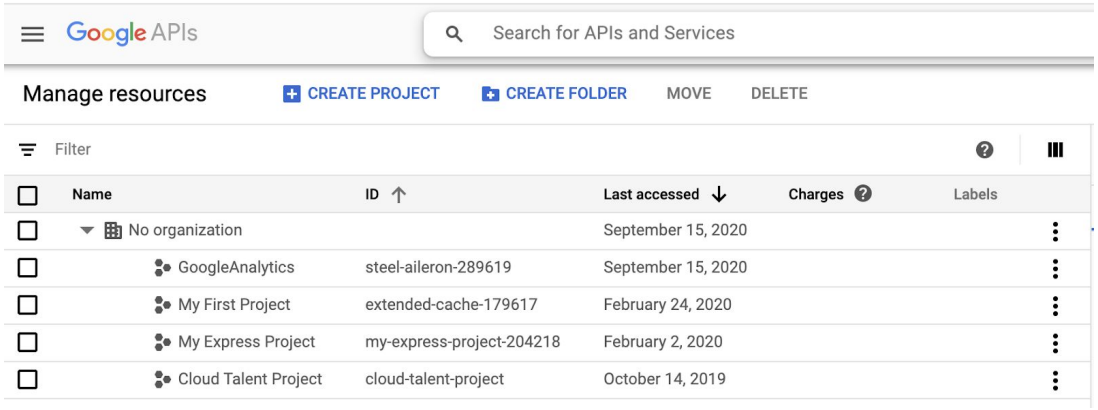
R for Web Analysts

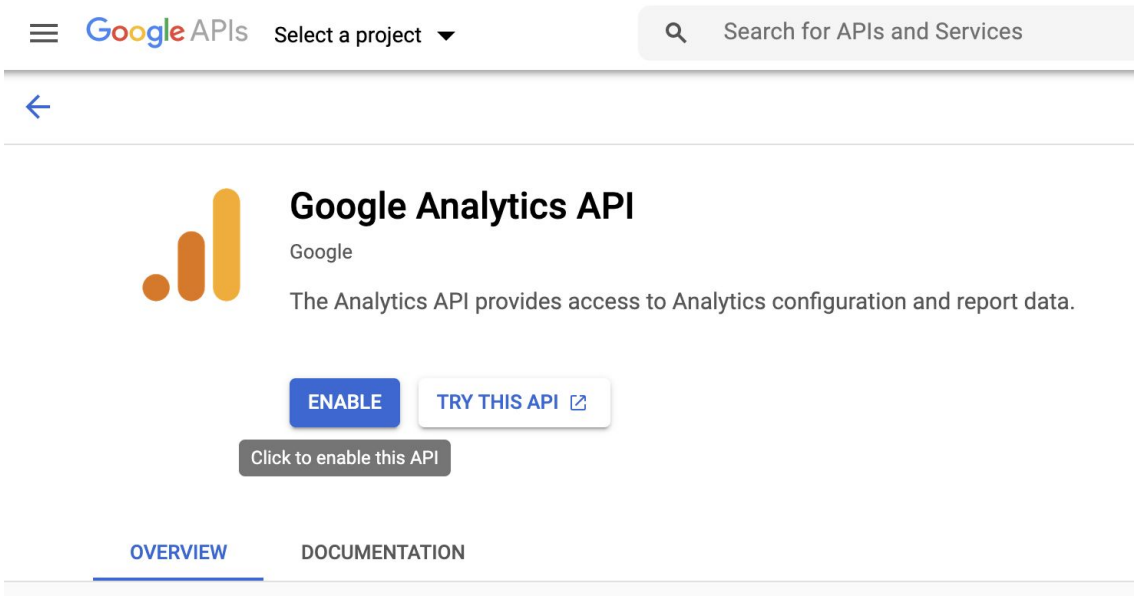
1. Go to https://console.developers.google.com/project
2. Click “create project”

   The project name and the project id are not important, but name it something meaningful. Ignore the advanced options and click “create”.

