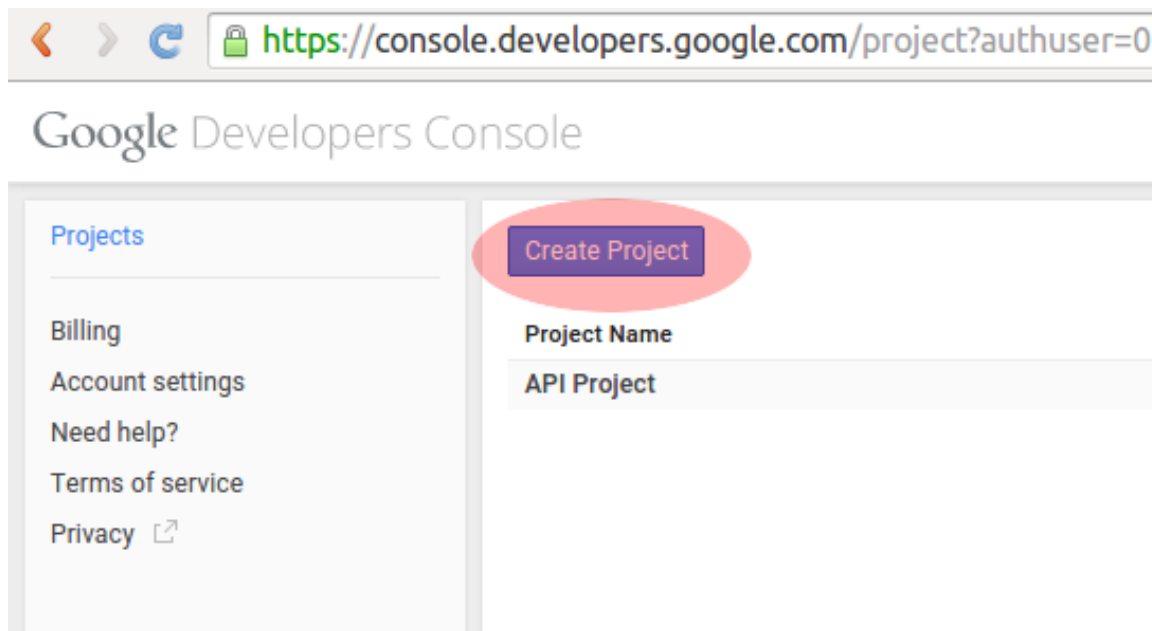


# 1. Register client application with Google

Every application has to be registered with the Google API so that we can use the OAuth 2.0 token during the authentication and authorisation process. To register an application the user has to login to the Google account and go to Google API console.

