Rubén Grávalos

10-11-2023

TRIPADVISOR SCRAPING SERVICE



Contenido

[LOCATING THE PROJECT FILES 2](#_Toc150507102)

[1- Open the Colab notebook. 2](#_Toc150507103)

[2- Locate into the working directory. 3](#_Toc150507104)

[(ONLY ONCE) Enter your TripAdvisor API User Key. 4](#_Toc150507105)

[Run the Colab Script 5](#_Toc150507106)

[Locate the Colab execution environment into the project directory. 5](#_Toc150507107)

[Understand the input parameters. 5](#_Toc150507108)

[3. Location Search Script 7](#_Toc150507109)

[4. Location Details Script. 9](#_Toc150507110)

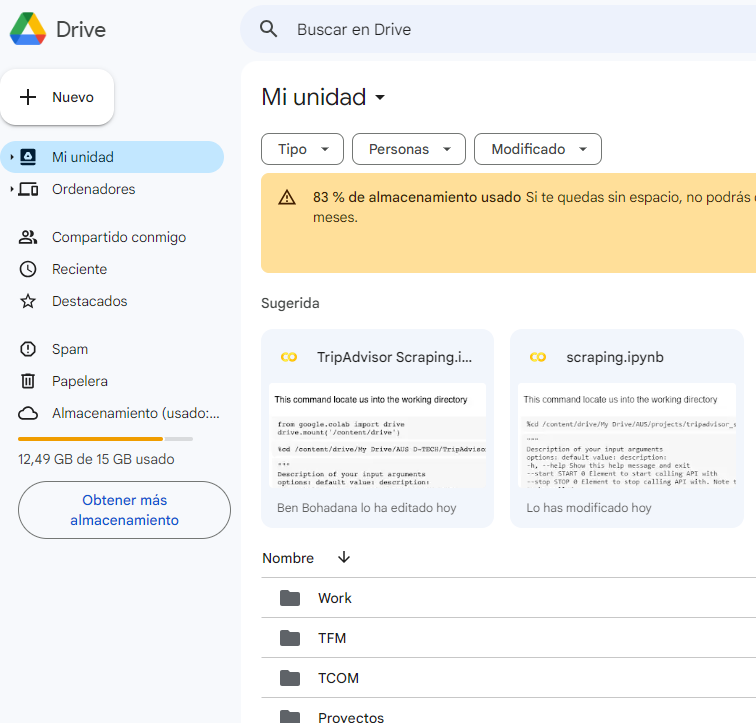
[5. Location Reviews Script. 10](#_Toc150507111)

