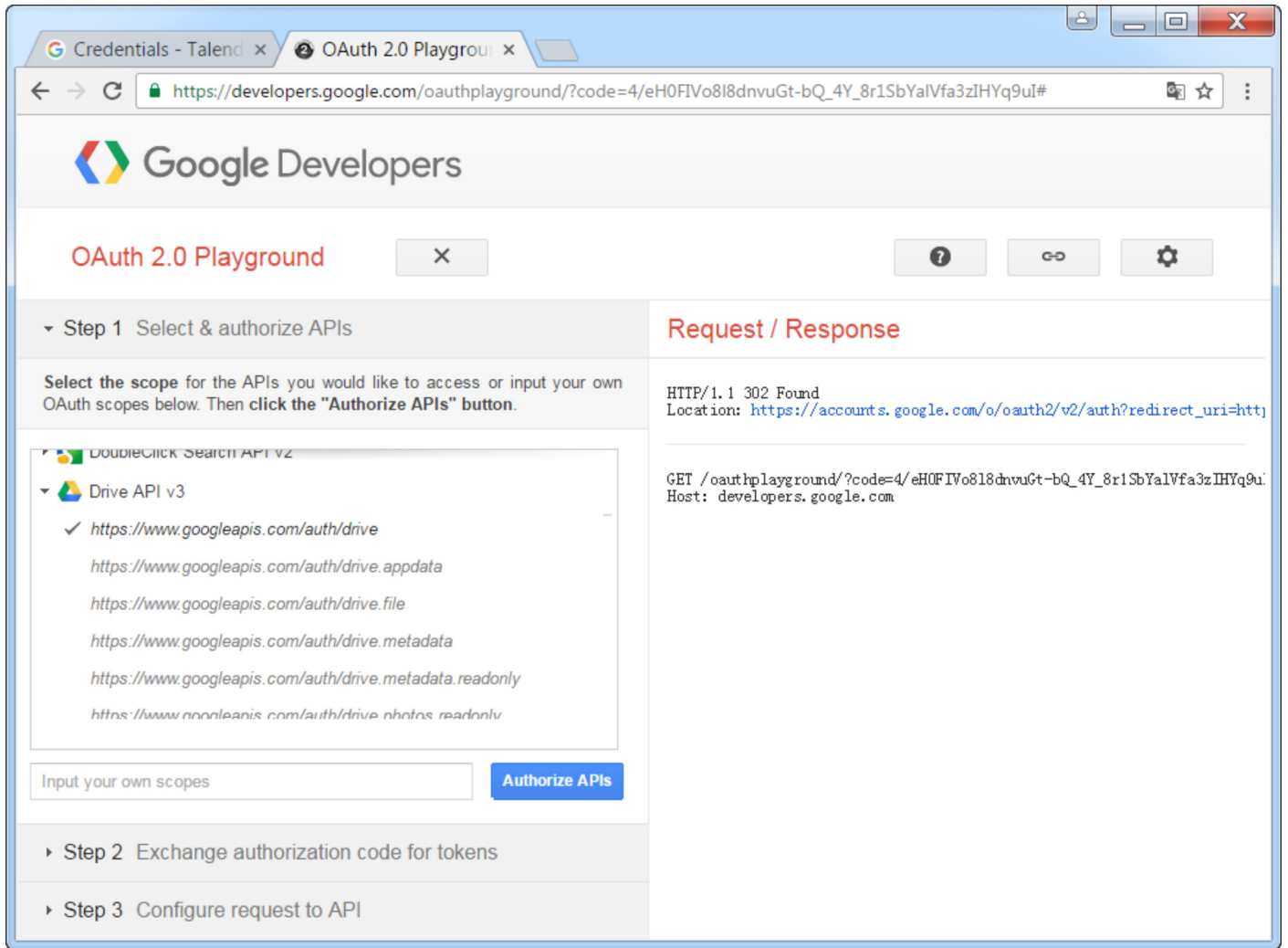
The created OAuth client ID and client secret will be used for obtaining the access token.



