

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.


Google APIs					
Search for APIs and Services					
Manage resources					
+ CREATE PROJECT + CREATE FOLDER MOVE DELETE					
Filter					
<input type="checkbox"/>	Name	ID ↑	Last accessed ↓	Charges ?	Labels
<input type="checkbox"/>	▼ No organization		September 15, 2020		⋮
<input type="checkbox"/>	GoogleAnalytics	steel-aileron-289619	September 15, 2020		⋮
<input type="checkbox"/>	My First Project	extended-cache-179617	February 24, 2020		⋮
<input type="checkbox"/>	My Express Project	my-express-project-204218	February 2, 2020		⋮
<input type="checkbox"/>	Cloud Talent Project	cloud-talent-project	October 14, 2019		⋮

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

Google APIs

Select a project ▼

Search for APIs and Services



Google Analytics API

Google

The Analytics API provides access to Analytics configuration and report data.

ENABLE

TRY THIS API ↗

Click to enable this API

OVERVIEW

DOCUMENTATION





Select the project created earlier (GoogleAnalytics in my case)

Select a project

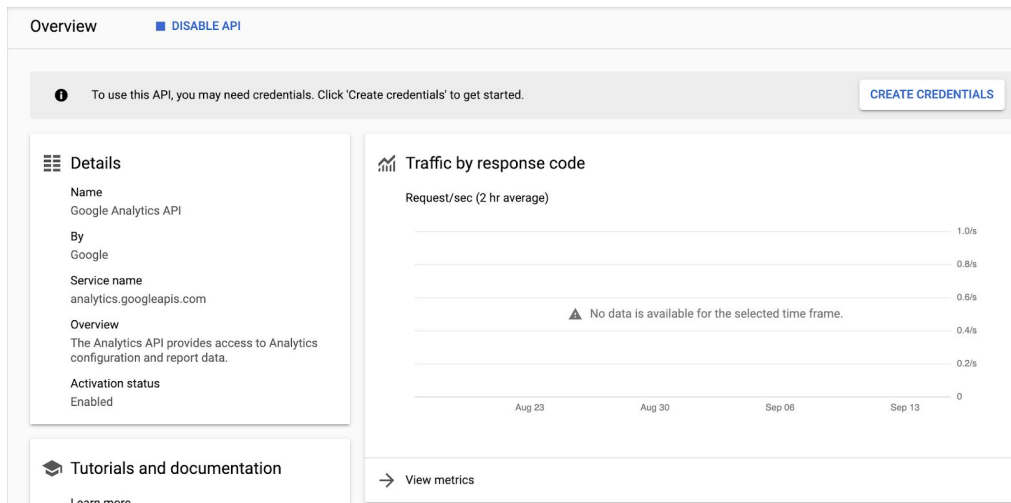
Search projects and folders

RECENT

ALL

Name	ID
 GoogleAnalytics ?	steel-aileron-289619
 My First Project ?	extended-cache-179617
 My Express Project ?	my-express-project-204218
 Cloud Talent Project ?	cloud-talent-project

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



Credentials

Add credentials to your project

- Find out what kind of credentials you need

We'll help you set up the correct credentials
If you wish you can skip this step and create an [API key](#), [client ID](#), or [service account](#)

Which API are you using?
Different APIs use different auth platforms and some credentials can be restricted to only call certain APIs.

Google Analytics API

Other API

- Select "Other UI (e.g. Windows, CLI tool)" in the "Where will you be calling the API from?"
 - 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.**

Set up OAuth consent screen

Before you can create an OAuth client ID, you must set up an OAuth consent screen to let users know who is requesting access to their data.

After you set up a consent screen, return to this page and refresh to create an OAuth client ID.


[NOT NOW](#) [SET UP CONSENT SCREEN](#)

Click setup consent screen


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.

☒ External 

Available to any user with a Google Account.

[CREATE](#)

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

OAuth consent screen

Before your users authenticate, this consent screen will allow them to choose whether they want to grant access to their private data, as well as give them a link to your terms of service and privacy policy. This page configures the consent screen for all applications in this project.

Verification status
Not published


Application name ?
The name of the app asking for consent

GGUDataAnalytics

Application logo ?
An image on the consent screen that will help users recognize your app

Local file for upload

Browse



Support email ?
Shown on the consent screen for user support

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 authorized domains
- You have made changes to a previously-verified OAuth consent screen

The verification process may take up to several weeks, and you will receive email updates as it

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

Credentials

Add credentials to your project

☒ Find out what kind of credentials you need
Calling Google Analytics API from a UI-based platform

2

Create an OAuth 2.0 client ID
Name ?

Other client 1

Create OAuth client ID

3

Download credentials

Cancel

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")
```

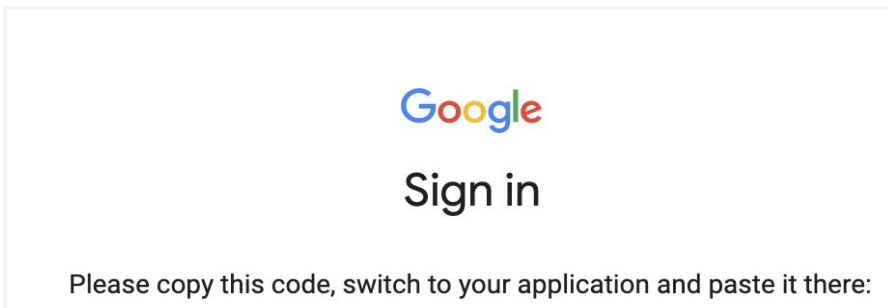
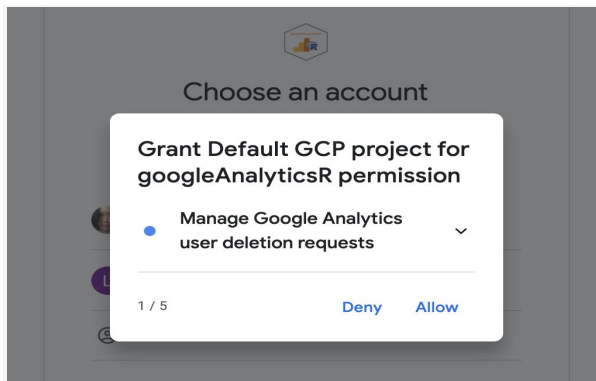
2. Google Analytics in R library. We will use [googleAnalyticsR](#)

```
> 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.

```
Console Terminal x R Markdown x Jobs x
~/Desktop/GGU/data 104/ ↗
> ga_auth()
i 2020-09-25 10:46:17 > Default Google Project for googleAnalyticsR is set.
This is shared with all googleAnalyticsR users.
If making a lot of API calls, please:
visit: https://bit.ly/2Evk6hn
for instructions on setting your own Google Project

httpuv not installed, defaulting to out-of-band authentication
Enter authorization code: 4/4gEht8EEU8rMqyyC1lktRuBFRcEbmQyRTui8nBPMktnvRkF2rwMQbJU|
```

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'))))
```

date	sessions			Sum
	1	2	6	
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