

Setting up access to the YouTube API

Julian Kohne
Johannes Breuer
M. Rohangis Mohseni

Setting up API access to the *YouTube* API

To effectively work with *YouTube* data, we need access to its Application Programming Interface (API). As this process, unfortunately, can be a bit tedious, we prepared this preliminary tutorial so you don't have to spend time setting up your API access during the workshop. Please follow this tutorial and check if the setup works before the workshop. If your setup does not work, feel free to contact us before the workshop (the sooner the better) and we'll try to assist you in setting up access to the *YouTube* API. If, for some reason, setting up access to the API does not work for you and we don't find a solution or if you cannot access the API during the workshop, we will have some data prepared that you can then work with in the workshop.

HINT: Pink texts are hyperlinks to the respective website that you can simply click on

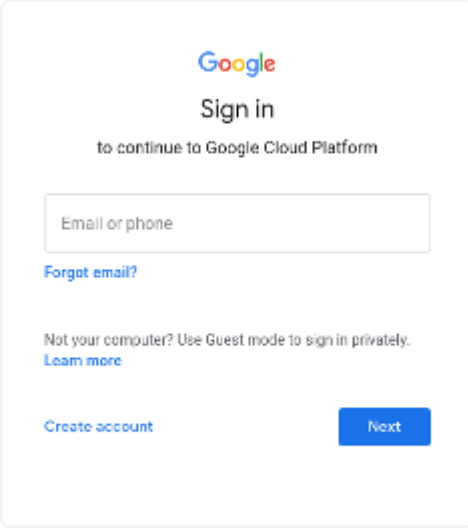
Setting up API Access

- For starters, you need a *Google* account
- You can use an existing one or create a new one
- We suggest that you create a new account (if you accidentally share your login credentials, people using the credentials only have access to this account and not your personal or work account)
- You can create a new account [here](#)
- Next, we need to:
 - Sign up for the *Google* developer console
 - Create a new project
 - Activate the *YouTube* Data API
 - Create authentication credentials
 - Test the credentials

We will explain these steps in detail on the following slides.

Google Developer Console

- Go to the **Developer Console** and log in with your (new) *Google* Account



Google

Sign in

to continue to Google Cloud Platform

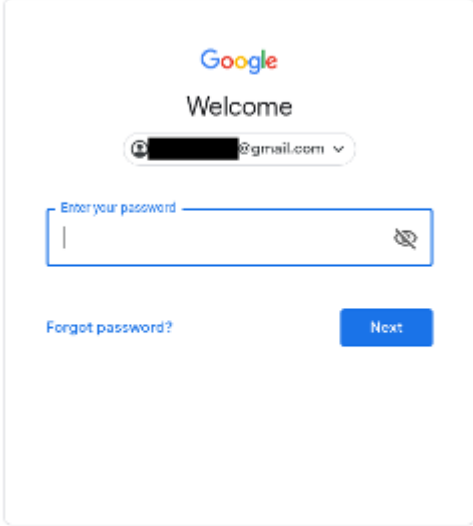
Email or phone

[Forgot email?](#)

Not your computer? Use Guest mode to sign in privately.
[Learn more](#)

[Create account](#) [Next](#)

English (United States) ▾ [Help](#) [Privacy](#) [Terms](#)



Google

Welcome

📧 [redacted]@gmail.com ▾

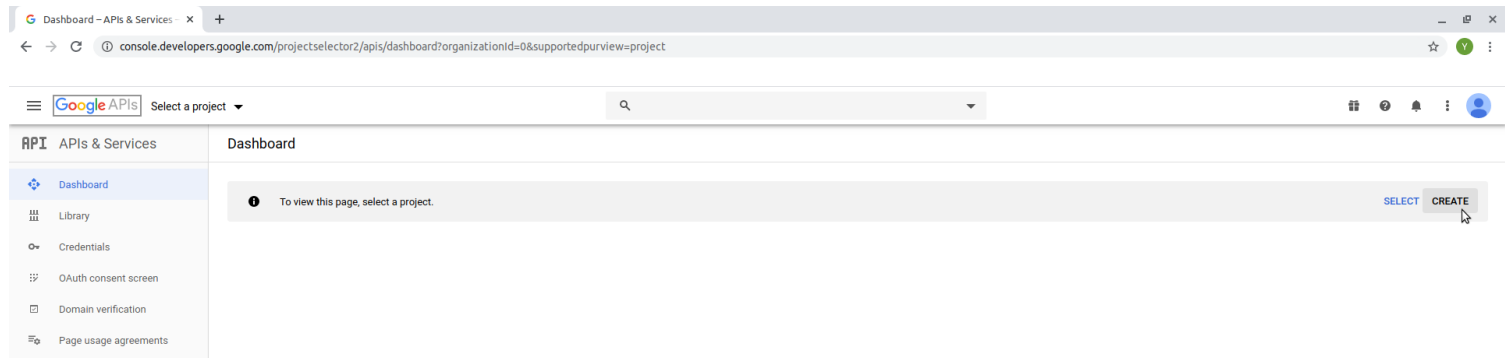
Enter your password

[Forgot password?](#) [Next](#)

English (United States) ▾ [Help](#) [Privacy](#) [Terms](#)

Google Developer Console

- Create a new project by clicking the "Create" button in the top right corner

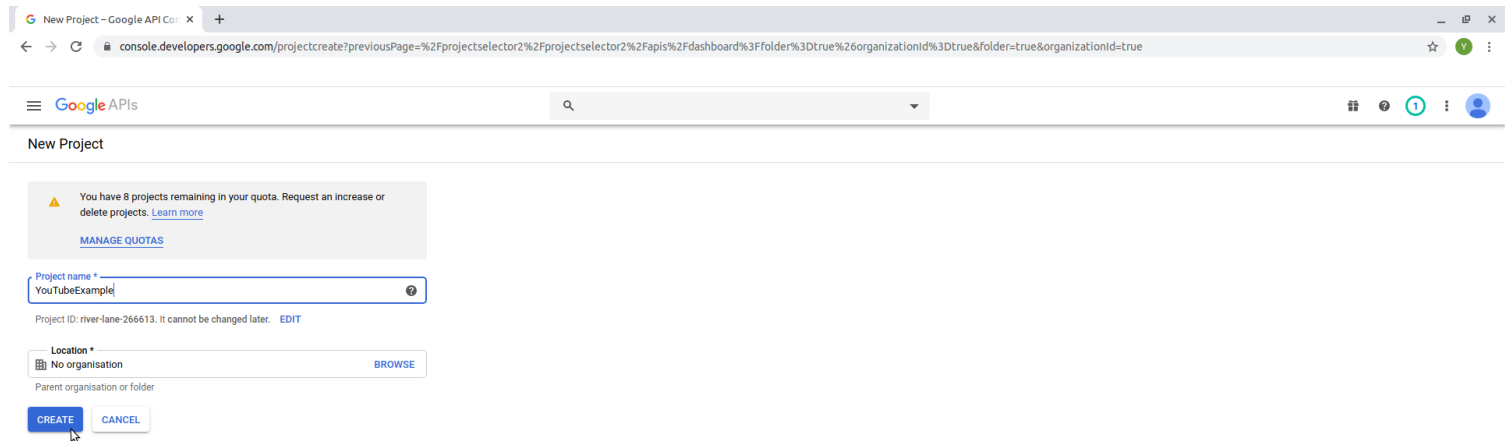


Project limit

- You only have a limited number of Projects that you can create, so be careful not to constantly delete and create new ones!
- More information on project limits can be found [here](#)

Creating a New Project

- Specify a project name (you do not have to specify an organisation) and click "create" once you are done



New Project – Google API Console

console.developers.google.com/projectcreate?previousPage=%2Fprojectselector2%2Fprojectselector2%2Fapis%2Fdashboard%3Ffolder%3Dtrue%26organizationid%3Dtrue&folder=true&organizationid=true

Google APIs

New Project

You have 8 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
YouTubeExample

Project ID: river-lane-266613. It cannot be changed later. [EDIT](#)