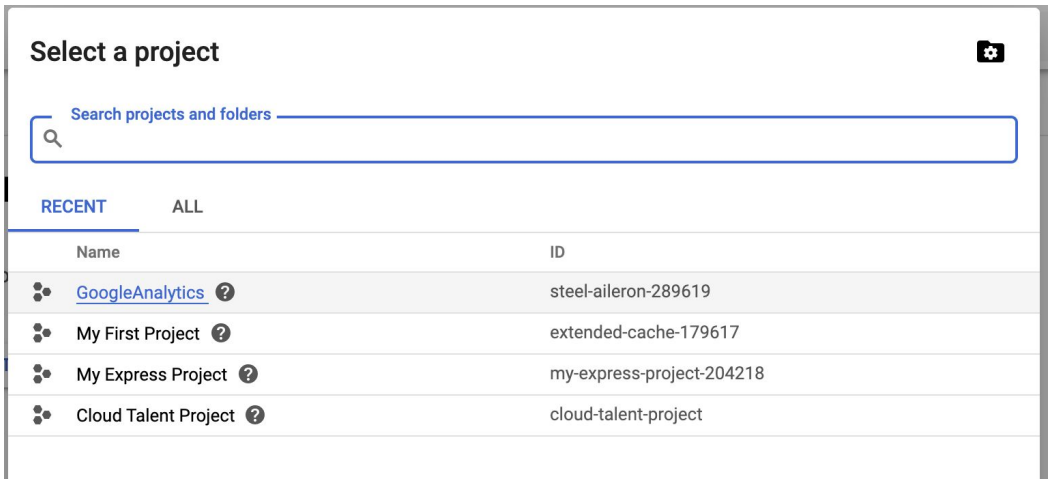
3. Wait for the project to be created.



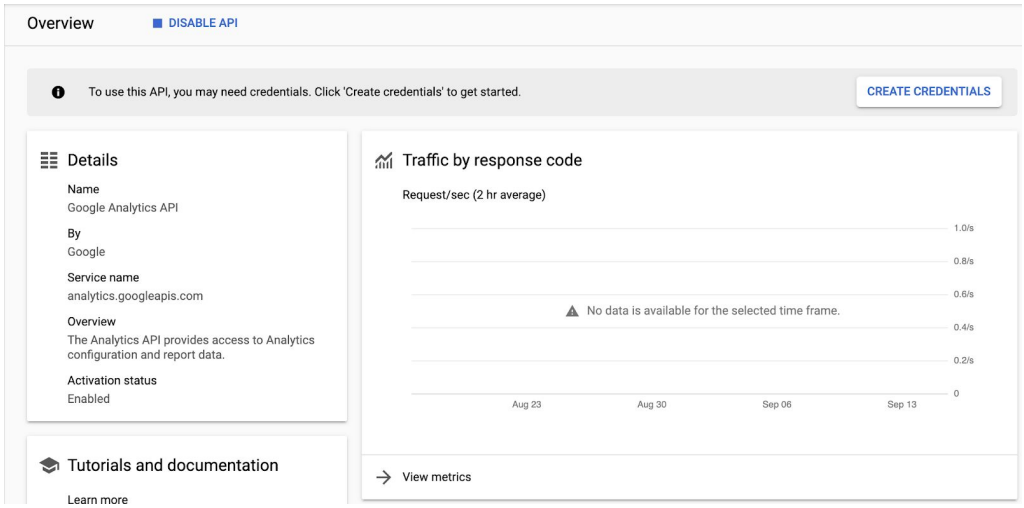
4. Search for “analytics” then enable the “Google Analytics API” by selecting it from the list and clicking “Enable”.

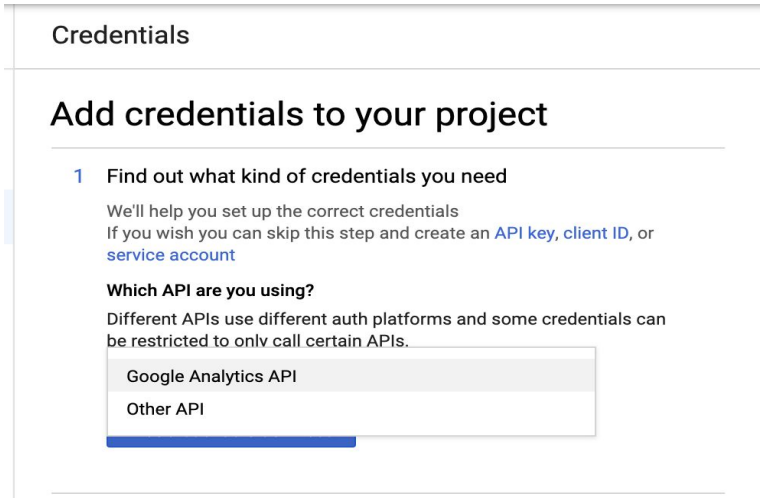


Select the project created earlier (GoogleAnalytics in my case)