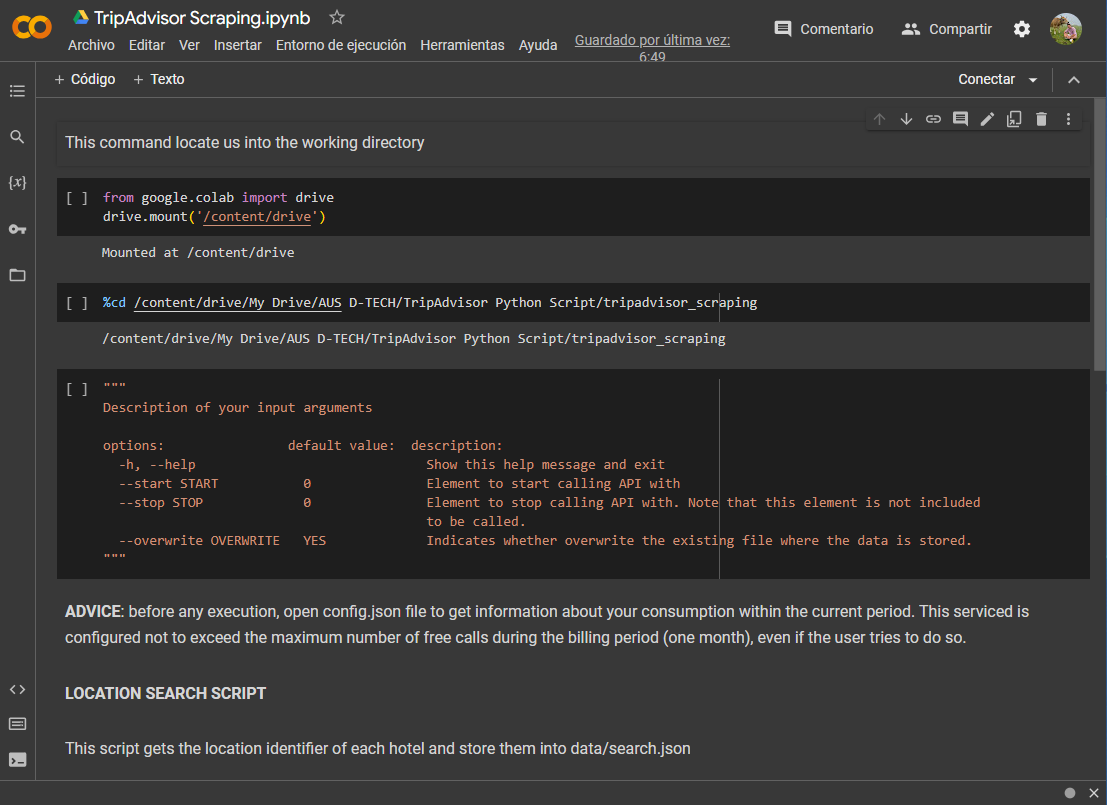
[Manage de .csv data 11](#_Toc150507112)

# LOCATING THE PROJECT FILES

## Open the Colab notebook.

* 1. Login to your Google Drive account in your browser and open your Colab notebook (i.e. “TripAdvisor Scraping.ipynb”.





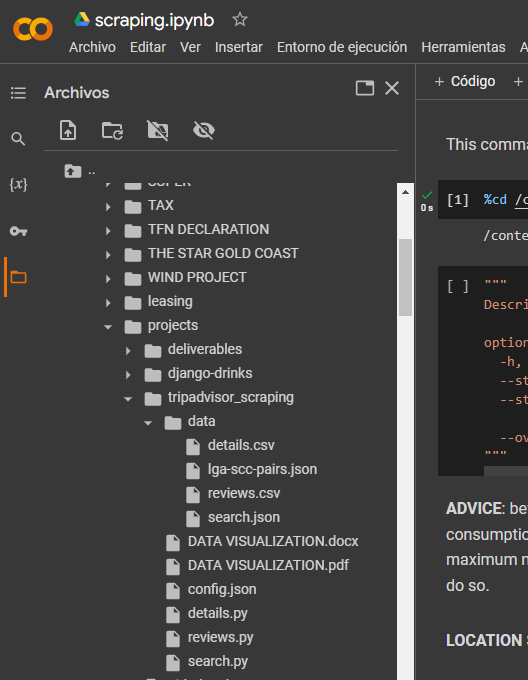
## Locate into the working directory.

* 1. Run the following section.