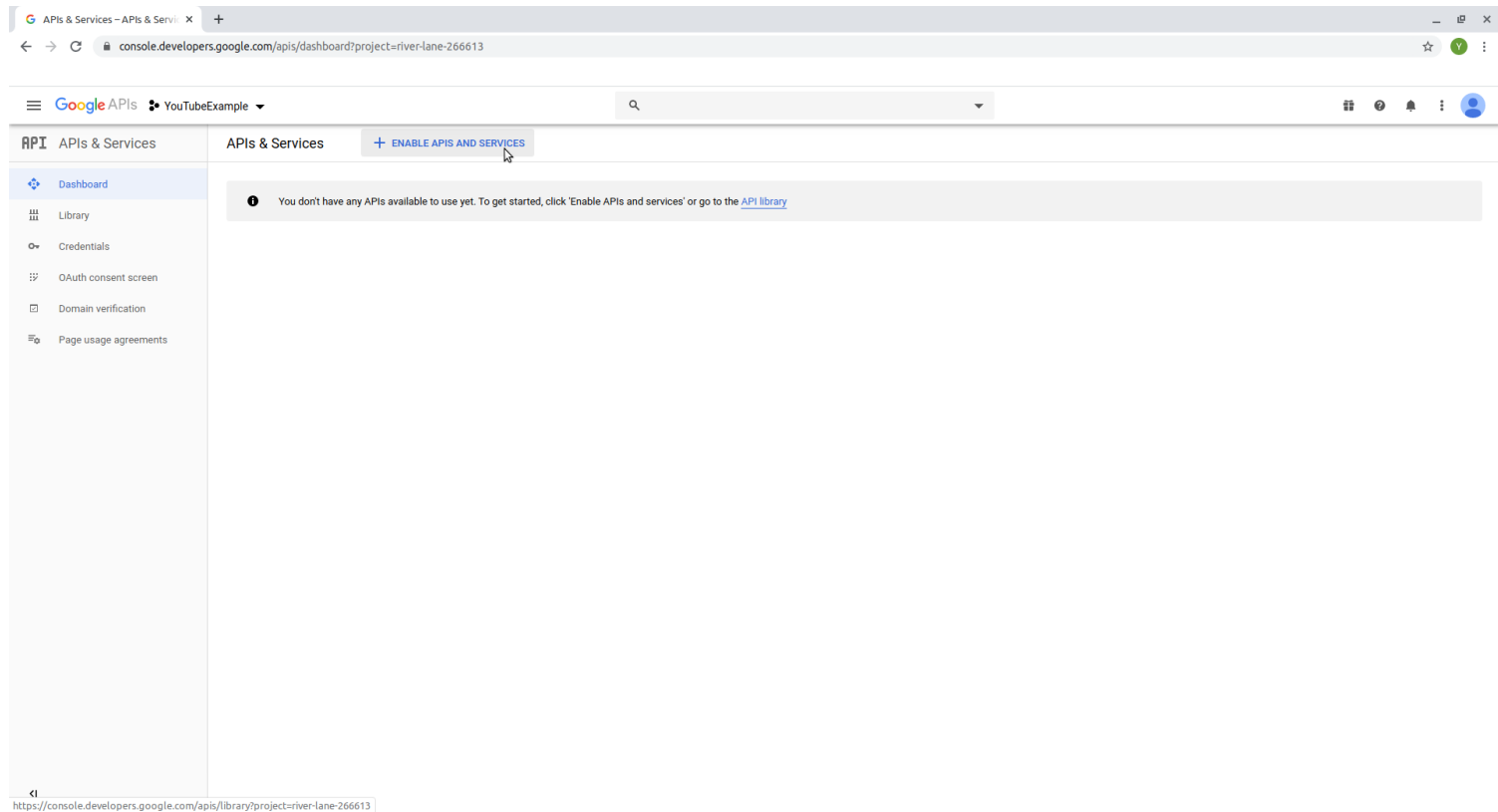
Location *
No organisation [BROWSE](#)

Parent organisation or folder

[CREATE](#) [CANCEL](#)

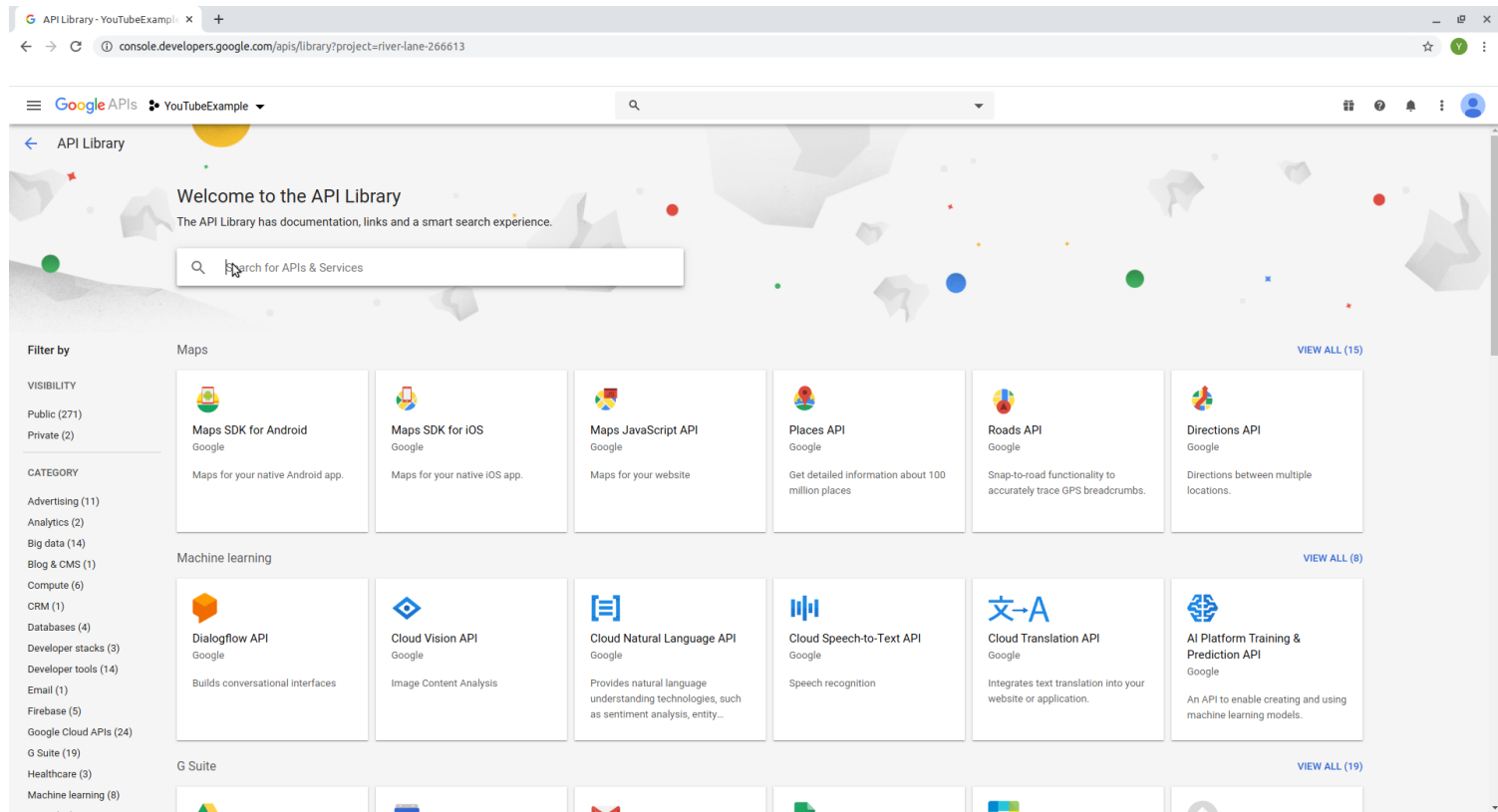
Activate the YouTube Data v3 API

- Click the button "Enable APIs and Services"



