Read and Update Google Spreadsheets with Python!

EXCEL INTERMEDIATE PROGRAMMING PYTHON

Overview

- Learn how to setup a Google service account
- Read and Write data in Google Spreadsheets using Python

Introduction

Automation of work has been one of the quickest ways to reach functional efficiency. Moreover, in today's era where success is dependent on speed, automation of myriad repetitive tasks play a key role in any industry and at the most basic level of functionality. But many of us fail to understand how to automate some tasks and end in the loop of manually doing the same things again.

For instance, we often spend hours daily extracting data and then copy-pasting to spreadsheets and creating reports leading to excessive time consumption. Consequently, it would be great if we just run a script, and data is uploaded in the spreadsheet and the report is prepared with just a click. There are multiple advantages of report automation like you would be able to save time on data collection and removing typos and focus would be more on the analysis part.

In this article, we will see a step by step process to set up a Google service account. We will make use of the Google APIs to read google spreadsheets data using python and we will also update the data in the spreadsheet using python. We are going to read the cricket commentary data from the spreadsheet and

find out the number of runs scored by each batsman and then upload the results into a separate spreadsheet.

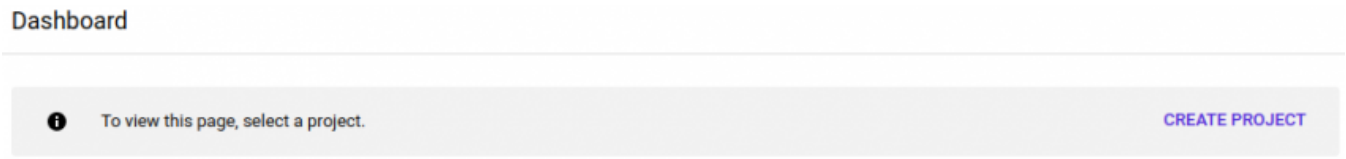
In case you are unfamiliar with Python, do have a look at our free course Introduction to Python

Table of Contents

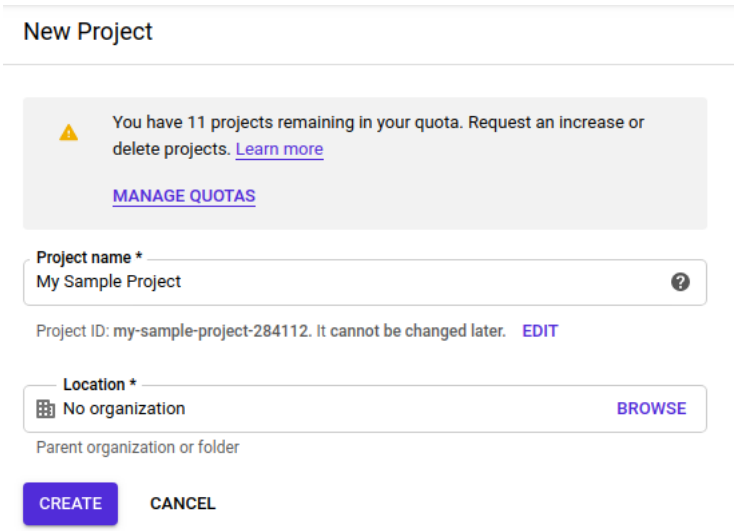
Create Google Service Account

In order to read and update the data from google spreadsheets in python, we will have to create a Service Account. It is a special type of account that is used to make authorized API calls to Google Cloud Services. First of all, make sure that you have a google account. If you have a Google account, you can follow these steps to create a Google service account.

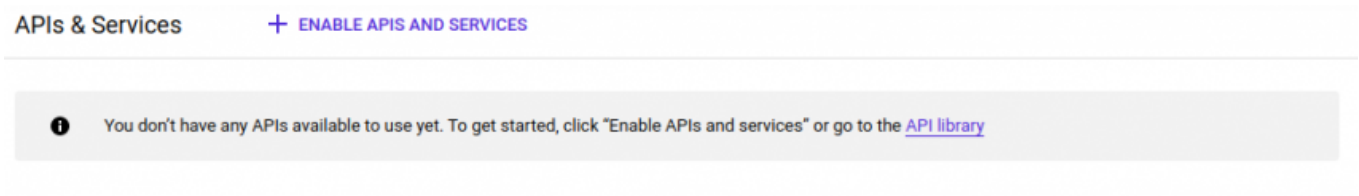
1. Go to the developer's console. Now, you will see something like this. Click on the Create Project button.