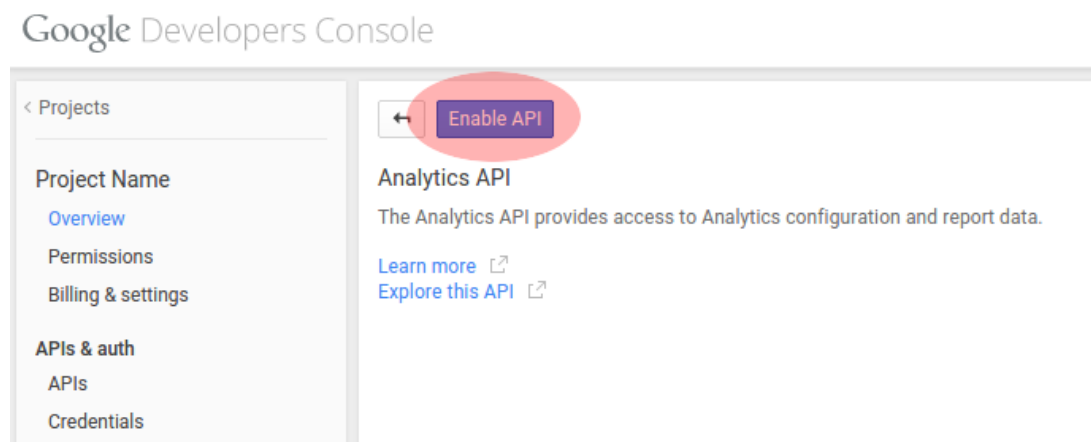
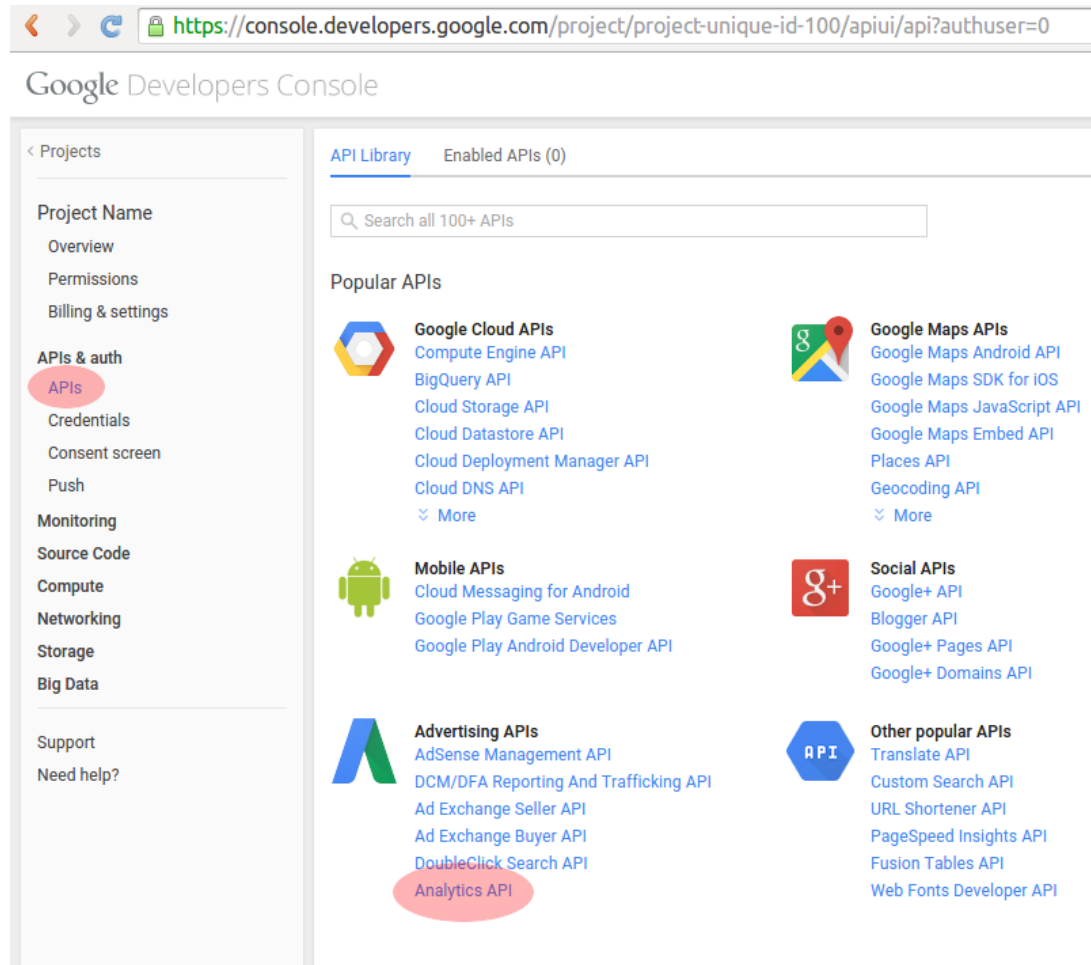
From Google API console Create a New Project using the Create Project button.