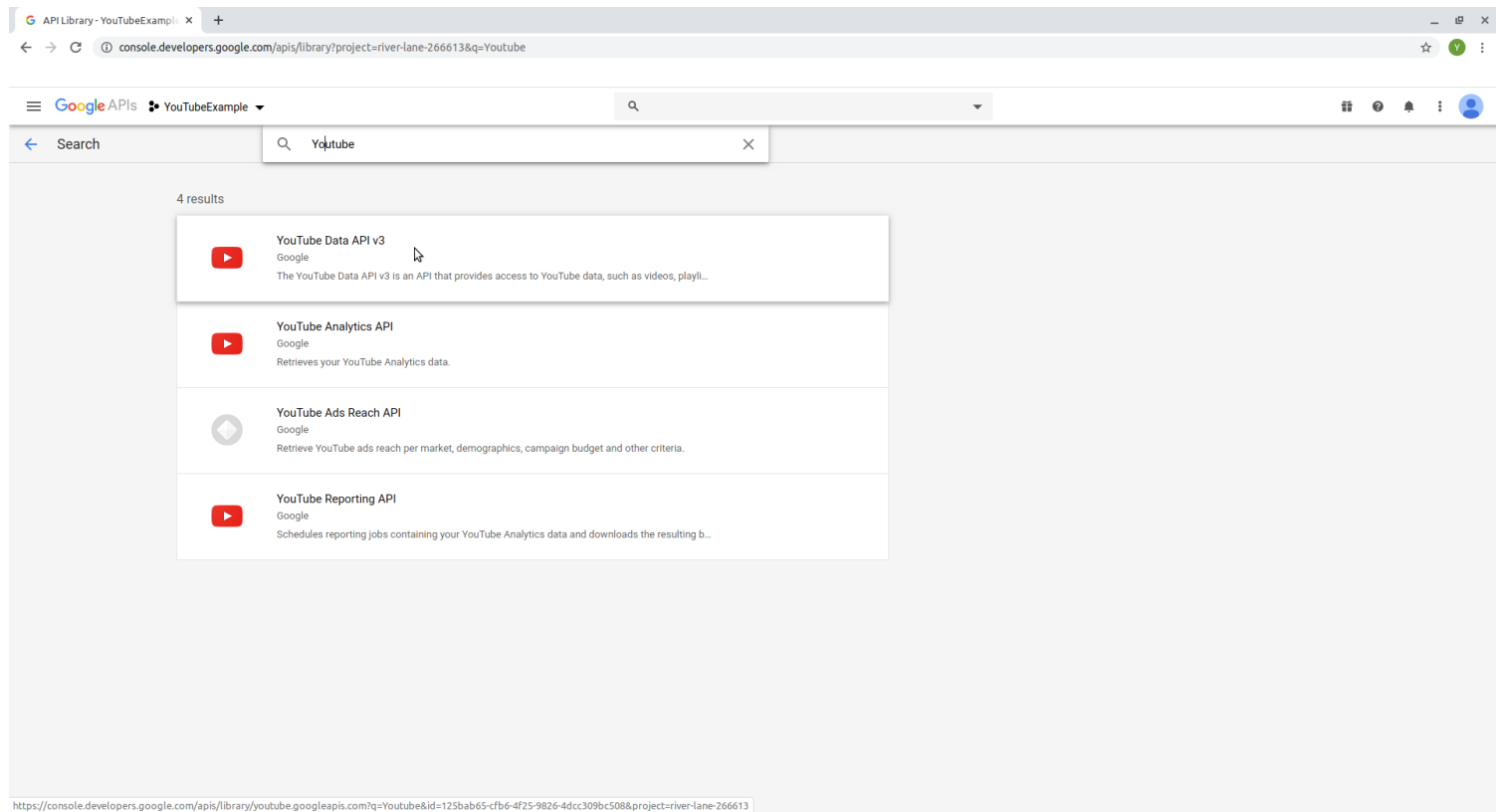
Activate the YouTube Data v3 API

- Enter "Youtube" in the search bar



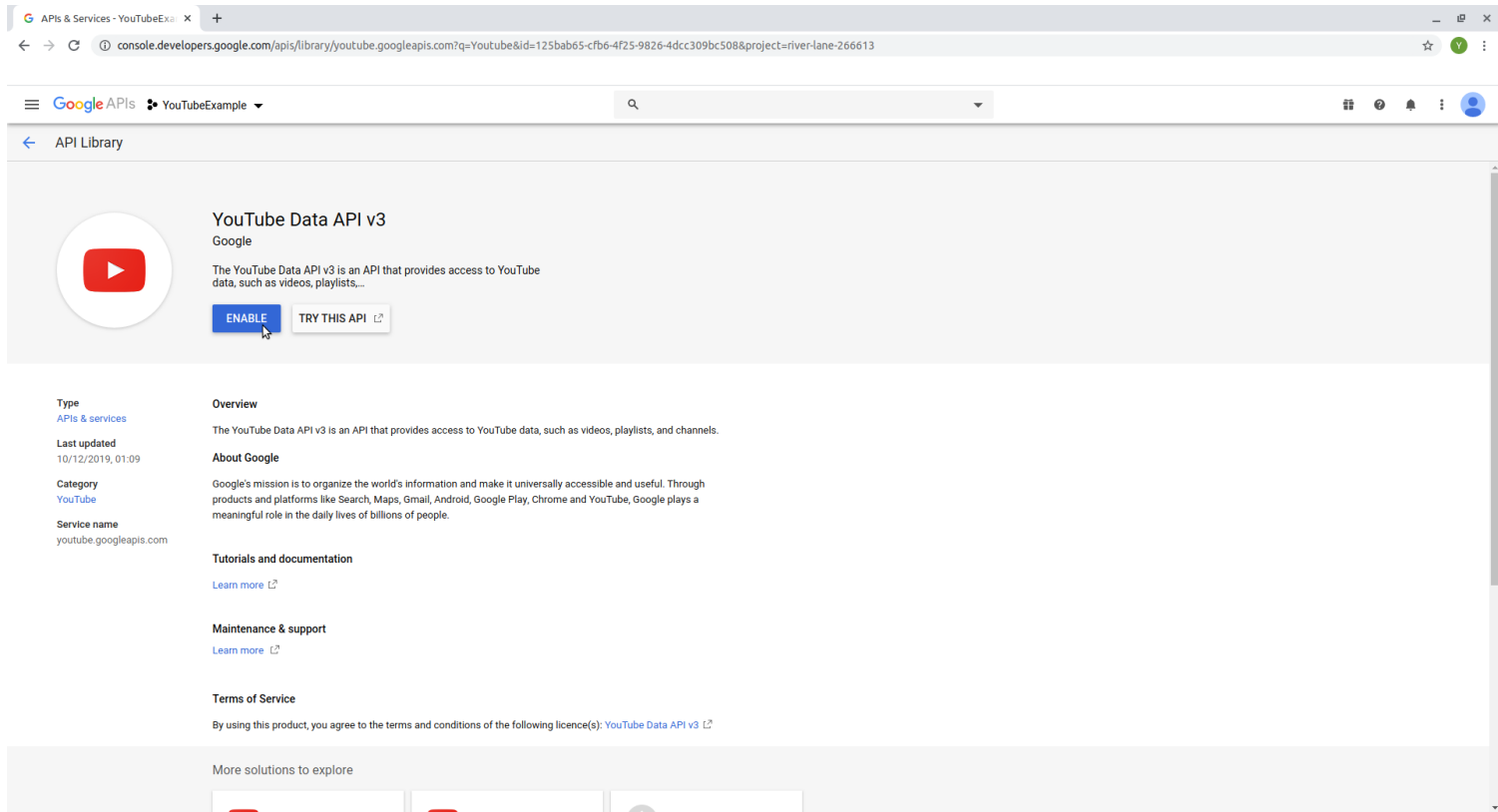
Activate the YouTube Data v3 API

- Click on "YouTube Data API v3"



