ΑΔΟΠΣΕ

Κωνσταντίνα Καραμπίδου

APIs Database Profiles: Google accounts or anything else

This document explains how applications installed on devices like phones, tablets, and computers use Google's OAuth 2.0 endpoints to authorize access to Google APIs.

OAuth 2.0 allows users to share specific data with an application while keeping their usernames, passwords, and other information private. For example, an application can use OAuth 2.0 to obtain permission from users to store files in their Google Drives.

Prerequisites(Προϋποθέσεις)

- Enable APIs for your project
  - Any application that calls Google APIs needs to enable those APIs in the API Console.

- Create authorization credentials
  - Any application that uses OAuth 2.0 to access Google APIs must have authorization credentials that identify the application to Google's OAuth 2.0 server.

- Identify access scopes

- Obtaining OAuth 2.0 access tokens

- Calling Google APIs

  - After your application obtains an access token, you can use the token to make calls to a Google API on behalf of a given user account or service account.

- Refreshing an access token

  - Access tokens periodically expire. You can refresh an access token without prompting the user for permission (including when the user is not present) if you requested offline access to the scopes associated with the token.

- Revoking a token

  - In some cases a user may wish to revoke access given to an application. A user can revoke access by visiting Account Settings. It is also possible for an application to programmatically revoke the access given to it. Programmatic revocation is important in instances where a user unsubscribes or removes an application. In other words, part of the removal process can include an API request to ensure the permissions granted to the application are removed.

Step-by-step implementation of Google OAuth 2.0

Roles

There are usually three roles involved in OAuth implementations:

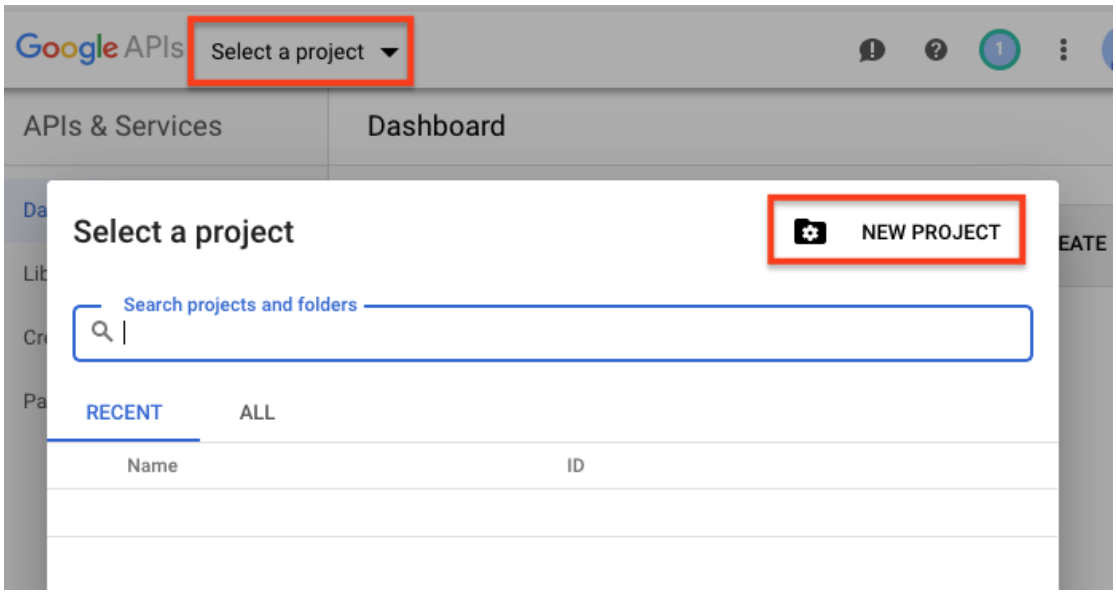
- Resource owner: The user that grants access to their data (or some of it).
- Third-party application: The website or application that wants to access the resource owner's data.
- Resource server: The server that holds the resource owner's data, which is exposed through APIs.