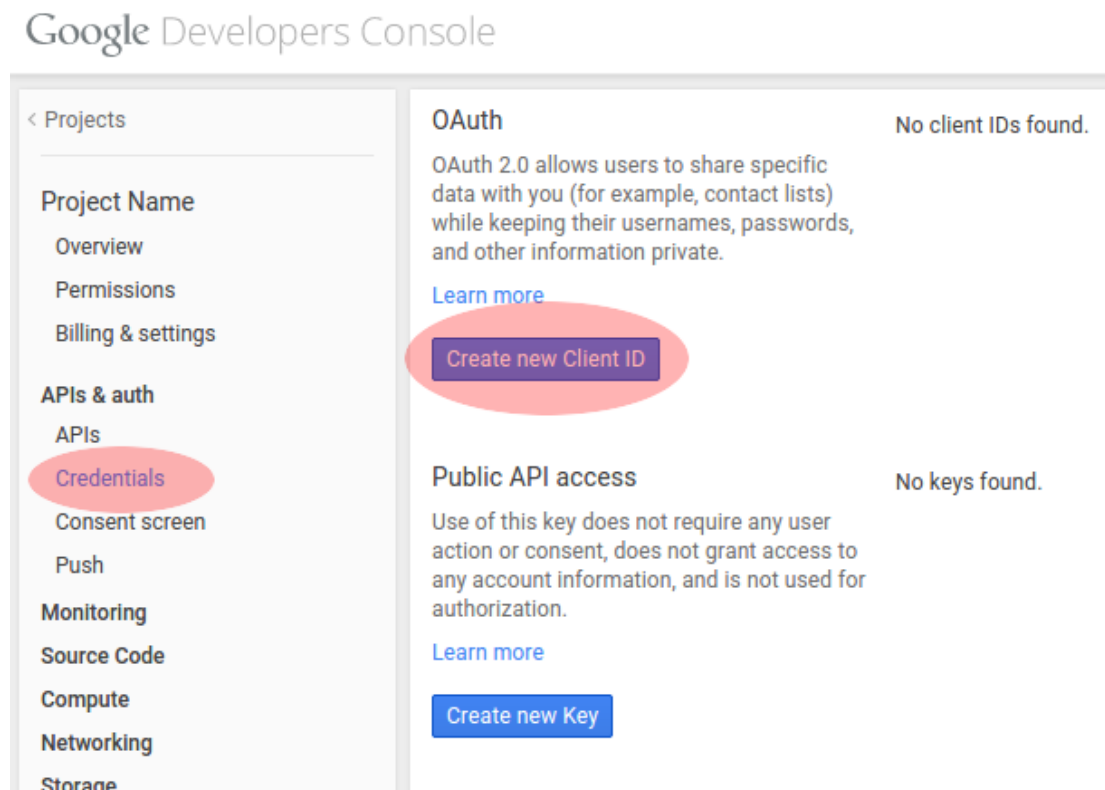
To set up properly the client application, select a unique project name and id.

A screenshot of the 'New Project' form in the Google Developers Console. The form has a title 'New Project'. It contains two input fields: 'Project name' with a placeholder 'Project Name' and a help icon, and 'Project ID' with a placeholder 'project-unique-id-100|' and a refresh icon. Below these fields is a link 'Show advanced options...'. At the bottom of the form are two buttons: 'Create' and 'Cancel'.