Interfaz de usuario gráfica, Texto

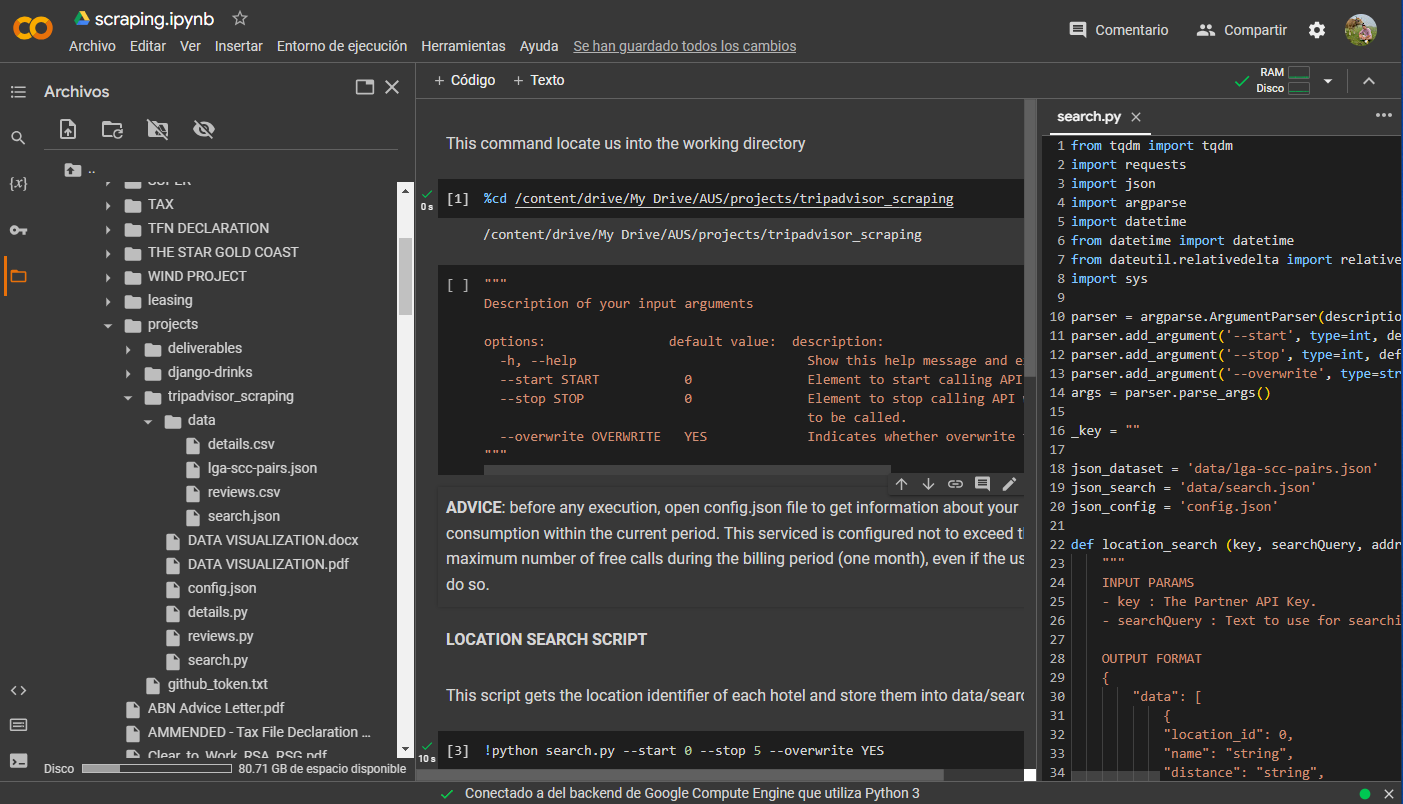
Descripción generada automáticamente con confianza media

* 1. Navigate to the working directory.

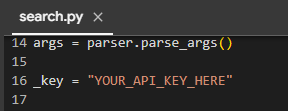


# (ONLY ONCE) Enter your TripAdvisor API User Key.

1. Double click on “search.py” file and a text editor will be opened at the right side of the screen.



1. Copy your TripAdvisor User API Key from <https://www.tripadvisor.com/developers?screen=credentials> and paste it in line 16 between the commas as in the next example:

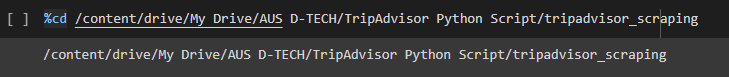


1. Ensure the changes are saved checking the name of the file, as in the above picture is “search.py”. This name means that the changes has been successfully saved. If the name of the file is shown as “\*search.py”, the symbol **\*** means that the changes hasn’t been saved yet. To save the changes, just double click on the name of the file “\*search.py”. It will turn into “search.py” meaning that the changes have been saved.

