

Course: The Complete Full-Stack Web Development Bootcamp

 [udemy.com/course/the-complete-web-development-bootcamp/learn/lecture/41780550](https://www.udemy.com/course/the-complete-web-development-bootcamp/learn/lecture/41780550)



Sign in

to continue to Google Cloud Platform

Email or phone

[Forgot email?](#)

Not your computer? Use Guest mode to sign in privately.

[Learn more about using Guest mode](#)

[Create account](#)

Next

English (United States)



[Help](#)

[Privacy](#)

[Terms](#)

Set up your Google OAuth Credentials

Setting up "Login with Google"

In the next lesson, you will create a button that allows users to log into your website using their google account. However, you'll first need to create your own credentials to use in the project (you can't use mine from the video).

You can use your Google/gmail account for the next steps.

Sign in to your google cloud console

Navigate to <https://console.cloud.google.com/> and sign in with your Google/gmail account.



Sign in

to continue to Google Cloud Platform

Email or phone

[Forgot email?](#)

Not your computer? Use Guest mode to sign in privately.

[Learn more about using Guest mode](#)

[Create account](#)

Next

English (United States)



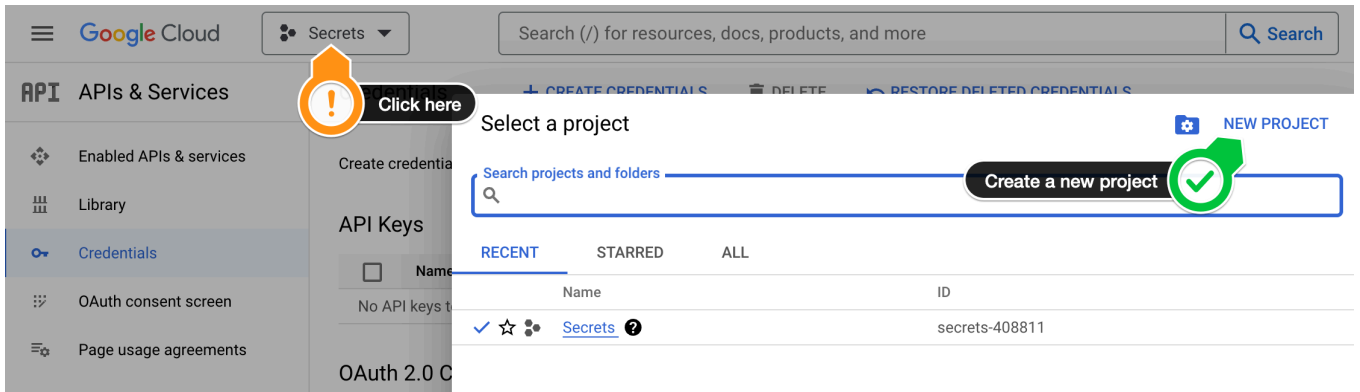
[Help](#)

[Privacy](#)

[Terms](#)