Activate the YouTube Data v3 API

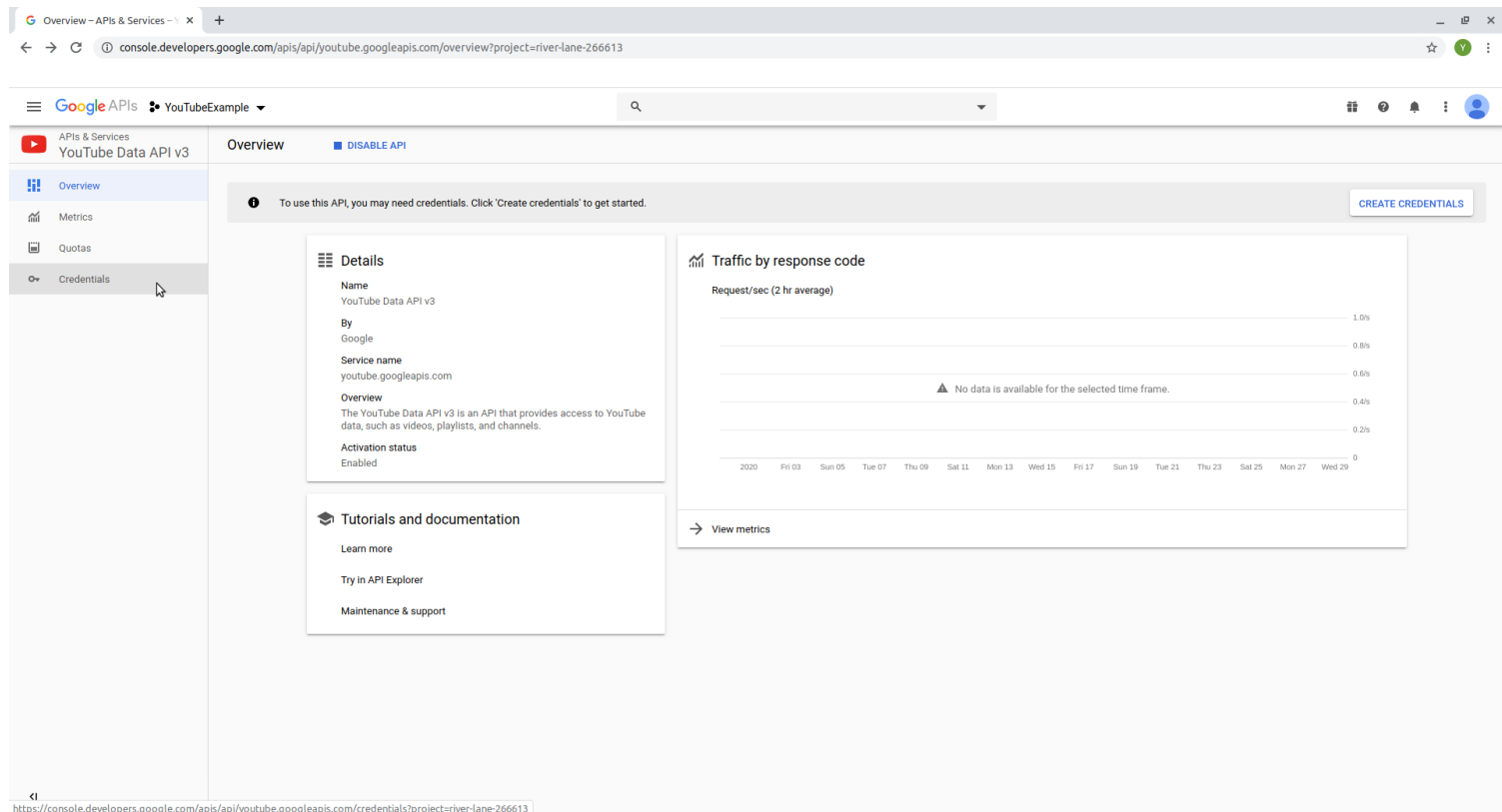
- Activate the API by clicking the "Enable" button



The screenshot shows the Google APIs console interface. At the top, the browser address bar displays the URL: `console.developers.google.com/apis/library/youtube.googleapis.com?q=Youtube&id=125bab65-cfb6-4f25-9826-4dcc309bc508&project=river-lane-266613`. The console header shows "Google APIs" and "YouTubeExample". The main content area is titled "API Library" and features the "YouTube Data API v3" by Google. A large red YouTube play button icon is on the left. To the right, the text reads: "The YouTube Data API v3 is an API that provides access to YouTube data, such as videos, playlists,..." Below this text are two buttons: a blue "ENABLE" button and a grey "TRY THIS API" button with an external link icon. A mouse cursor is hovering over the "ENABLE" button. On the left sidebar, under "APIs & services", the "Last updated" date is "10/12/2019, 01:09", the "Category" is "YouTube", and the "Service name" is "youtube.googleapis.com". The main content area has sections for "Overview" (describing the API's purpose), "About Google" (stating Google's mission), "Tutorials and documentation" (with a "Learn more" link), "Maintenance & support" (with a "Learn more" link), and "Terms of Service" (stating that using the product implies agreement to the terms of the "YouTube Data API v3" license, with a "Learn more" link). At the bottom, there is a section titled "More solutions to explore" with three placeholder cards.