Obtain the OAuth client ID and client secret

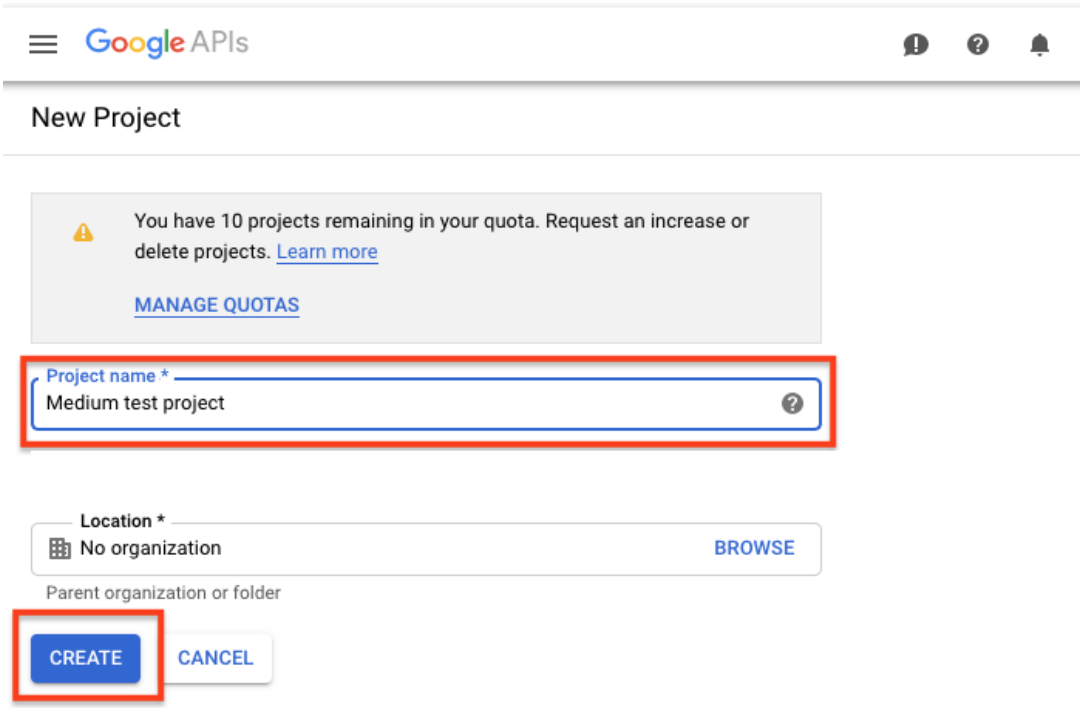
The first things we need are a client ID and client secret.

You can get them by going to the Google API console and then following the steps below.

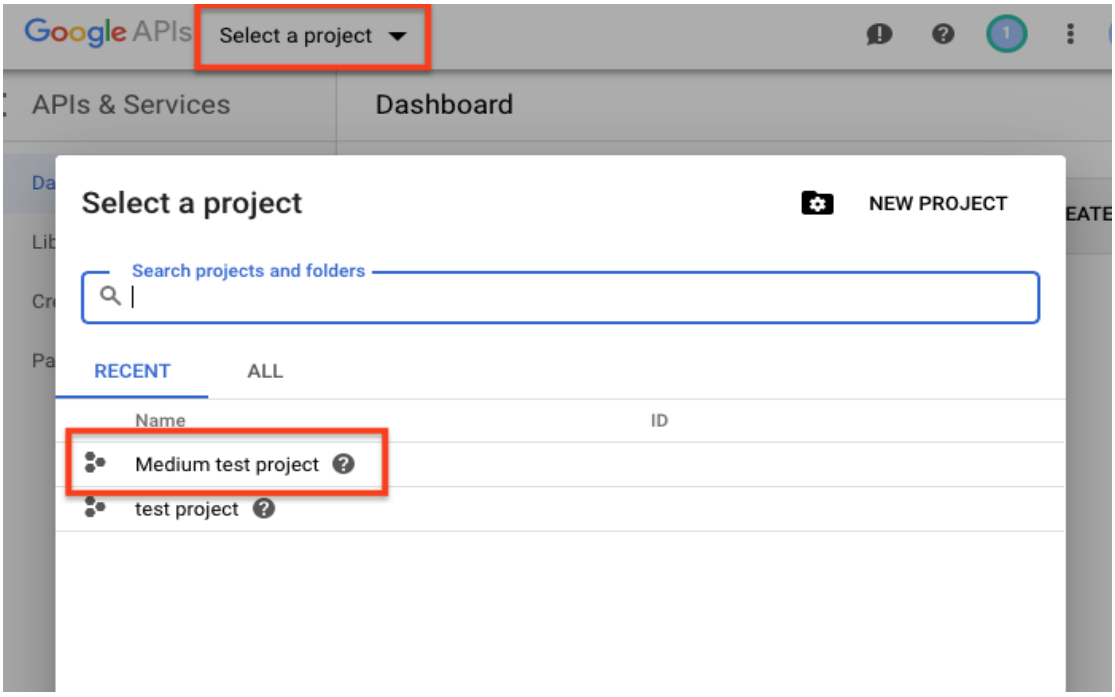
1. Create a new project.