Create a new project

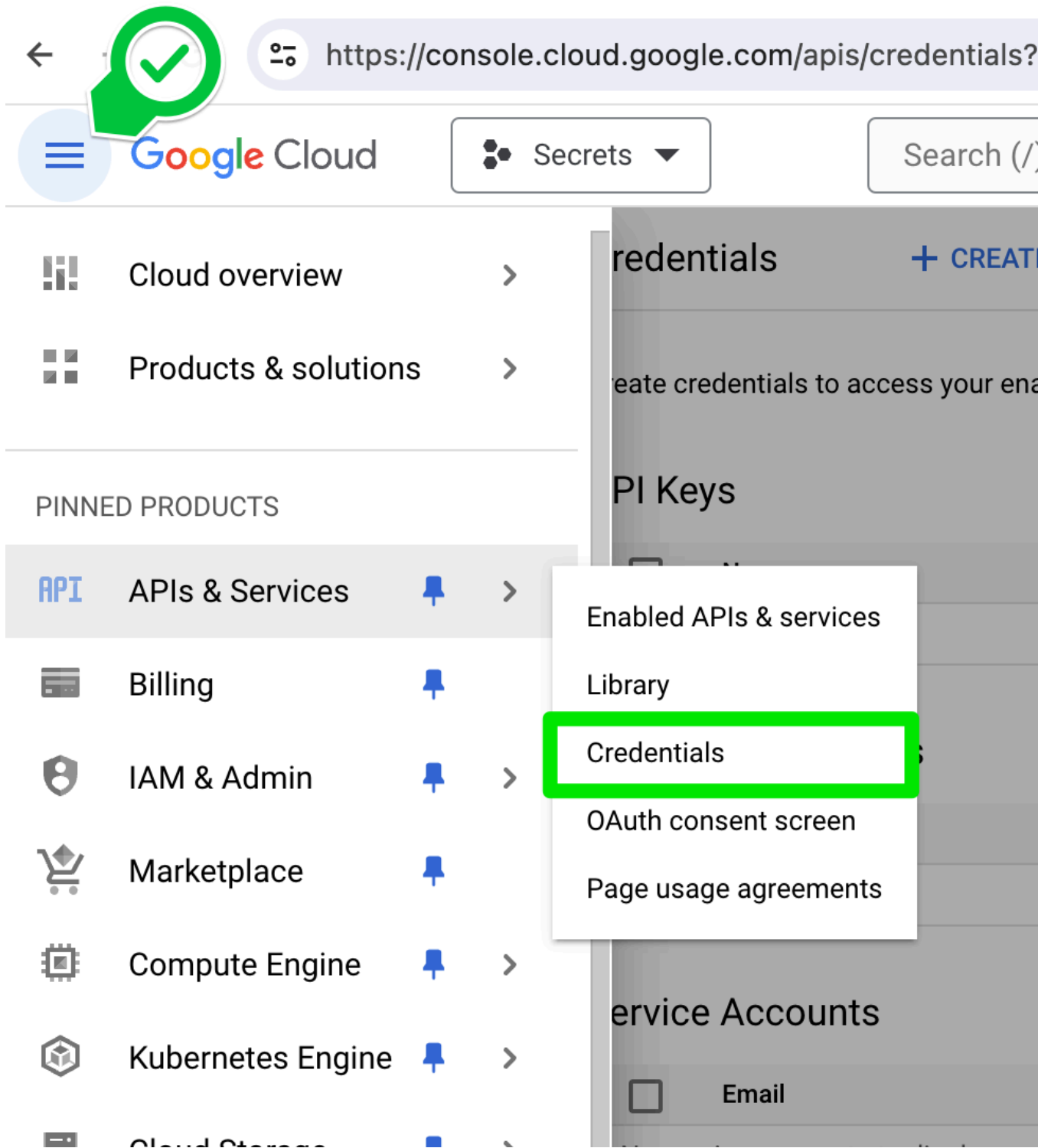
There's a drop down on the top left where you can select a project or create a new one. Create your own project called "Secrets".



You can leave the location as "No organisation".

Access "Credentials" under APIs & Services

Click on the hamburger menu and navigate to "Credentials".



The screenshot shows the Google Cloud console interface. At the top, there's a navigation bar with the Google Cloud logo, a 'Secrets' dropdown menu, and a search bar. Below the navigation bar, there's a sidebar with various product categories. The 'APIs & Services' category is highlighted, and a dropdown menu is open, showing options like 'Enabled APIs & services', 'Library', 'Credentials', 'OAuth consent screen', and 'Page usage agreements'. The 'Credentials' option is highlighted with a green rectangular box. In the background, the 'Credentials' page is partially visible, showing a '+ CREATE' button and a section for 'API Keys'.

Configure Consent Screen

Before you can create your credentials you'll have to configure the consent screen. You'll be prompted to do this when you click "Create OAuth Client ID", but there's also a tab on the side where you can do this right away.

APIs & Services



Enabled APIs & services



Library



Credentials



OAuth consent screen



Page usage agreements


You'll get taken to a form and will have to provide some information about yourself and your app. You're basically configuring the screen that the user sees when they are in the process of granting your app access to their information.

OAuth consent screen


Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.

User Type

☐ Internal 

Only available to users within your organization. You will not need to submit your app for verification. [Learn more about user type](#) 

☒ External 

Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. [Learn more about user type](#) 


CREATE

Select "External" for the application type. We be staying in testing mode, so we won't need to go through the verification process for our app.

Fill in the required information like the app name, the support email, and the developer email addresses. Then click continue. You don't need to provide a domain or logo.