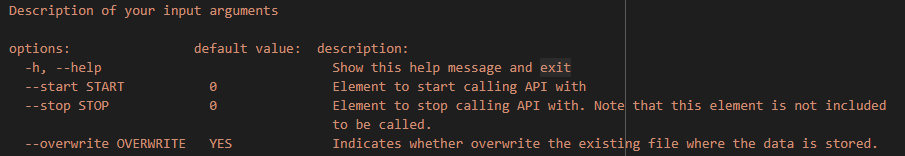
# Run the Colab Script

## Locate the Colab execution environment into the project directory.

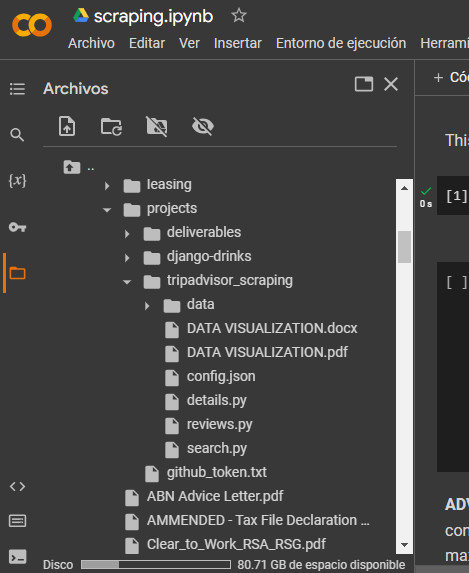
Run the following section.



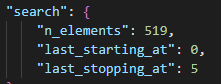
## Understand the input parameters.



1. Example for Location Search Script.
   1. Open the configuration file “config.json”.



* 1. Choose the number of hotels you want to get the information from.



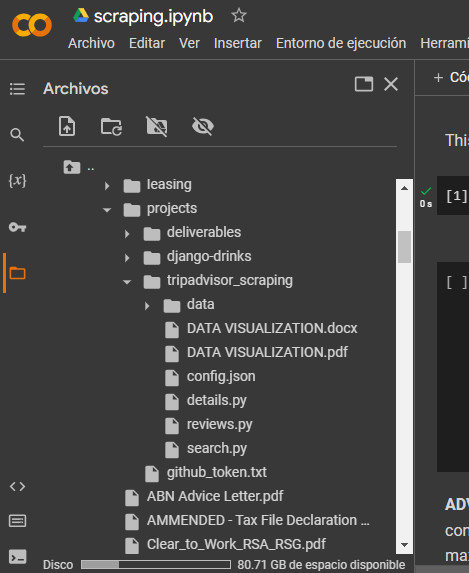
*n\_elements* : total number of city-suburb pairs in the dataset you are available to call. The dataset is located inside the “data” directory, in “lga-scc-pairs.json” file.

*last\_starting\_at* : The pair you started calling in the last execution.

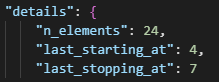
*last\_stopping\_at* : The pair you stopped calling in the last execution.

**NOTE**: The last pair of an input range when executing the script is never called. For example, when you make a call starting at 0 and stopping at 5, you call the pairs from 0 to 4 both included. It means that in the next call, you’ll need to start calling from 5. To the edge situation, when you mean to call the last pair (519), you’ll need to start calling, for example, at 515, and stop calling at 520.

1. Example for Location Details Script.
   1. Open the configuration file “config.json”.



* 1. Choose the number of hotels you want to get the information from.



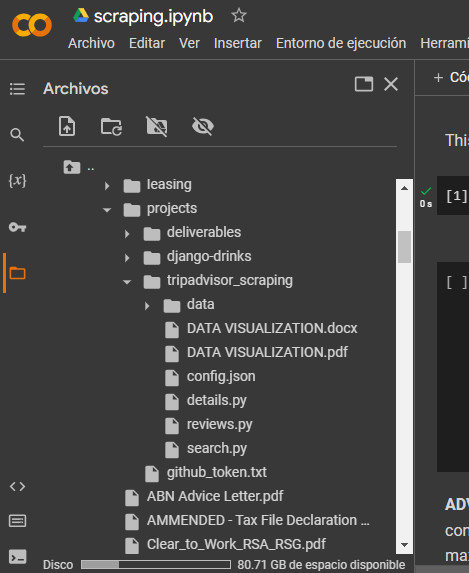
*n\_elements* : total number of location\_id saved into “search.json” you can use to make calls to get de contact details.