5. Click “Create Credentials”. In the 2nd screen, select Google Analytics API.

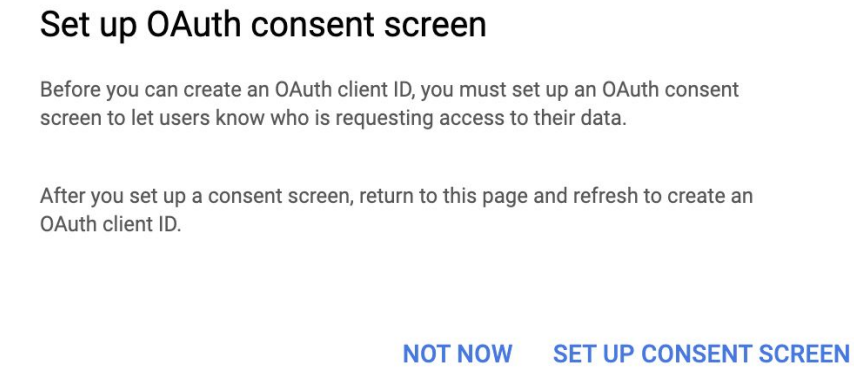




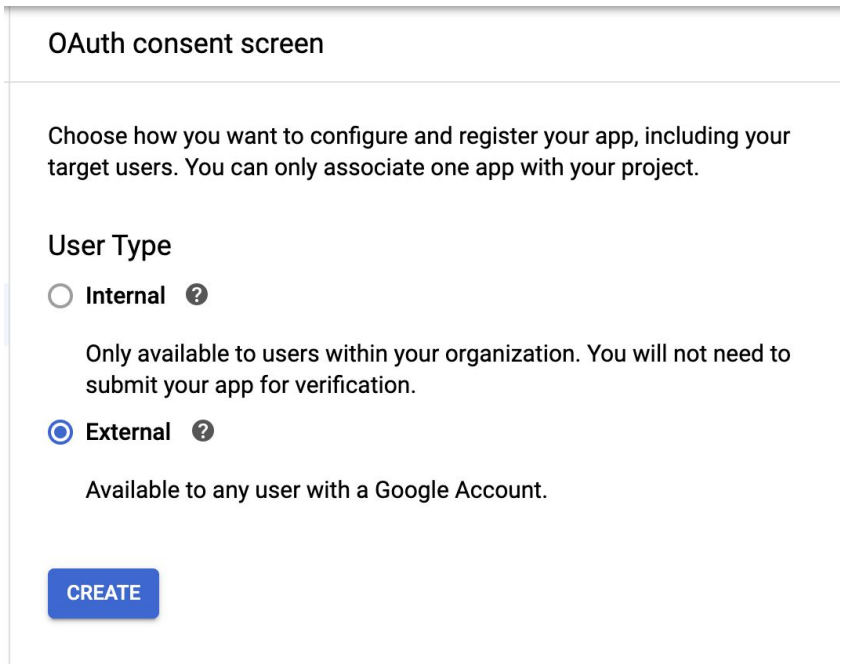
a. Select "Other UI (e.g. Windows, CLI tool)" in the "**Where will you be calling the API from**?"
b. Check "User data" for the "**What data will you be accessing**" field.

6. Click "What credentials do I need?" to proceed.

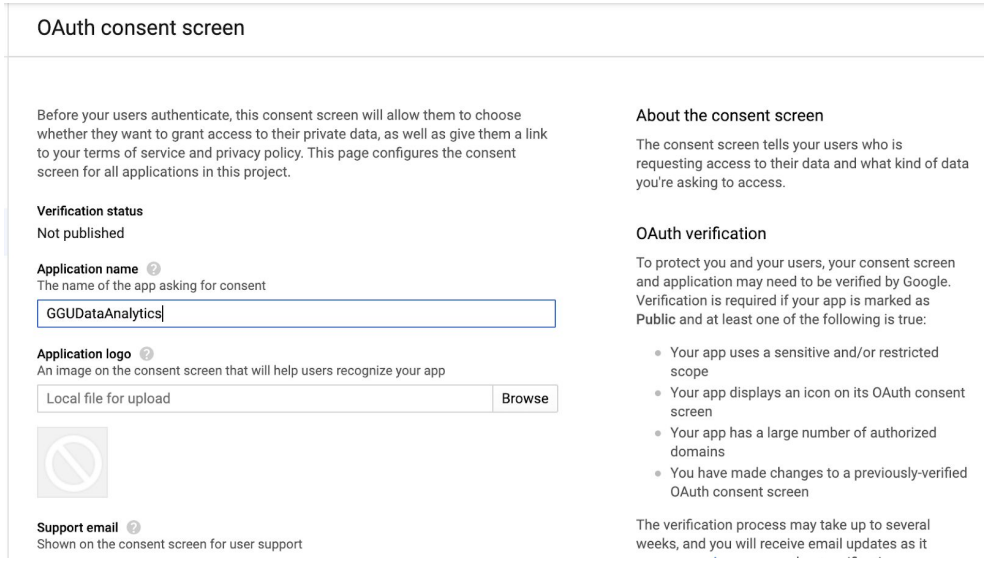
   You will also have to fill in some details for the OAuth consent screen, shown to people when they authorise your app. **Only the project name and email address fields are required.**