2. Then provide the project name and the organization name which is optional. Then click on the create button.



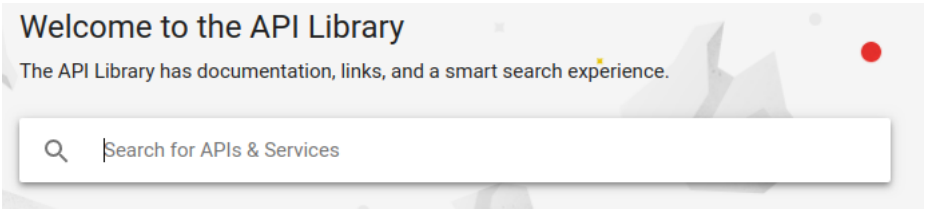
3. Now, that our project is created we need to enable the APIs that we require in this project. Click on the Enable APIs and Services button to search for the APIs that Google provides.



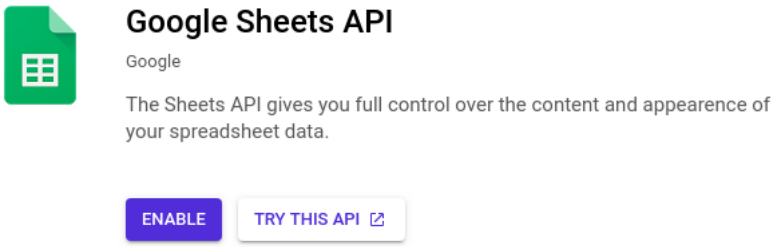
Consequently, we will add two APIs for our project.

- Google Sheets API
- Google Drive API

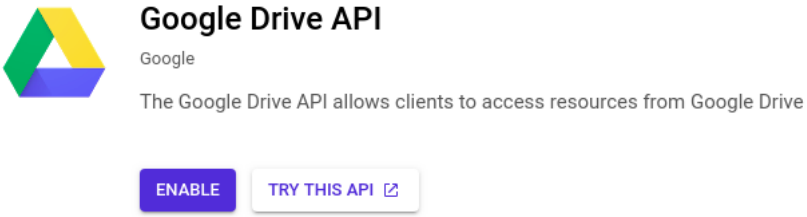
4. Then, in the search bar, search for these APIs and click on the enable button.



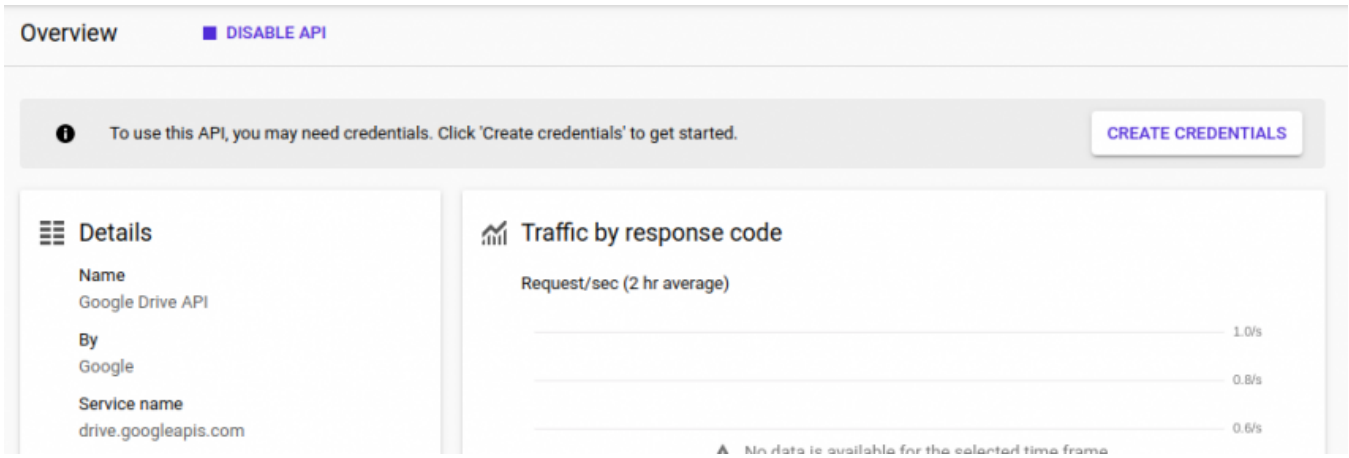
5. Google Sheets API will look something like this. It will allow you to access Google Spreadsheets. You would be able to read and modify the content present in the Spreadsheets.