Developer contact information

Email addresses *

appbreweryinfo@gmail.com  |

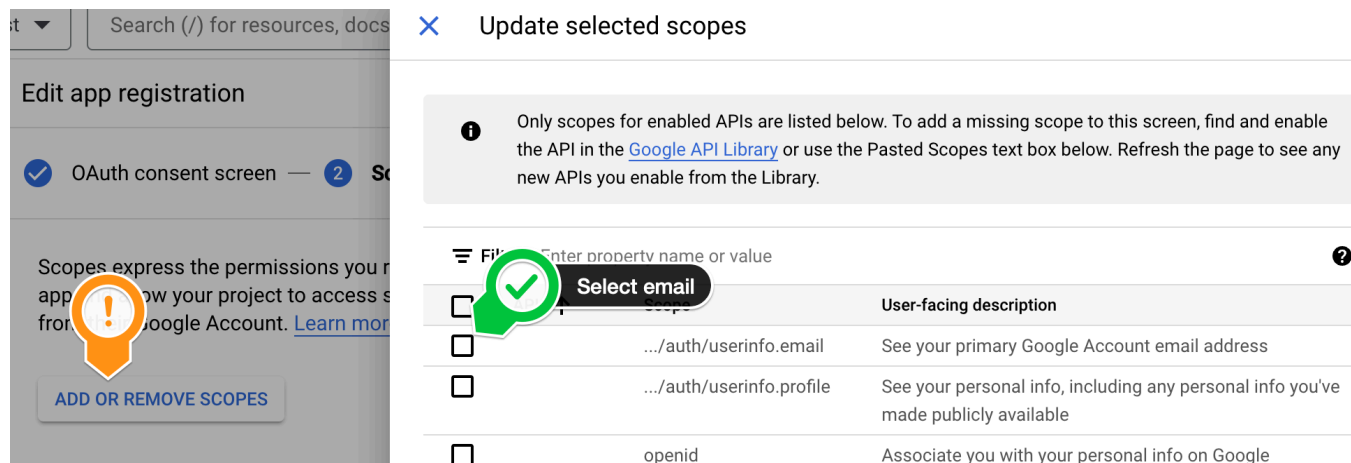
These email addresses are for Google to notify you about any changes to your project.

SAVE AND CONTINUE

CANCEL

Configure your Scope

Scopes are the fields that you will receive once the user logs in through google. We're just interested in the email so click on "Add Or Remove Scopes". And add a checkmark:



Search (/) for resources, docs

Update selected scopes

Only scopes for enabled APIs are listed below. To add a missing scope to this screen, find and enable the API in the [Google API Library](#) or use the Pasted Scopes text box below. Refresh the page to see any new APIs you enable from the Library.