*last\_starting\_at* : The pair you started calling in the last execution.

*last\_stopping\_at* : The pair you stopped calling in the last execution.

**NOTE**: The last pair of an input range when executing the script is never called. For example, when you make a call starting at 0 and stopping at 5, you call the pairs from 0 to 4 both included. It means that in the next call, you’ll need to start calling from 5. To the edge situation, when you mean to call the last pair (24), you’ll need to start calling, for example, at 20, and stop calling at 25.

1. Example for Location Reviews Script.
   1. Open the configuration file “config.json”.



* 1. Choose the number of hotels you want to get the information from.

Texto

Descripción generada automáticamente

*n\_elements* : total number of location\_id saved into “details.csv” you can use to make calls to get de reviews of the locations.

*last\_starting\_at* : The pair you started calling in the last execution.

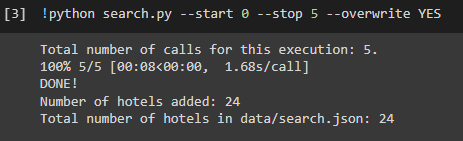
*last\_stopping\_at* : The pair you stopped calling in the last execution.

**NOTE**: The last pair of an input range when executing the script is never called. For example, when you make a call starting at 0 and stopping at 5, you call the pairs from 0 to 4 both included. It means that in the next call, you’ll need to start calling from 5. To the edge situation, when you mean to call the last pair (24), you’ll need to start calling, for example, at 20, and stop calling at 25.

## Location Search Script

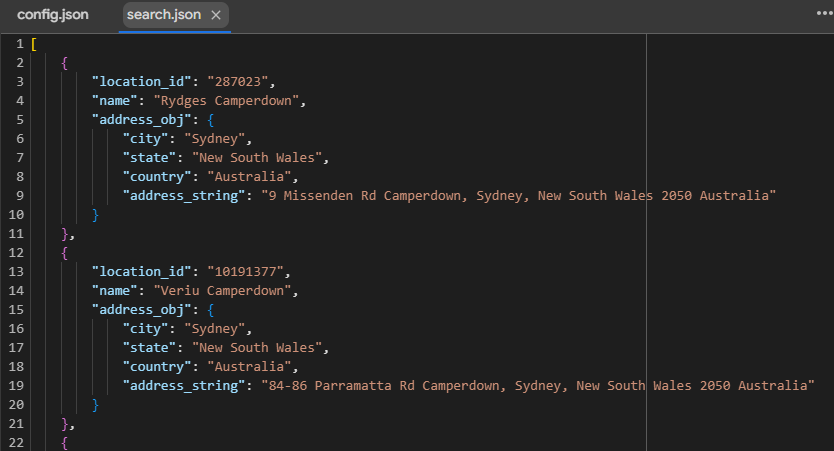
* 1. Run the section.

**NOTE**: The parameter –overwrite should be set to NO unless you want to delete the content of the file when running the command. You may want to set it to YES the first time you start calling from pair 0.



* 1. Watch the scraped information.

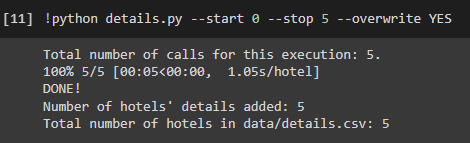
Go to the file explorer at the left of the screen and double click on “search.json” file inside the “data” directory. You will see the stored information of each hotel in .json format. You don’t have to use this file to get the contact details of the hotels, as they won’t be shown in here. Here is just stored the location\_id parameter you’ll use to make calls with that will get you the contact details of each hotel, which will be stored in a different file.