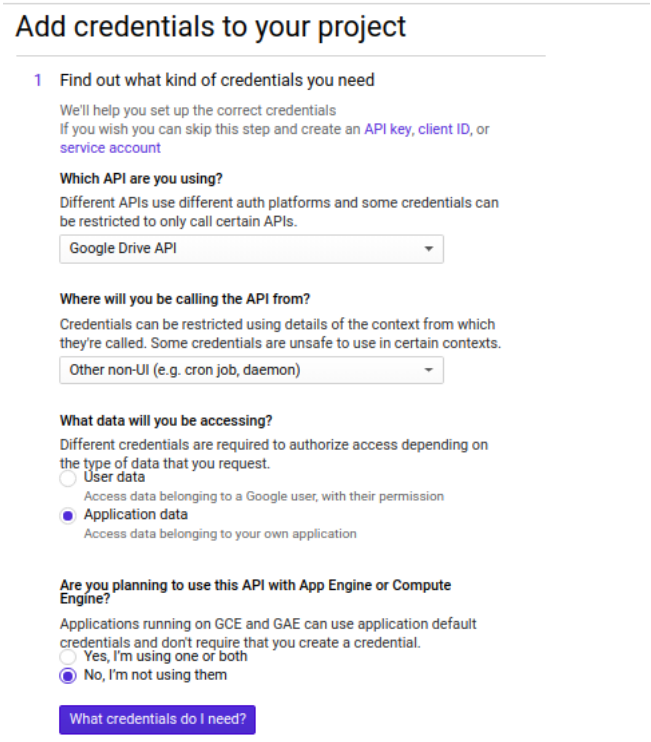
Google Drive API will look something like this. It will allow you to access the resources from Google Drive.



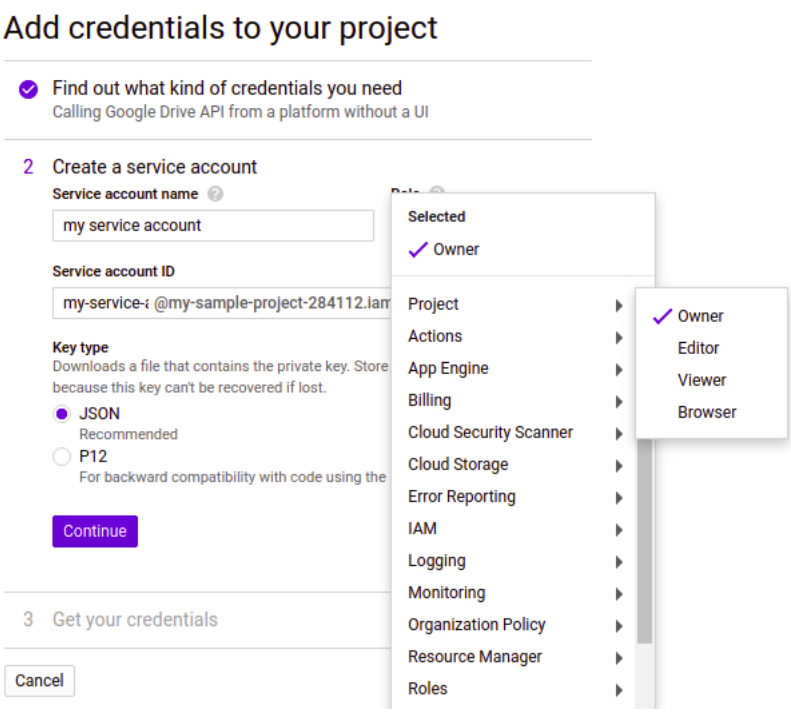
6. Once you have enabled the required APIs in your project then it's time to create credentials for the service account. Click on the Create Credentials button to continue.