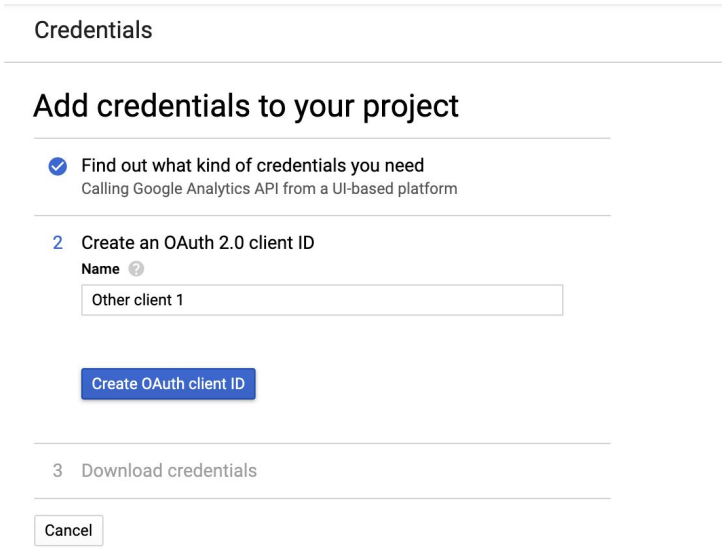
Click setup consent screen



Click create, then provide an Application name and email. Then Save.



7. Once the Oauth consent screen has been created, return to the Credentials wizard.
8. Create a client id - the default of “Other client 1” is okay.



Click Create OAuth ClientID

Eventually you will be shown your Client ID. You also need the Client secret. If you don’t see the secret, click “Done” and navigate to "Credentials.

9. Click your Client ID in the list and you should see the **Client ID** and **Client Secret**.

You should now see your client ID and client secret on the screen. If you also see things like “email address” and “javascript origins” then you have generated the wrong type of client ID and should start again from step 5.

Store the client id and client secret in some R variables.

```
> clientid <- "YOUR CLIENT ID"
> clientsecret <- "YOUR CLIENT SECRET"
```

```
55 ```{r}
56 options(googleAuthR.webapp.client_id = clientid)
57 options(googleAuthR.webapp.client_secret = clientsecret)
58 #options(googleAuthR.verbose=2)
59
60 # do from console for each new R session
61 #ga_auth()
62
63 ```
```

To view your credentials later:
https://console.developers.google.com/apis/credentials?project=steel-aileron-289619

Setting up R

Execute the following from the Console (not source editor) of RStudio

1. Install the library “Rcpp” - not having a current version of this library can causes problems later in the process. It’s a package that integrates R and C++ for performance

```
> install.packages("Rcpp")
```

2. Google Analytics in R library. We will use googleAnalyticsR

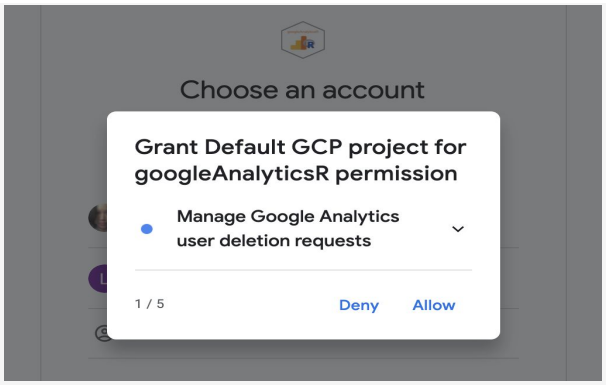
```
> install.packages("googleAnalyticsR")
```