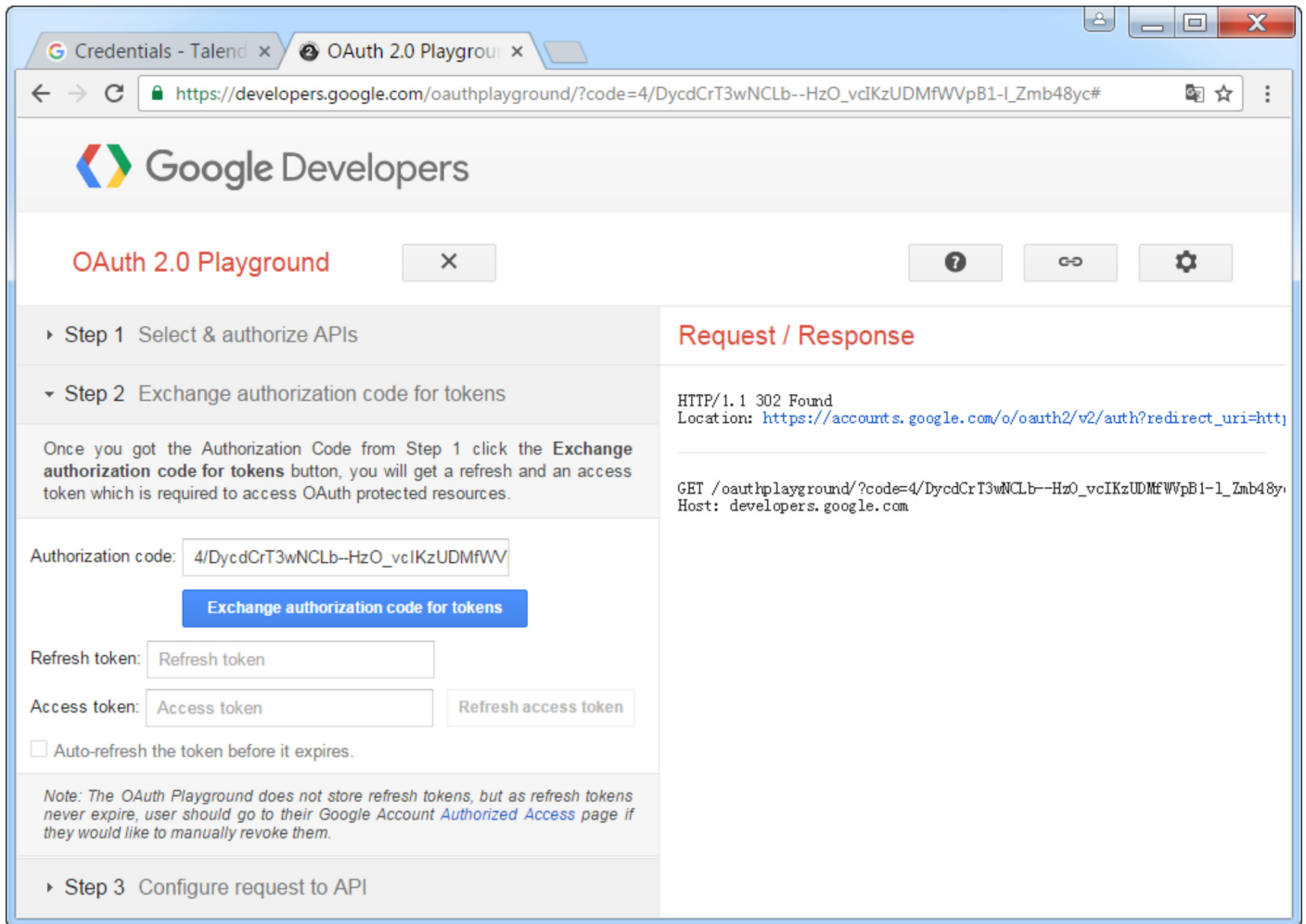
5. Go to the [Google Developers OAuth 2.0 Playground](https://developers.google.com/oauthplayground/), click **OAuth 2.0 Configuration** and select **Use your own OAuth credentials** check box, enter the OAuth client ID and client secret you have already created in the **OAuth Client ID** and **OAuth Client secret** fields respectively.



6. In **OAuth 2.0 Playground Step 1**, select the scope `https://www.googleapis.com/auth/drive` for the Google Drive API and click **Authorize APIs**, then click **Allow** to generate the authorization code.



7. In **OAuth 2.0 Playground Step 2**, click **Exchange authorization code for tokens** to generate the OAuth access token.



The OAuth access token is displayed on the right panel as shown in below figure. Note that the access token expires in every 3600 seconds. You can click **Refresh access token** in **OAuth 2.0 Playground Step 2** to refresh it.

Google Developers

OAuth 2.0 Playground

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.

Authorization code: 4/DycdCrT3wNCLb-HzO_vdIKzUDMFVV

Exchange authorization code for tokens

Refresh token: 1/q6EThURBzXZX0HirpeQ-4aXOil

Access token: ya29.Ci9rA8AQXHROgSOpofSR0qfeVmq6GBXMH1rwFME1LVJZid6090GwNq8QLx03gz8tSg

Refresh access token

☐ Auto-refresh the token before it expires.

The access token will expire in **3475** seconds.

Note: The OAuth Playground does not store refresh tokens, but as refresh tokens never expire, user should go to their Google Account [Authorized Access](#) page if they would like to manually revoke them.

Step 3 Configure request to API

Request / Response

POST /oauth2/v4/token HTTP/1.1
 Host: www.googleapis.com
 Content-length: 208
 content-type: application/x-www-form-urlencoded
 user-agent: google-oauth-playground

client_secret=lyGwkz42jgkxZUQX1sMwJZN&grant_type=refresh_token&refresh_token=1%2Fq6EThURBzXZX

HTTP/1.1 200 OK
 Content-length: 141
 X-xss-protection: 1; mode=block
 X-content-type-options: nosniff
 Expires: Mon, 01 Jan 1990 00:00:00 GMT
 Vary: Origin, X-Origin
 Server: GSE
 Pragma: no-cache
 Cache-control: no-cache, no-store, max-age=0, must-revalidate
 Date: Tue, 27 Sep 2016 03:54:04 GMT
 X-frame-options: SAMEORIGIN
 Content-type: application/json; charset=UTF-8

```
{
  "access_token": "ya29.Ci9rA8AQXHROgSOpofSR0qfeVmq6GBXMH1rwFME1LVJZid6090GwNq8QLx03gz8tSg",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

END OF CONTENT