7. Now, select Google Drive API in the type of **API required** question. We will be calling the API from a non UI based platform so select **Other non-UI (e.g. cron job, daemon)**. Select the **Application Data** in the next question as we do not require any user data to run our application. And also we are not using any cloud-based compute engine for our application. Finally, click on the **What credentials do I need?** button.



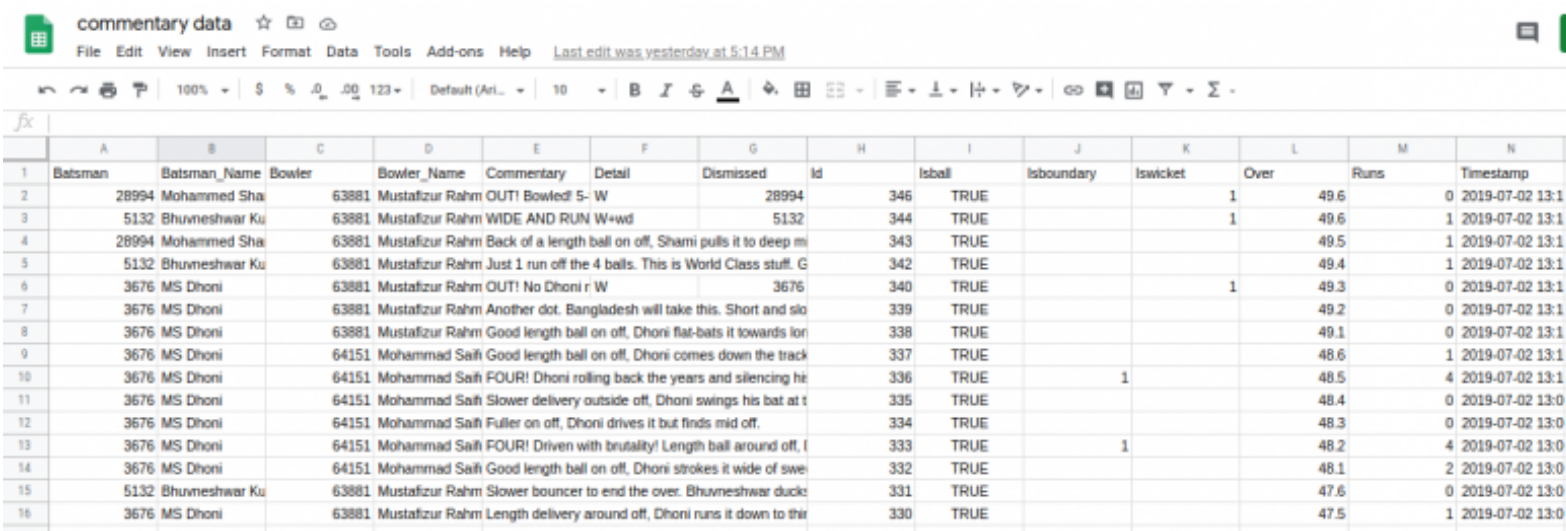
8. Then, share the google spreadsheets with other people and provide permission like edit or view only. Similarly, we will provide access to our service account. We will give it the complete access so that we will be able to read as well as write the spreadsheets and download the JSON file of the credentials.



Now, a JSON file will be downloaded which contains the keys to access the API. Our google service account is ready to use. In the next section, we will read and modify the data in the spreadsheet.

Read Data from Google Sheets

We will read the commentary data of the India Bangladesh cricket match. You can access the data here.