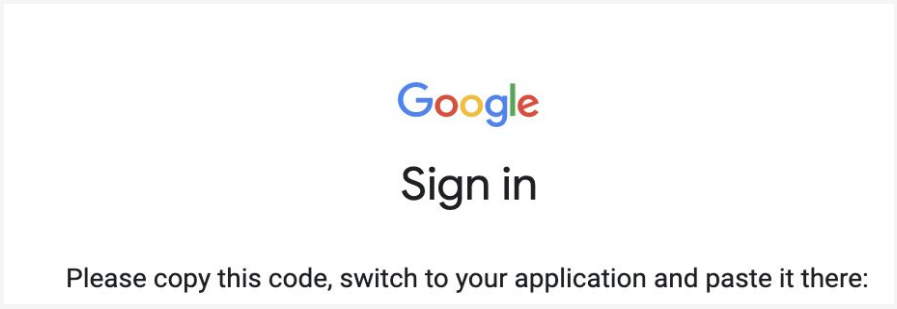
3. Authorise R to access Google Analytics data and create a token file with saved details, so you don't have to authorise every time .

    a. Load the library into R
    > library(googleAnalyticsR)

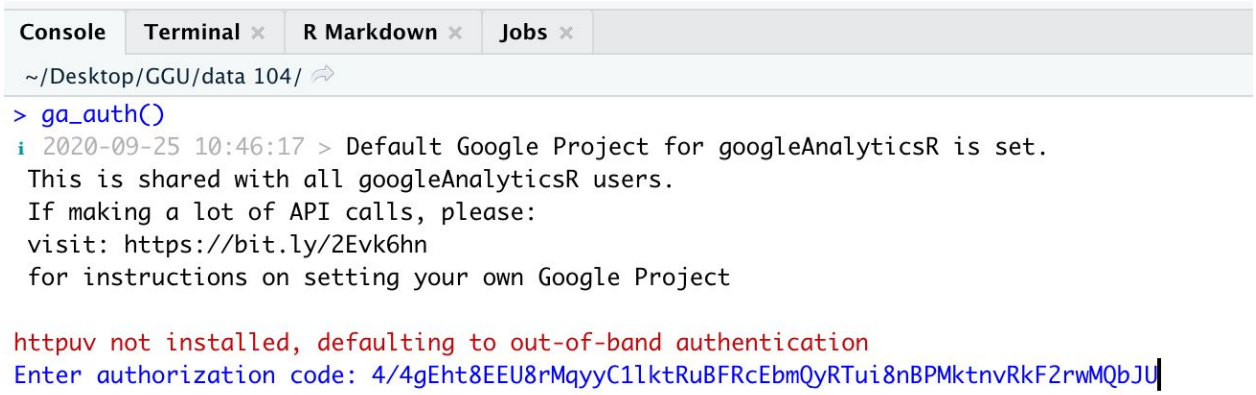
    b. Get your authorization token
    > ga_auth()

Click through all 5 options with Allow





You will be provided with an authorization code which should be entered below.



Making API Calls

API Reference:
https://cran.r-project.org/web/packages/googleAnalyticsR/googleAnalyticsR.pdf

```
## get your accounts

accounts = ga_account_list()
head(accounts)

# account list will have a column called "viewId"
accounts[c('webPropertyName', 'viewId')]
```

webPropertyName <chr>	viewId <chr>
bellasjardin.com	121593691
Data 104 GitHub Page	228283512
GWC	228264821
Google Merchandise Store	92320289
Google Merchandise Store	92324711
Google Merchandise Store	90822334

6 rows

 # Getting users, sessions and bounce rate. Use the viewId for your GitHub page

```
ga_data <- google_analytics(viewid2,
       # date_range = c("2015-07-30","2015-10-01"),
       date_range = c("30daysAgo", "yesterday"),
       metrics = c("users", "sessions", "bounces"),
       dimensions = c("date"),
       # max = 10  # number of rows
   )

# number of non-zero sessions
df = subset(ga_data, sessions > 0)
addmargins(with(df, table(df$date, df$sessions, dnn = c('date', 'sessions'))))
```

```
             sessions
date          1 2 6 Sum
  2020-09-06 0 0 1   1
  2020-09-08 1 0 0   1
  2020-09-19 0 1 0   1
  2020-09-23 1 0 0   1
  2020-09-24 1 0 0   1
  Sum        3 1 1   5
```