## Location Details Script.

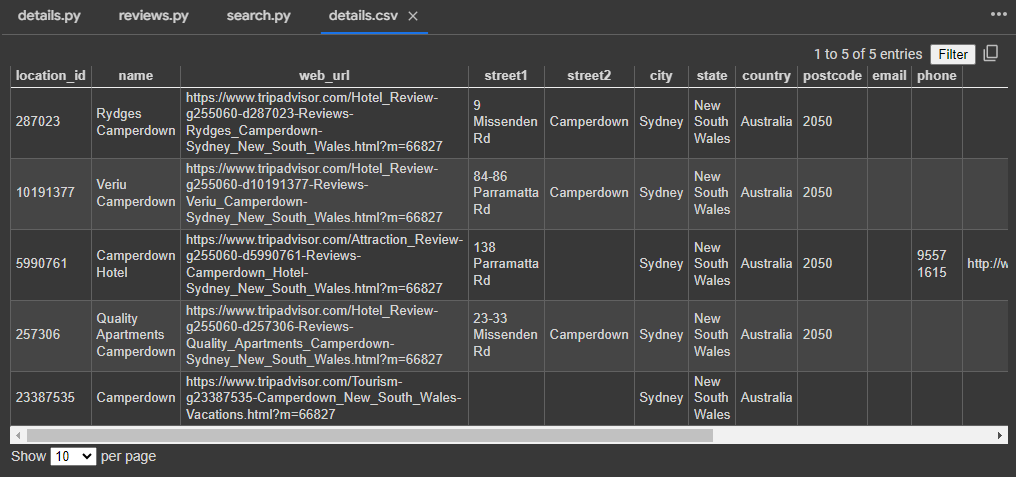
* 1. Run the section.

**NOTE**: The parameter –overwrite should be set to NO unless you want to delete the content of the file when running the command. You may want to set it to YES the first time you start calling from pair 0.



* 1. Watch the scraped information.

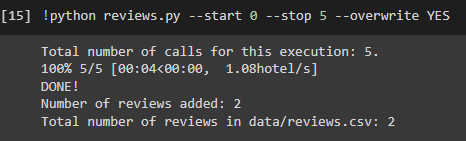
Go to the file explorer at the left of the screen and double click on “details.csv” file inside the “data” directory. You will see the stored information of each hotel in a table format. For those cells where no data is shown, it means that there’s not such information coming from TripAdvisor.



## Location Reviews Script.

* 1. Run the section.

**NOTE**: The parameter –overwrite should be set to NO unless you want to delete the content of the file when running the command. You may want to set it to YES the first time you start calling from pair 0. If you enter NO, the OLD reviews that are scraped again will turn into OLD value for the STATUS column.



* 1. Watch the scraped information.

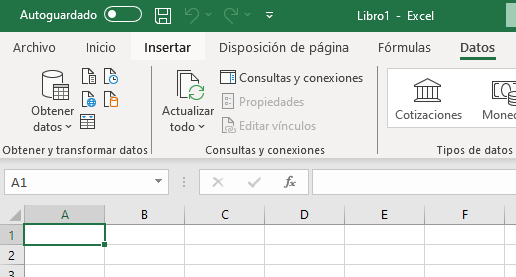
Go to the file explorer at the left of the screen and double click on “reviews.csv” file inside the “data” directory. You will see the stored information of each review matching the filter criteria of finding “bug” in the title or in the text of any review, and the contact details of the hotel in a table format. If no data is shown, it means that there isn’t any review matching the filter criteria for the hotels scraped from TripAdvisor.

Texto

Descripción generada automáticamente con confianza media

# Manage de .csv data

You can download the .csv files to your local computer, open Excel, click on Data, Get Data, From File, From Text/CSV, select the .csv file, and press the Load button. This way, you can save de data in a Excel file for your convenience to manipulate the details.



Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