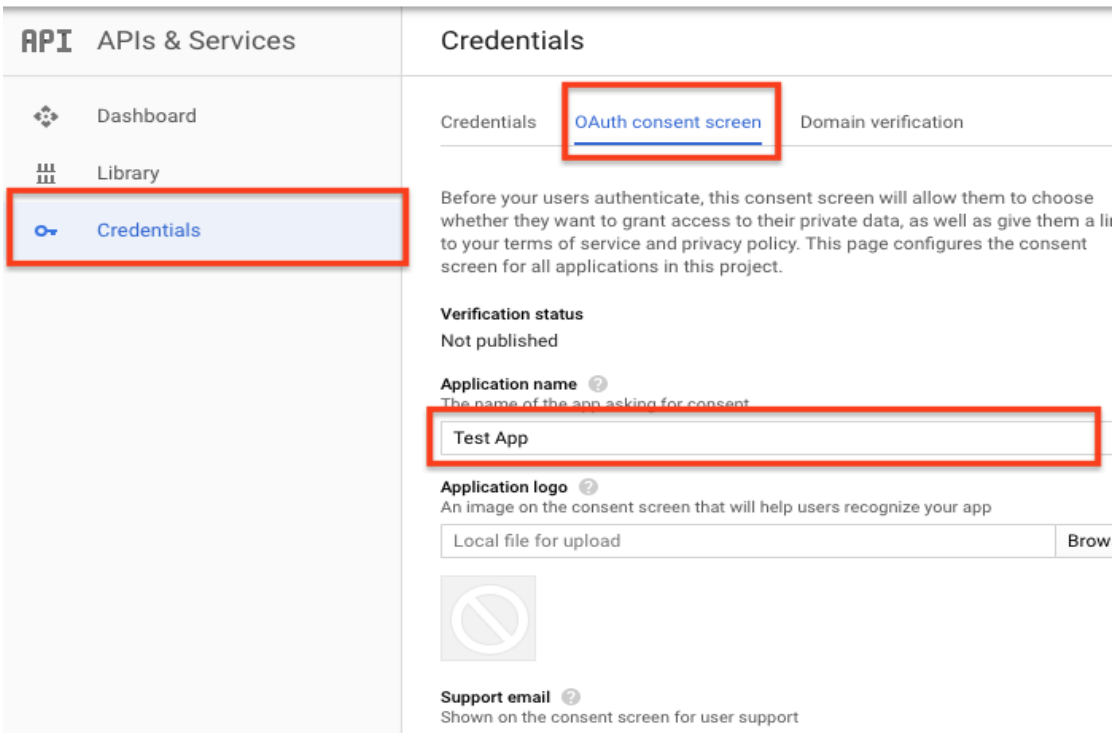
2. Choose a name for the project.