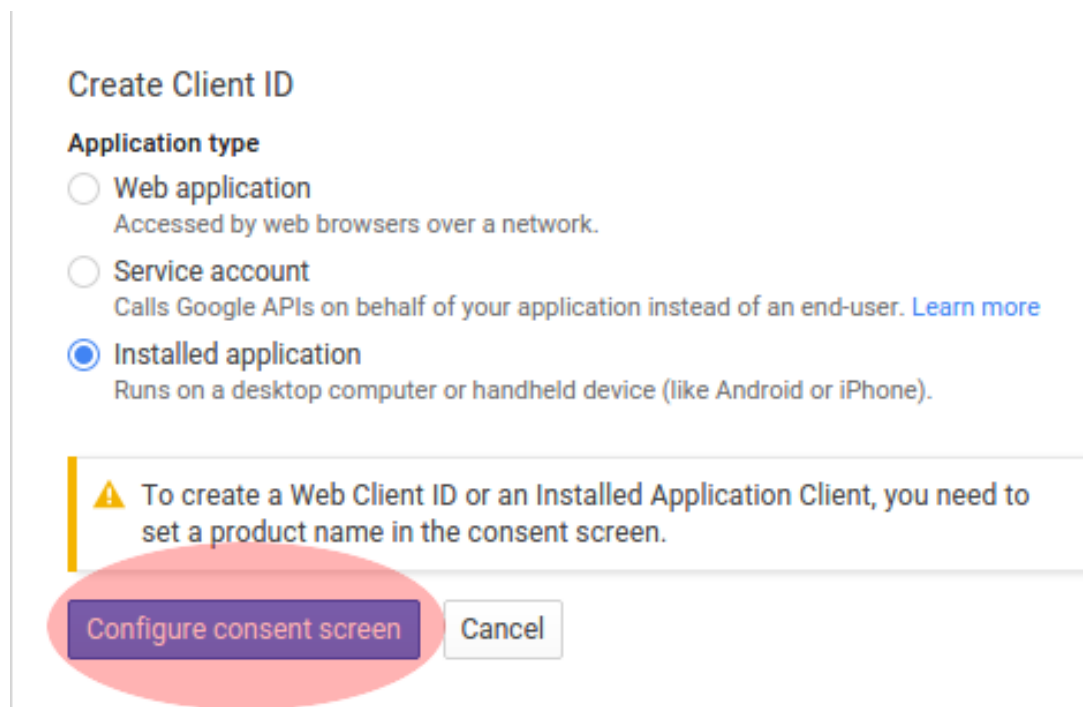
Enable the Analytics API in order to be accessed.



From the APIs → Credentials tab create an OAuth 2.0 Client ID.



Select the application type. In our example we select “Installed application” usually running on the local system.



Create Branding information for Client ID.

## Google Developers Console

< Projects

Project Name

Overview

Permissions

Billing & settings

APIs & auth

APIs

Credentials

Consent screen

Push

Monitoring

Source Code

Compute

Networking

Storage

Big Data

Support

Need help?

Consent screen

The consent screen will be shown to users whenever you request access to their private data using your client ID.

Note: This screen will be shown for all of your applications registered in this project

Email address


Choose your email

Product name

Toplytics GAPI

Homepage URL (Optional)

Product Logo (Optional) ?

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

Privacy policy URL (Optional)

Terms of service URL (Optional)

Google+ Page (Optional) ?

plus.google.com/ Page ID

Save Cancel

By pressing the Create Client ID button, the client application will eventually be created, having the following details: Client ID and Client secret.

**Create Client ID**

**Application type**

- ☐ Web application  
Accessed by web browsers over a network.
- ☐ Service account  
Calls Google APIs on behalf of your application instead of an end-user. [Learn more](#)
- ☒ **Installed application**  
Runs on a desktop computer or handheld device (like Android or iPhone).

**Installed application type**

- ☐ Android [Learn more](#)
- ☐ Chrome Application [Learn more](#)
- ☐ iOS [Learn more](#)
- ☐ PlayStation 4
- ☒ **Other**

**Create Client ID** **Cancel**

The final step is to download the JSON file containing the API credentials. This file must be used in order to set the credentials from WordPress Menu → Settings → Topolytics page

**OAuth**

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

[Learn more](#)

**Create new Client ID**

**Client ID for native application**

<b>Client ID</b>	370128449605-dtkiatldvmhmhac0q21fe0120rhquq0.apps.googleusercontent.com
<b>Client secret</b>	q4awgYH3KoVcOuAI-G8KfAf-
<b>Redirect URIs</b>	urn:ietf:wg:oauth:2.0:oob http://localhost

**Reset secret** **Download JSON** **Delete**

**Public API access**

No keys found.

Use of this key does not require any user action or consent, does not grant access to any account information, and is not used for authorization.

[Learn more](#)

**Create new Key**

## **2. Authorising Requests**

The registration gives the `clientId` and `clientSecret` value for your application so that it can work with the Google API. These keys will avoid the client application to share the username or password with any other Topytics users. When the client application is executed, it prompts the users to allow access and redirects the users to the Google URL which in turn provides an authorisation code. The authorisation code is fed back to the client application which then uses the code to get the access token.