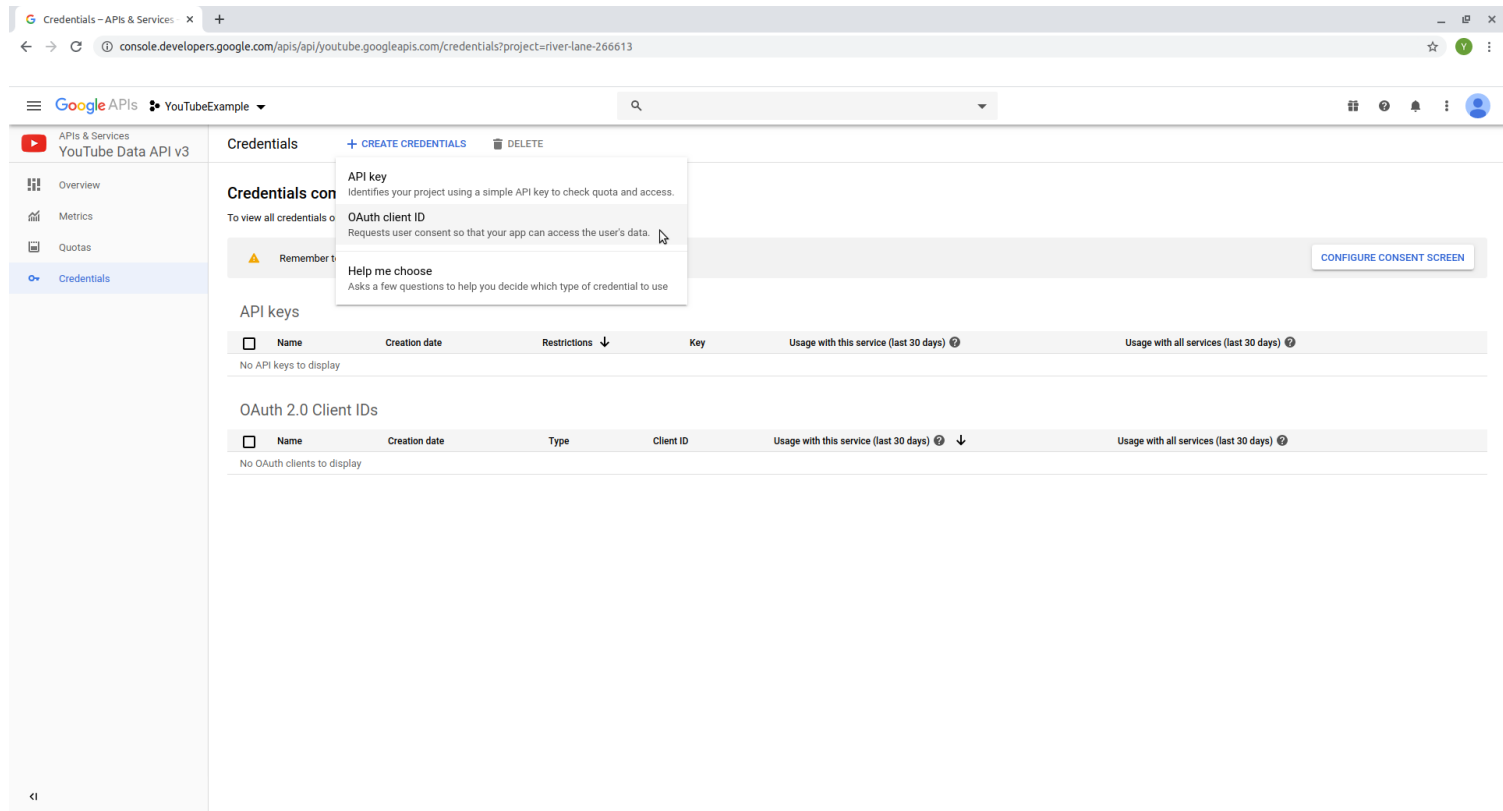
Obtaining OAuth credentials

- The API is now enabled, but we still need credentials to use it
- Click on the "Credentials" pane in the menu on the left side



Obtaining OAuth credentials

- Click the "+ Create Credentials" button and select "OAuth Client ID" from the dropdown menu



Obtaining OAuth credentials

- To be able to create new credentials, we first need to configure a consent screen for our "app"
- Click the blue "Configure Consent Screen" button on the right side of your screen

Configuring the consent screen

- Enter the name of your "app" in the the field "application name", scroll to the bottom of the page, and click "save"

The screenshot shows the 'OAuth consent screen' configuration page in the Google Developers console. The left sidebar contains navigation links: Dashboard, Library, Credentials, OAuth consent screen (selected), Domain verification, and Page usage agreements. The main content area is divided into two columns.

Left Column (Configuration):

- Verification status:** Not published
- Application name:** YouTubeScraPerBot
- Application logo:** Local file for upload (with a 'Browse' button)
- Support email:** [redacted]@gmail.com
- Scopes for Google APIs:** email, profile, openid (with an 'Add scope' button)
- Authorized domains:** example.com

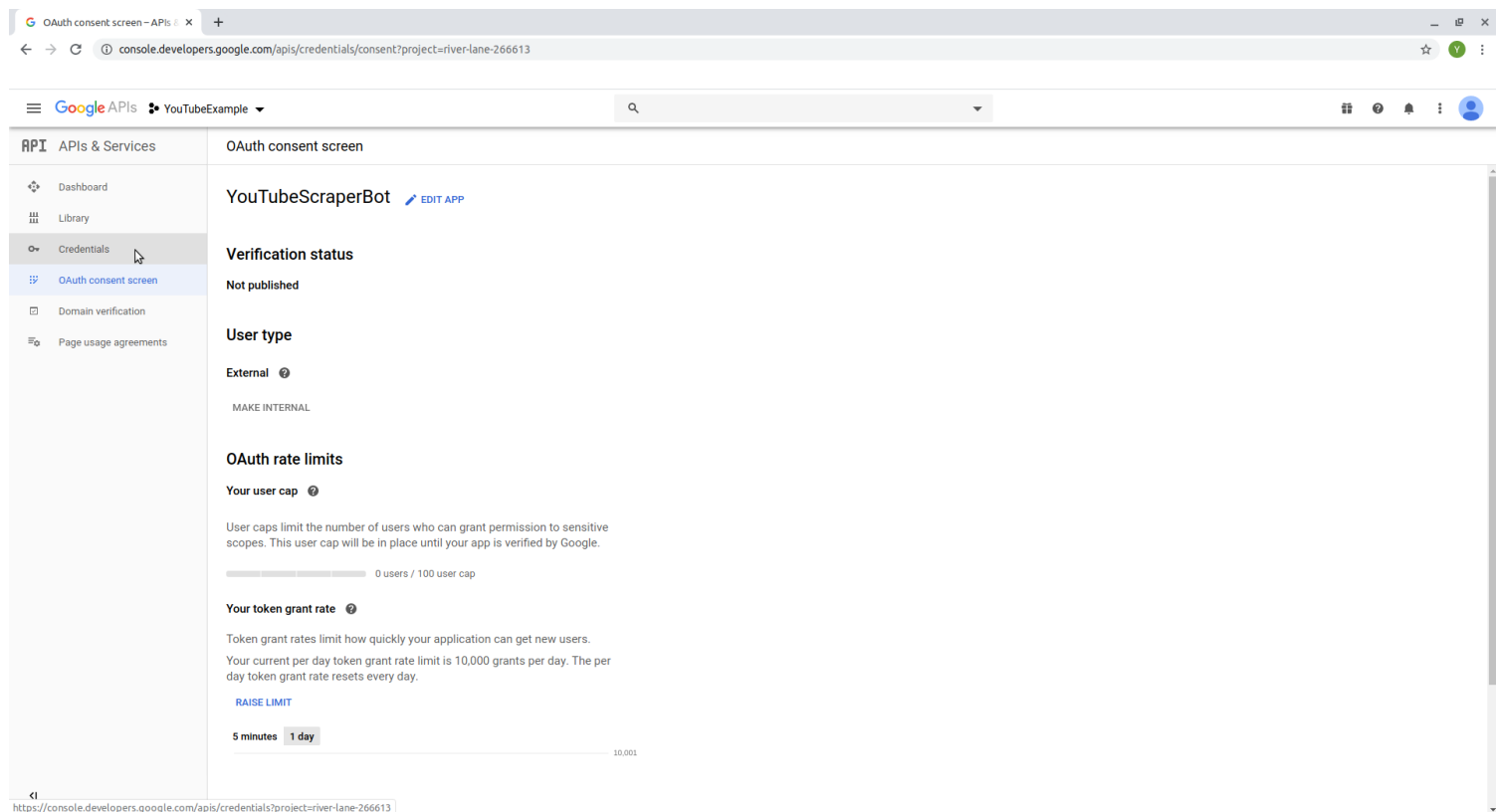
Right Column (Information):

- About the consent screen:** The consent screen tells your users who is requesting access to their data and what kind of data you're asking to access.
- OAuth verification:** To protect you and your users, your consent screen and application may need to be verified by Google. Verification is required if your app is marked as Public and at least one of the following is true:
 - Your app uses a sensitive and/or restricted scope
 - Your app displays an icon on its OAuth consent screen
 - Your app has a large number of authorised domains
 - You have made changes to a previously verified OAuth consent screen
- Token grant rate:** Your current per minute token grant rate limit is 100 grants per minute. The per minute token grant rate resets every minute. Your current per day token grant rate limit is 10,000 grants per day. The per day token grant rate resets every day.
- Raise limit:** A button to raise the limit.
- OAuth grant limits:** A table showing the current grant rate limit (100 grants per minute) and the per minute token grant rate (100 grants per minute).

At the bottom of the page, there is a 'Save' button and a 'No data for this time interval' message.