	Batsman	Batsman_Name	Bowler	Bowler_Name	Commentary	Detail	Dismissed	Id	Isball	Isboundary	Iswicket	Over	Runs	Timestamp
2	28994	Mohammed Sha	63881	Mustafizur Rahm	OUT! Bowled! 5-	W	28994	346	TRUE		1	49.6	0	2019-07-02 13:1
3	5132	Bhuvneshwar Ku	63881	Mustafizur Rahm	WIDE AND RUN	W+wd	5132	344	TRUE		1	49.6	1	2019-07-02 13:1
4	28994	Mohammed Sha	63881	Mustafizur Rahm	Back of a length ball on off, Shami pulls it to deep m			343	TRUE			49.5	1	2019-07-02 13:1
5	5132	Bhuvneshwar Ku	63881	Mustafizur Rahm	Just 1 run off the 4 balls. This is World Class stuff. G			342	TRUE			49.4	1	2019-07-02 13:1
6	3676	MS Dhoni	63881	Mustafizur Rahm	OUT! No Dhoni r	W	3676	340	TRUE		1	49.3	0	2019-07-02 13:1
7	3676	MS Dhoni	63881	Mustafizur Rahm	Another dot. Bangladesh will take this. Short and slo			339	TRUE			49.2	0	2019-07-02 13:1
8	3676	MS Dhoni	63881	Mustafizur Rahm	Good length ball on off, Dhoni flat-bats it towards lon			338	TRUE			49.1	0	2019-07-02 13:1
9	3676	MS Dhoni	64151	Mohammad Saifi	Good length ball on off, Dhoni comes down the track			337	TRUE			48.6	1	2019-07-02 13:1
10	3676	MS Dhoni	64151	Mohammad Saifi	FOUR! Dhoni rolling back the years and silencing his			336	TRUE	1		48.5	4	2019-07-02 13:1
11	3676	MS Dhoni	64151	Mohammad Saifi	Slower delivery outside off, Dhoni swings his bat at t			335	TRUE			48.4	0	2019-07-02 13:0
12	3676	MS Dhoni	64151	Mohammad Saifi	Fuller on off, Dhoni drives it but finds mid off.			334	TRUE			48.3	0	2019-07-02 13:0
13	3676	MS Dhoni	64151	Mohammad Saifi	FOUR! Driven with brutality! Length ball around off, I			333	TRUE	1		48.2	4	2019-07-02 13:0
14	3676	MS Dhoni	64151	Mohammad Saifi	Good length ball on off, Dhoni strokes it wide of swe			332	TRUE			48.1	2	2019-07-02 13:0
15	5132	Bhuvneshwar Ku	63881	Mustafizur Rahm	Slower bouncer to end the over. Bhuvneshwar ducks			331	TRUE			47.6	0	2019-07-02 13:0
16	3676	MS Dhoni	63881	Mustafizur Rahm	Length delivery around off, Dhoni runs it down to thir			330	TRUE			47.5	1	2019-07-02 13:0

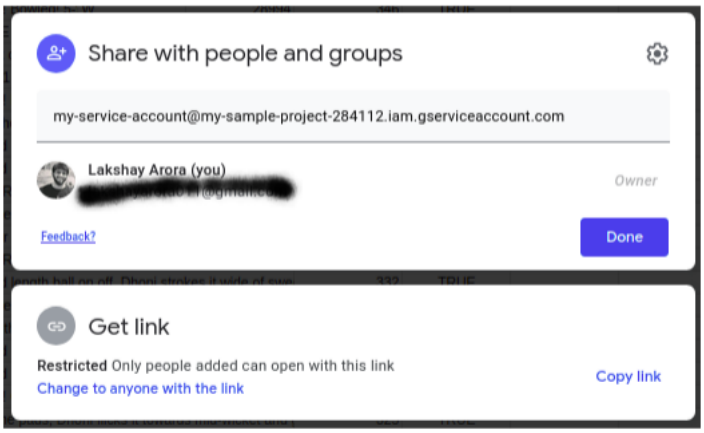
We have a ball by ball data of the complete match in the spreadsheet. Now, we will do a very basic task and calculate how many runs are scored by each of the batsmen. We can do this by using a simple groupby in pandas. And finally, we will upload the results in a separate sheet.