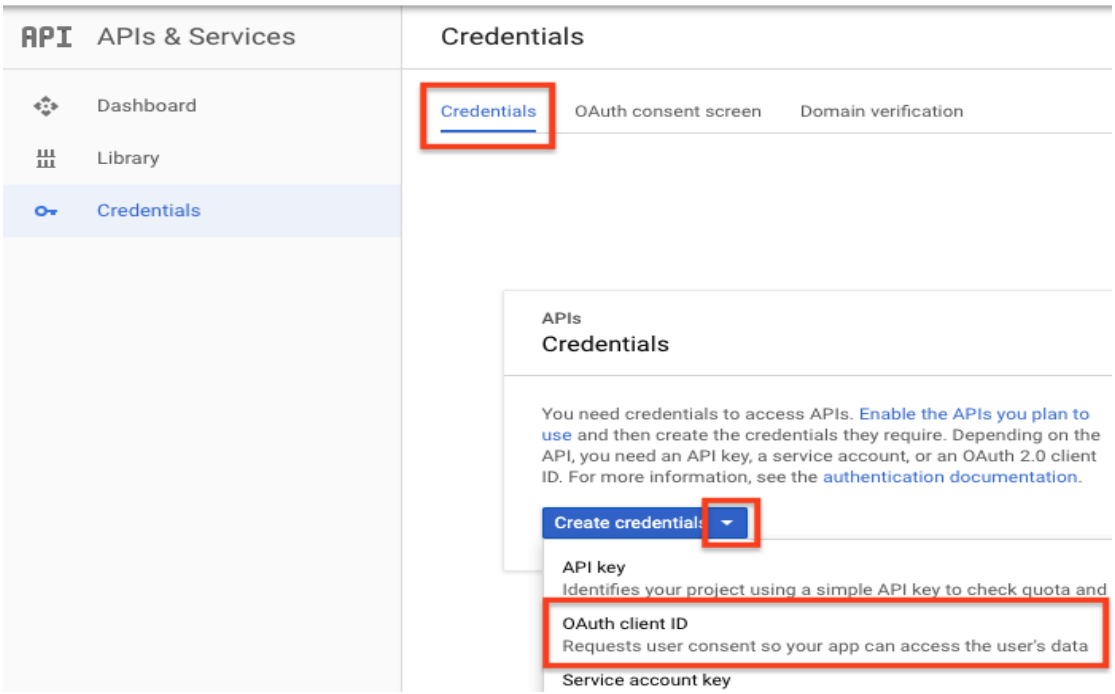
3. Select the project you just created.



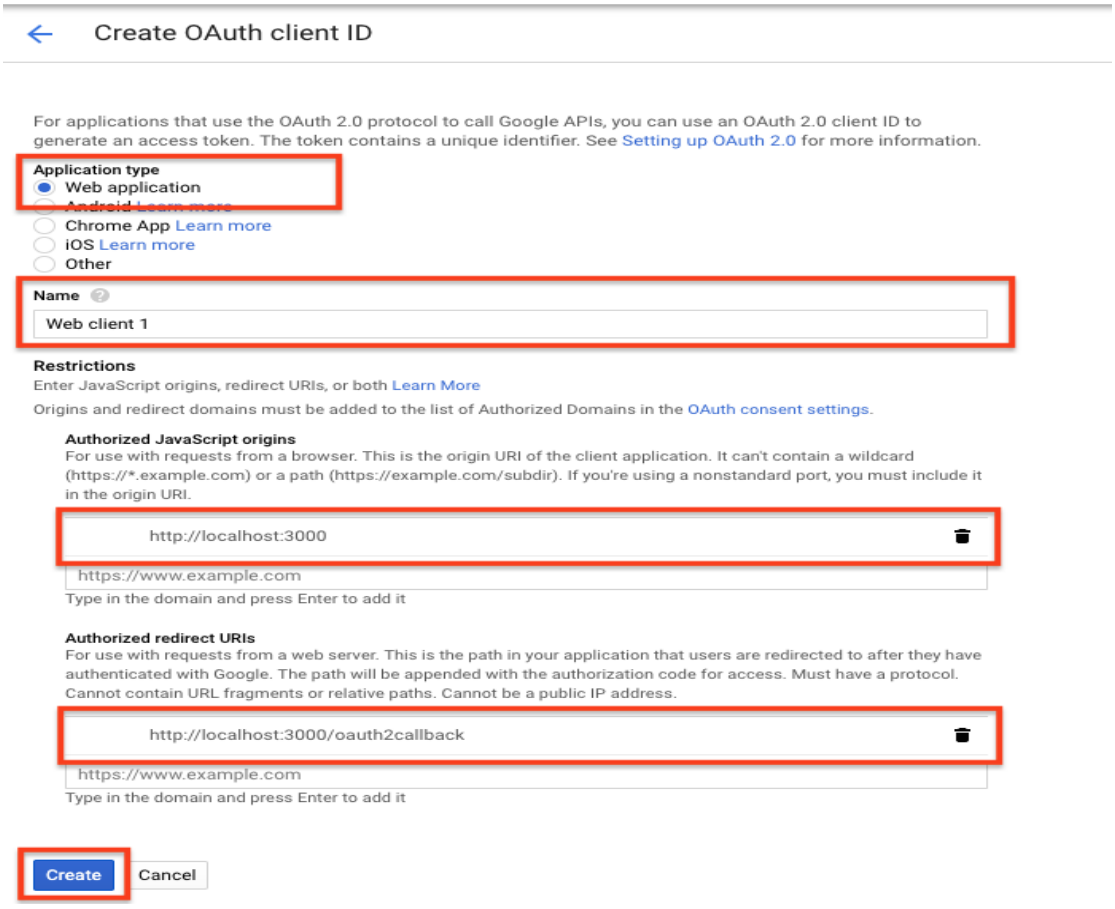
4. Give your app a name. You can also upload your app's logo, it will appear on Google's consent screen.