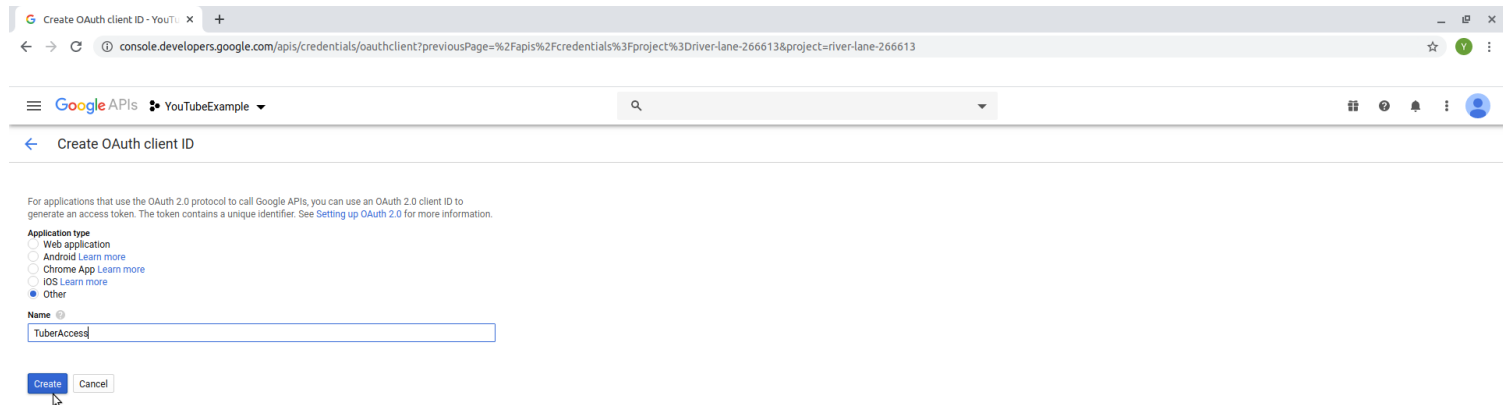
Obtaining OAuth credentials

- On the following screen, select the "Credentials" pane again from the menu on the left side of your screen



Obtaining OAuth Credentials

- Select the application type "other" and enter a name for your set of credentials. Afterwards, click "create".



Create OAuth client ID - YouTube

console.developers.google.com/apis/credentials/oauthclient?previousPage=%2Fapis%2Fcredentials%3Fproject%3Driver-lane-266613&project=river-lane-266613

Google APIs • YouTubeExample

Create OAuth client ID

For applications that use the OAuth 2.0 protocol to call Google APIs, you can use an OAuth 2.0 client ID to generate an access token. The token contains a unique identifier. See [Setting up OAuth 2.0](#) for more information.

Application type

- ☐ Web application
- ☐ Android [Learn more](#)
- ☐ Chrome App [Learn more](#)
- ☐ iOS [Learn more](#)
- ☒ Other

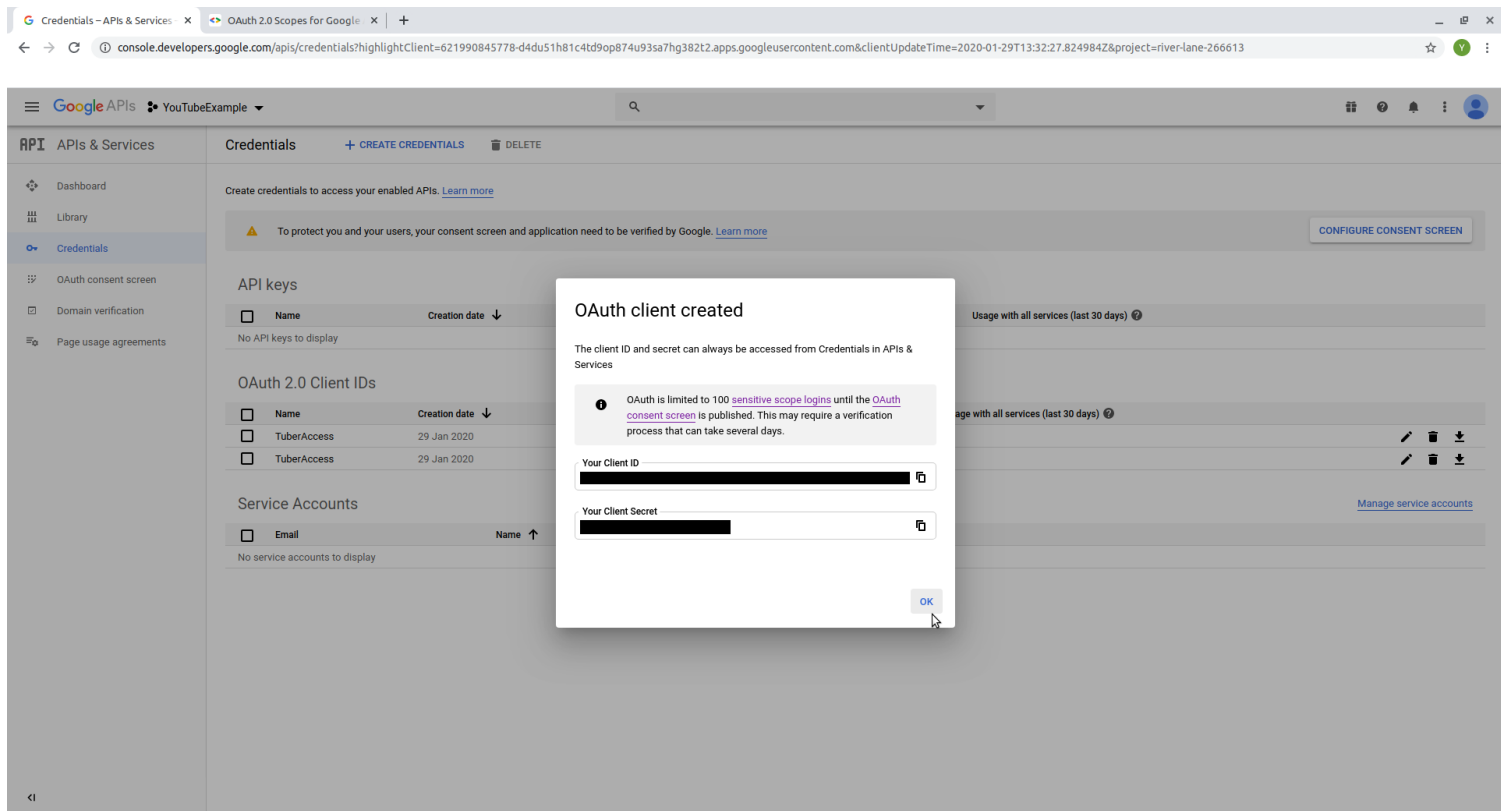
Name

TuberAccess

Create Cancel

OAuth credentials obtained

- You now have our first set of OAuth credentials for using the YouTube API
- Don't worry about closing the window, we can access them again anytime



A few words of advice on handling your *YouTube* API access

- Have one account and project ready for the workshop for which you know that it works
- Do not use your API access too excessively before the workshop to avoid any risk of it being suspended
- Never share your OAuth login credentials! This can easily happen by accident when
 - saving code in a GitHub Repo
 - sending scripts to colleagues
 - troubleshooting someone else's code
 - for added security you can use the **keyring package** (we'll say a bit more about this in the workshop)