Provide access to the Google Sheet

Now, we need to provide access to the google sheet so that the API can access it. Open the JSON file that we downloaded from the developer's console. Look for the **client_email** in the JSON file and copy it.

```
"client_email": "my-service-account@my-sample-project-284112.iam.gserviceaccount.com",
```

Then click on the Share button on the Spreadsheet and provide access to this client email.



Now, we are ready to code and access the sheet using python. The following are the steps-

1. Importing the Libraries

We will use the **gspread** and **oauth2client** service to authorize and make API calls to Google Cloud Services.

You can install the libraries using the following commands.

```
!pip3 install gspread !pip3 install --upgrade google-api-python-client oauth2client
```

```
# importing the required libraries
import gspread
import pandas as pd
from oauth2client.service_account import ServiceAccountCredentials
```

view raw

libraries.py hosted with ❤ by **GitHub**

2. Define the scope of the application

Then, we will define the scope of the application and add the JSON file that has the credentials to access the API.

```
# define the scope
scope = ['https://spreadsheets.google.com/feeds','https://www.googleapis.com/auth/drive']

# add credentials to the account
creds = ServiceAccountCredentials.from_json_keyfile_name('add_json_file_here.json', scope)

# authorize the clientsheet
client = gspread.authorize(creds)
```

view raw

new_scope.py hosted with ❤ by GitHub

3. Create the Sheet Instance

Use the client object and open the sheet. You just need to pass the title of the sheet as the argument. Also, you can pass the URL of the sheet if you want to do so.

Access Particular Sheet: We have multiple sheets in a single spreadsheet. You can access particular google spreadsheets with python by providing the index of that sheet in the **get_worksheet** function. For the first sheet, pass the index 0 and so on.

```
# get the instance of the Spreadsheet
sheet = client.open('commentary data')

# get the first sheet of the Spreadsheet
sheet_instance = sheet.get_worksheet(0)
```

view raw

client.py hosted with ❤ by GitHub

Basic functionalities

The API provides some basic functionalities such as the number of columns by using col_count and get the value in a particular cell. Here are some examples of the same.

```
# get the total number of columns
sheet_instance.col_count
## >> 26


# get the value at the specific cell
sheet_instance.cell(col=3,row=2)
## >> <Cell R2C3 '63881'>
```

view raw

cell.py hosted with ❤ by GitHub

4. Get all records

Then, we will get all the data present in the sheet using the **get_all_records** function. It will return a JSON string containing the data.

```
# get all the records of the data
records_data = sheet_instance.get_all_records()

# view the data
records_data
```

view raw

get_records.py hosted with ❤ by GitHub

```
[{'Batsman': 28994,
  'Batsman_Name': 'Mohammed Shami',
  'Bowler': 63881,
  'Bowler_Name': 'Mustafizur Rahman',
  'Commentary': 'OUT! Bowled! 5-fer to finish a tremendous last over. His 4th 5-wicket haul in ODIs. Around off,
Shami moves across to paddle but misses the ball hits his pads and goes onto hit the stumps. 2 wickets and just 3
runs from the final over. Top notch from Mustafizur. INDIA FINISH WITH 314/9 FROM THEIR 50 OVERS.',
  'Detail': 'W',
  'Dismissed': 28994,
  'Id': 346,
  'Isball': 'TRUE',
  'Isboundary': '',
  'Iswicket': 1,
  'Over': 49.6,
  'Runs': 0,
  'Timestamp': '2019-07-02 13:18:47'},
 {'Batsman': 5132,
  'Batsman_Name': 'Bhuvneshwar Kumar',
  'Bowler': 63881,
```


5. Convert the Dictionary to the Dataframe

In data science, **pandas** is one of the most preferred libraries to do data manipulation tasks. So we will first convert the JSON string to the pandas dataframe.