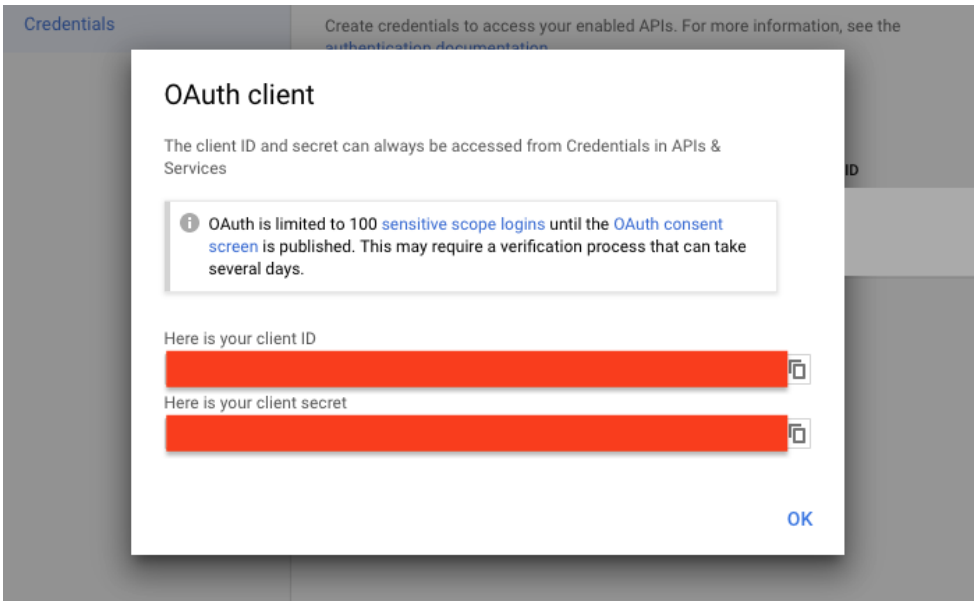
5. Create the OAuth client ID and secret.



6. Fill in the details as highlighted.



7. You now have a client ID and a client secret. You will need them later on (they are always available in your API console).



What is OAuth?

OAuth is an open-standard authorization protocol or framework that provides applications the ability for "secure designated access." For example, you can tell Facebook that it's OK for ESPN.com to access your profile or post updates to your timeline without having to give ESPN your Facebook password. This minimizes risk in a major way: In the event ESPN suffers a breach, your Facebook password remains safe.

OAuth doesn't share password data but instead uses authorization tokens to prove an identity between consumers and service providers. OAuth is an authentication protocol that allows you to approve one application interacting with another on your behalf without giving away your password.

It is commonly used by companies such as Google, Facebook, Twitter, etc. A company like Google can grant a third-party application or website access to its user's data: name, email address, profile picture, emails, etc.

Access Tokens

Access Tokens are used in token-based authentication to allow an application to access an API. The application receives an Access Token after a user successfully authenticates and authorizes access, then passes the Access Token as a credential when it calls the target API. The passed token informs the API that the bearer of the token has been authorized to access the API and perform specific actions specified by the scope that was granted during authorization.

Facebook and Twitter Sign in APIs:

- ✓ https://developers.facebook.com/docs
- ✓ https://developer.twitter.com/en/docs/api-reference-index

Sources:

- ❖ https://www.varonis.com/blog/what-is-oauth/
- ❖ https://developers.google.com/identity/protocols/OAuth2InstalledApp
- ❖ https://medium.com/better-programming/log-in-with-the-google-oauth-demo-app-9e7d0e801c29
- ❖ https://auth0.com/docs/tokens/access-tokens