Testing Access

- You can try whether your *YouTube* API access works as intended by running the following commands in R:

```
# installing the tuber package
install.packages("tuber")

# loading tuber package
library(tuber)

# Your Credentials (NEVER SHARE THIS INFORMATION)
ID <- "ENTER-YOUR-APP-ID"
secret <- "ENTER-YOUR-APP-SECRET"

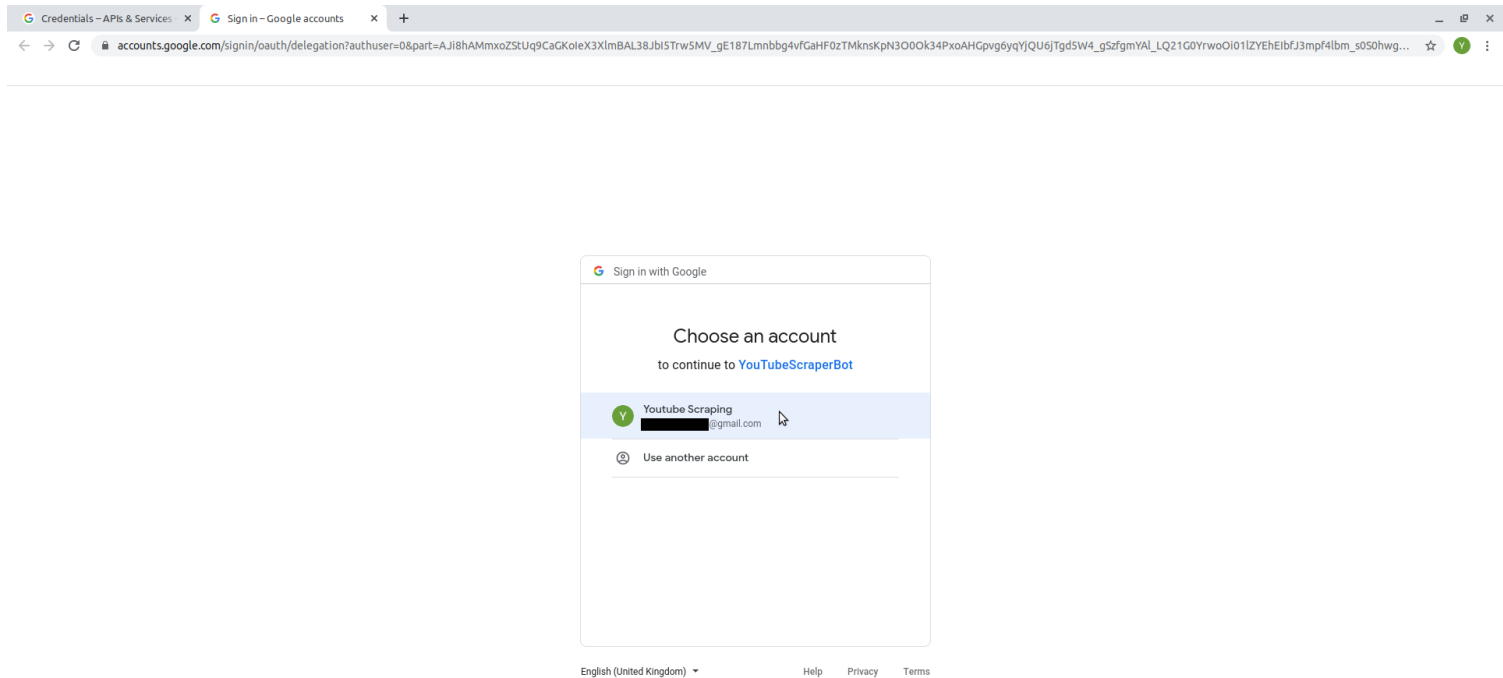
# authentication
yt_oauth(ID,secret)

# You will be asked in the R-console to
# save an access token: Select no

# You will be send to your browser to
# login with your account
```

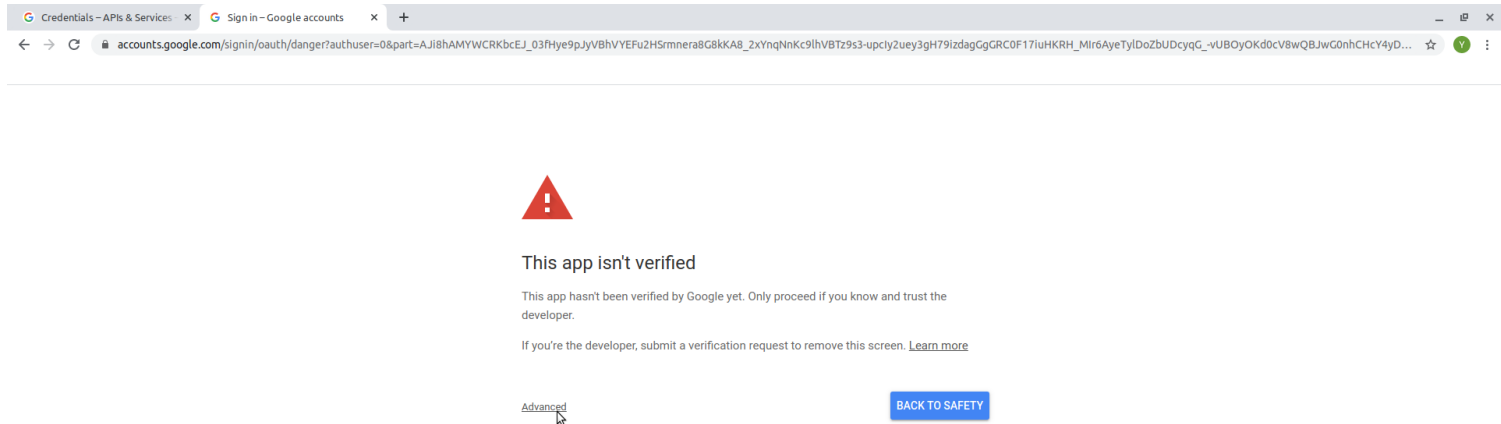
Login

- Select the *Google* Account that you used to set up the credentials



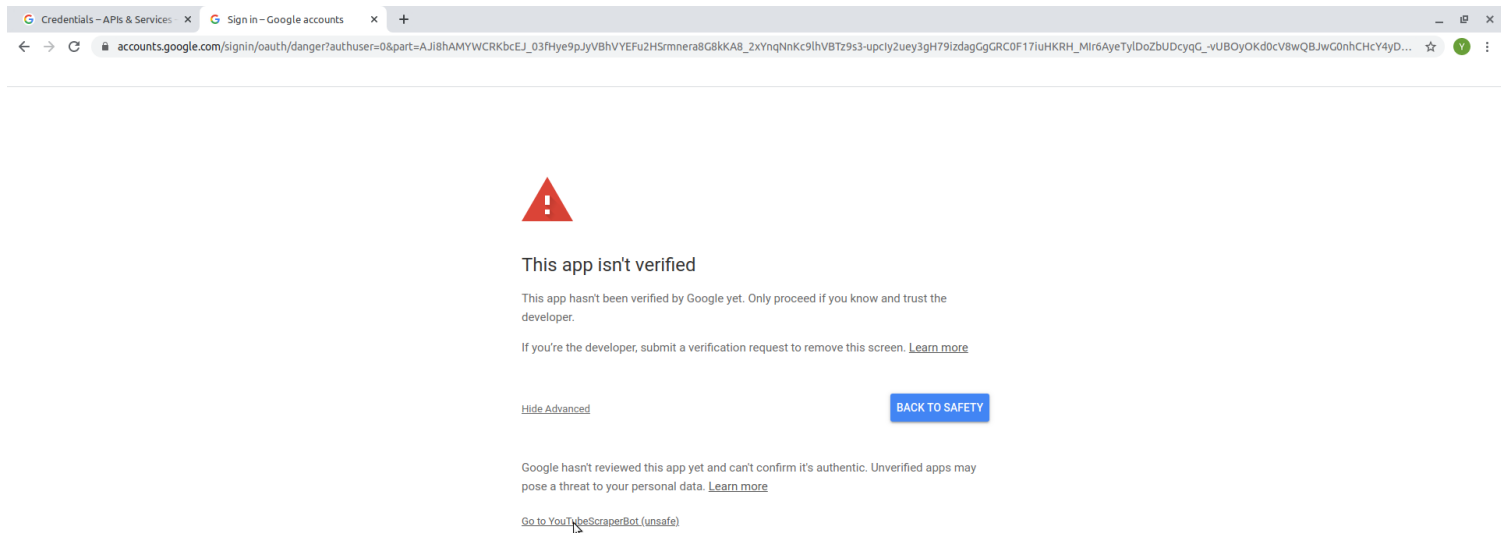
Granting access to your "unsecure app"

- As we did not verify our "App", *Google* labels it as "unsecure"
- Because we're the only ones using it, this shouldn't bother us. Click "Advanced" on the bottom left.