In case you are not comfortable with the pandas, I would highly recommend you to enroll in this free course: [Pandas for Data Analysis in Python]

```
# convert the json to dataframe
records_df = pd.DataFrame.from_dict(records_data)

# view the top records
records_df.head()
```

view raw

df.py hosted with ❤ by **GitHub**

	Batsman	Batsman_Name	Bowler	Bowler_Name	Commentary	Detail	Dismissed	Id	Isball	Isboundary	Iswicket	Over	Runs	Timestamp
0	28994	Mohammed Shami	63881	Mustafizur Rahman	OUT! Bowled! 5-fer to finish a tremendous last...	W	28994	346	TRUE		1	49.6	0	2019-07-02 13:18:47
1	5132	Bhuvneshwar Kumar	63881	Mustafizur Rahman	WIDE AND RUN OUT! Slower delivery outside off,...	W+wd	5132	344	TRUE		1	49.6	1	2019-07-02 13:17:28
2	28994	Mohammed Shami	63881	Mustafizur Rahman	Back of a length ball on off, Shami pulls it t...			343	TRUE			49.5	1	2019-07-02 13:16:03
3	5132	Bhuvneshwar Kumar	63881	Mustafizur Rahman	Just 1 run off the 4 balls. This is World Clas...			342	TRUE			49.4	1	2019-07-02 13:15:17
4	3676	MS Dhoni	63881	Mustafizur Rahman	OUT! No Dhoni magic in the last over. Slower b...	W	3676	340	TRUE		1	49.3	0	2019-07-02 13:13:39

6. Grouping Batsman

Then, we will create a groupby of the number of runs scored by a batsman and upload that dataframe in the separate sheet.

```
# number of runs by each batsman
runs = records_df.groupby(['Batsman_Name'])['Runs'].count().reset_index()
runs
```

view raw

runs.py hosted with ❤ by **GitHub**

	Batsman_Name	Runs
0	Bhuvneshwar Kumar	4
1	Dinesh Karthik	9
2	Hardik Pandya	2
3	KL Rahul	93
4	Liton Das	24
5	MS Dhoni	33
6	Mashrafe Mortaza	5
7	Mohammad Saifuddin	42
8	Mohammed Shami	2
9	Mosaddek Hossain	7
10	Mushfiqur Rahim	23
11	Mustafizur Rahman	1
12	Rishabh Pant	43
13	Rohit Sharma	94
14	Rubel Hossain	11
15	Sabbir Rahman	40
16	Shakib Al Hasan	75
17	Soumya Sarkar	39
18	Tamim Iqbal	31
19	Virat Kohli	27

Now, we will add this dataframe into the google sheets.

Update Data in Google Sheets

The following are steps to update data in google sheets.

1. Create a Separate Sheet

Firstly, we will create a separate sheet to store the results. For that, use the **add_worksheet** function and pass the number of rows and columns required and the title of the sheet. After that get the instance of the second sheet by providing the index which is 1.

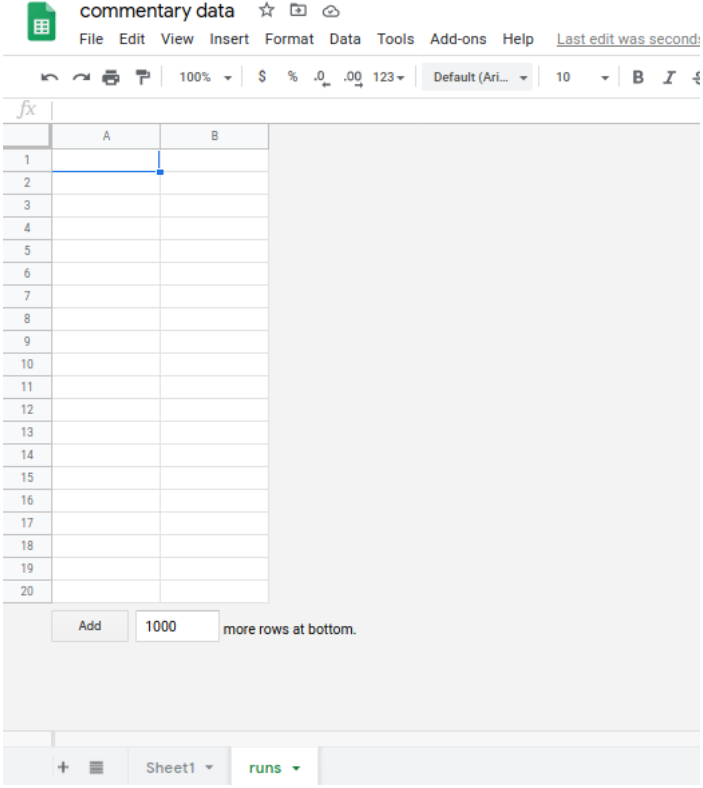
Once you run this command, you will see that a separate sheet is created.

```
# add a sheet with 20 rows and 2 columns
sheet.add_worksheet(rows=20,cols=2,title='runs')

# get the instance of the second sheet
sheet_runs = sheet.get_worksheet(1)
```