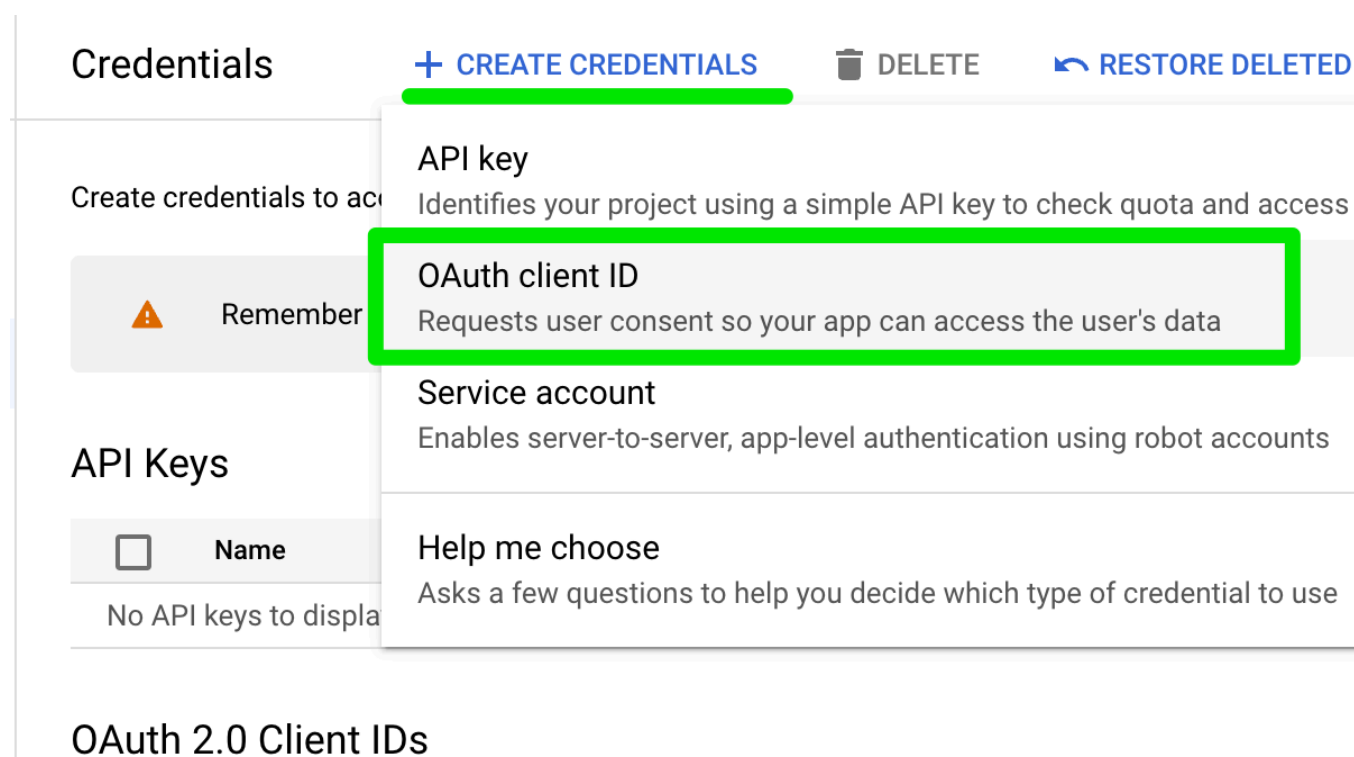
Filter: Enter property name or value

		User-facing description
<input checked="" type="checkbox"/>	Select email	
<input type="checkbox"/>	.../auth/userinfo.email	See your primary Google Account email address
<input type="checkbox"/>	.../auth/userinfo.profile	See your personal info, including any personal info you've made publicly available
<input type="checkbox"/>	openid	Associate you with your personal info on Google

You don't need to add any test users.

Create your client ID

Once your done, navigate back to "Credentials", click "Create Credentials - OAuth client ID.



Credentials

+ CREATE CREDENTIALS DELETE RESTORE DELETED

Create credentials to access your project

Remember

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
Enables server-to-server, app-level authentication using robot accounts

API Keys

Name

No API keys to display

Help me choose
Asks a few questions to help you decide which type of credential to use

OAuth 2.0 Client IDs

and choose Web Application

Add localhost to your Javascript origins

use:

http://localhost:3000

Authorized JavaScript origins

For use with requests from a browser

URIs 1 *

http://localhost:3000

 ADD URI

Add localhost to your redirect URI

use:

http://localhost:3000/auth/google/secrets

This is the path where your users are redirected to after they have authenticated with google. You basically want to show your logged in users the secret page, so that's what you need to configure here.

Authorized redirect URIs

For use with requests from a web server

URIs 1 *

http://localhost:3000/auth/google/secrets



 ADD URI

and finally click "**Create**" to create your client ID.

Copy your Client ID and Client Secret

Make a note of these two, as you'll need them in the next lesson

Secrets

Search (/) for resources, docs, products, and more

Search

1

Credentials

+ CREATE CREDENTIALS

DELETE

RESTORE DELETED CREDENTIALS

Create credentials to access your enabled APIs. [Learn more](#)

API Keys

Name

Creation date

Restrictions

Act

No API keys to display

OAuth 2.0 Client IDs

Name

Creation date

Type

Client ID

Act

Secrets

Dec 21, 2023

Web application

368592561563-r1vf...

Client ID for Web application

DELETE

Name *

Secrets

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 [authorized domains](#).

Authorized JavaScript origins

For use with requests from a browser

URIs 1 *

http://localhost:3000

+ ADD URI

Additional information

Client ID	<div></div> 368592561563-rlvff0g80mlq5rhk8qaqmc0cp4des4ub.apps.googleusercontent.com
Creation date	December 21, 2023 at 11:24:49 AM GMT+0

Client secrets

If you are in the process of changing client secrets, you can manually rotate them without downtime. [Learn more](#)

Client secret	<div></div> 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
Creation date	December 21, 2023 at 11:24:49 AM GMT+0
Status	<div></div> Enabled