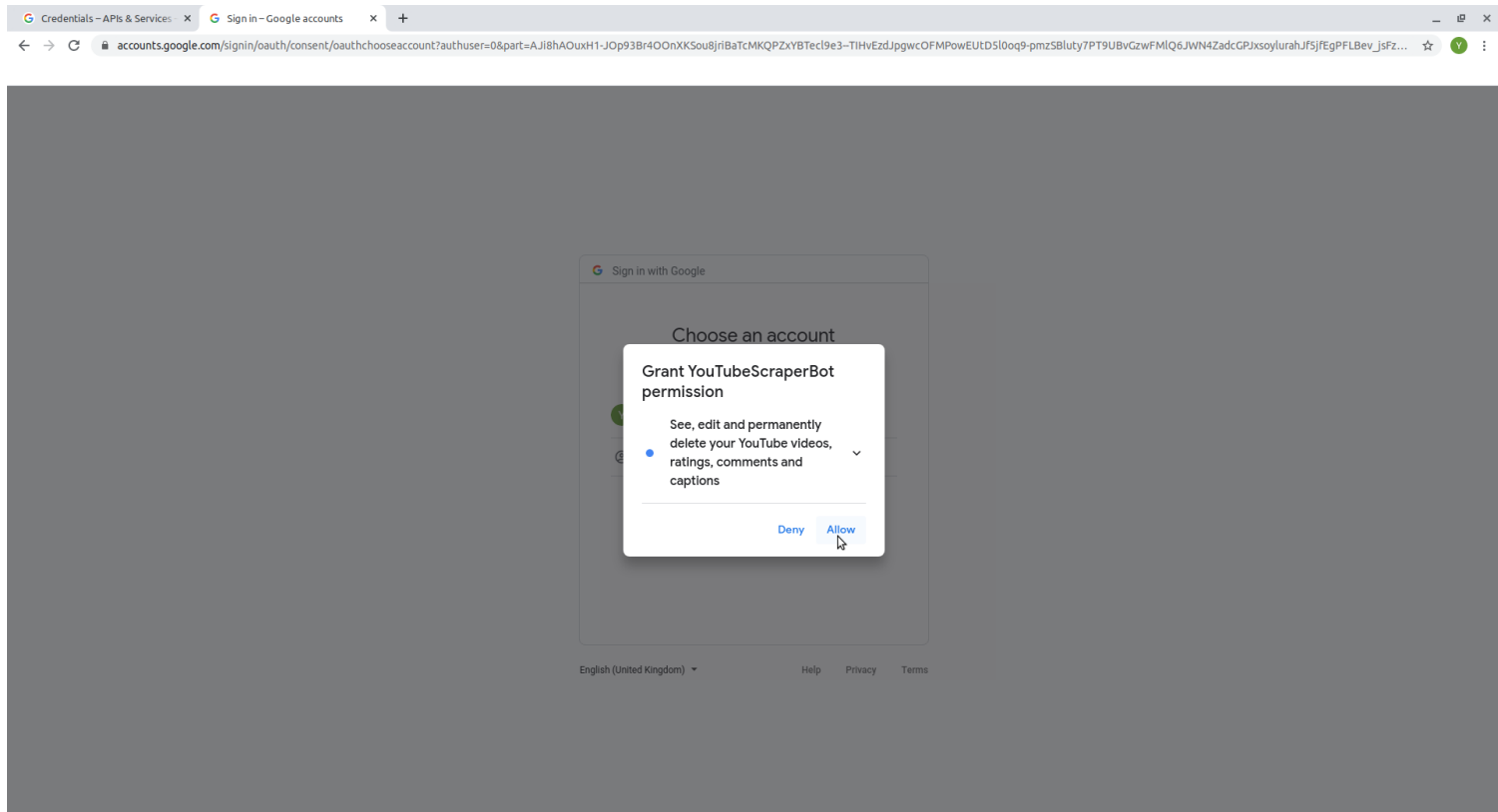
Granting access to your "unsecure app"

- Click "Go to [Name of your App] (unsafe)" on the bottom left of your screen



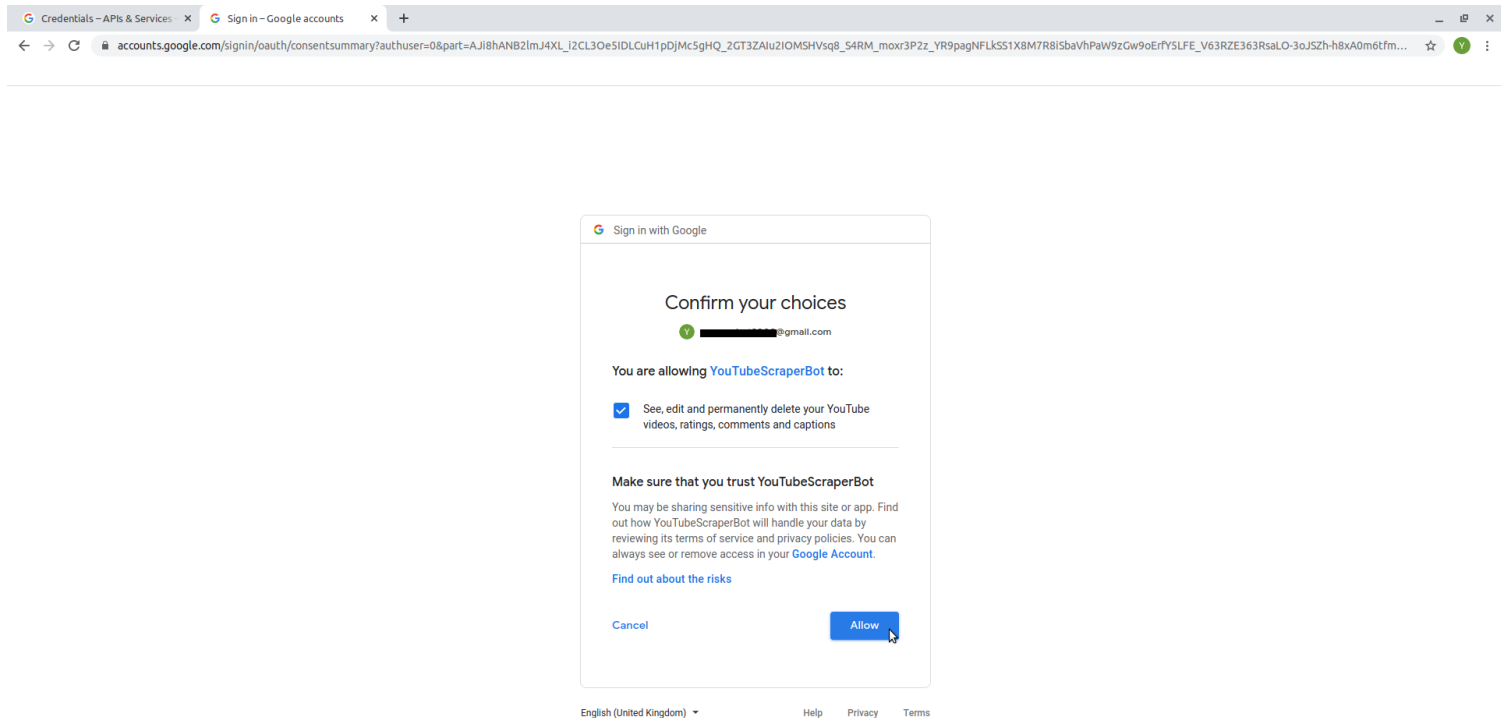
Granting access to your "unsecure app"

- Click "Allow"



Access "unsecure App"

- Click "Allow" again to confirm your choice



Access "unsecure App"

- You should now see the message "Authentication complete. Please close this page and return to R"



Test API call

- To check if everything worked, let's try to collect some data about a video

```
# get statistics of 1 video  
get_stats(video_id="HluANRwPyNo")
```

Output

- If everything worked correctly, your output should look something like this:

```
$id  
[1] "HluANRwPyNo"  
  
$viewCount  
[1] "10244939"  
  
$likeCount  
[1] "509791"  
  
$dislikeCount  
[1] "4799"  
  
$favoriteCount  
[1] "0"  
  
$commentCount  
[1] "20581"
```

**We hope that you are now all set up and
look forward to seeing you in Cologne!**