view raw

add_sheet.py hosted with ❤ by **GitHub**



2. **Update values to the sheet**

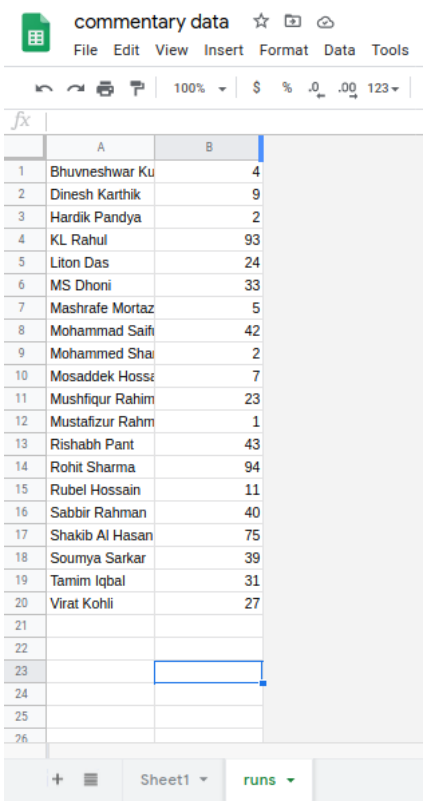
Then, convert the runs dataframe into the 2-D list and use the function to add values in the sheet. With this single line of code, you can update the sheet. Then, you will get a message of the number of rows and columns updated with some more details.

```
sheet_runs.insert_rows(runs.values.tolist())
```

view raw

insert_into_sheet.py hosted with ❤ by **GitHub**

```
{'spreadsheetId': '1ww2UcaIz19H_1MRJLmCLp6UGQMvg1_xg65yrn0uZ0rQ',
 'updates': {'spreadsheetId': '1ww2UcaIz19H_1MRJLmCLp6UGQMvg1_xg65yrn0uZ0rQ',
  'updatedCells': 40,
  'updatedColumns': 2,
  'updatedRange': 'runs!A1:B20',
  'updatedRows': 20}}
```



End Notes

To summarize, in this article, we dived into understanding various steps involved in the process of creating a service account. And how to read the write in the google spreadsheets right from your python console. We downloaded the spreadsheet data and converted it into the pandas dataframe and created a groupby table and uploaded that on the spreadsheet again. This API can be very helpful in the automation of reports.

In case you want to brush up your spreadsheet concepts, I recommend the following article and course-

- *Microsoft Excel: Formulas & Functions*
- *10+ Simple Yet Powerful Excel Tricks for Data Analysis*

I hope this helps you in automating scripts and saving loads of your valuable time. Reach out in the comment section in case of any doubts. I will be happy to help.

Article Url - https://www.analyticsvidhya.com/blog/2020/07/read-and-update-google-spreadsheets-with-python/

LAKSHAY ARORA

Ideas have always excited me. The fact that we could dream of something and bring it to reality fascinates me. Computer Science provides me a window to do exactly that. I love programming and use it to solve problems and a beginner in the field of Data Science.