**Reflection**

* Reflect on your Portfolio Assignment – are there areas you would like to still improve it in?

Sure. During this portfolio assignment, I learned how to use input function to input data, pull data from text files and JSON file, import data to Sqlite database. I still want to learn how to pull data from tables from other databases such as Postgresql, Sql or oracle and so on, pulling data from other different kinds of data sources. For data visualization, I would like to improve on using some other libraries, such as seaborn and ggplot. I would like to explore use pandas to perform more other data analysis.

* Can you think of other functionality that would be beneficial?

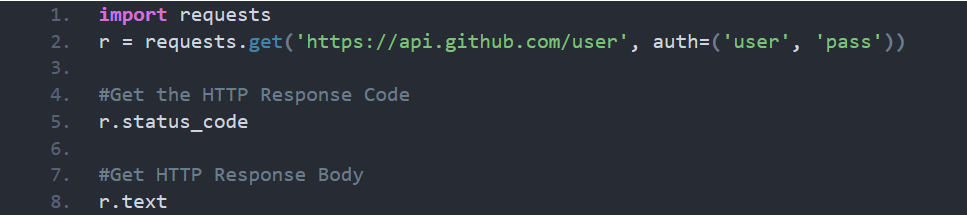
I think Requests is very beneficial. It’s a very popular HTTP client library in Python, being used to make API requests and supports all HTTP methods and advanced HTTP features such as authentication, session cookies, SSL handling. Faster Than Requests, as its name, mimics the Requests library, and works faster than Requests. Its advantage is lightweight of lines of code. It contains bunch of utility functions beyond usual HTTP methods. This library provides some convenient utility functions for tasks such as scraping URLs, downloading resources such as image files, or clubbing requests to multiple URLs and returning a single response.

* What was your experience implementing the new functionality?

I did not have personal experience of implementing this functionality so far, but on my work, I saw some colleague built program using this Request library to pull data from website for staff timesheet data to build report on it. Sounds like it’s a very good library to be used.

* Was the functionality hard to implement?

I don’t think so. First install this library like what we did with other library in the course. Import this library and use requests.get() method to connect to HTTP page to get response code and response body. Such as below, invoking an HTTP GET API using requests:



* Was the documentation easy or difficult to find?

No. Googled for the documentation, it is available at hand. Below links are some options for use to review documentation for Requests library:

<https://readthedocs.org/projects/requests/downloads/pdf/master/>

<https://realpython.com/python-requests/>

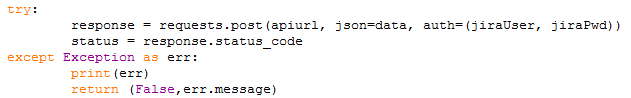
<https://requests.readthedocs.io/en/master/>

* Was the documentation difficult to interpret?

No. It describes the usage very clear, such as methods, parameters and API documentation guide. Documentation covers all the interfaces of Requests. All of Requests’ functionality can be accessed by 7 methods and they all return an instance of the [Response](https://requests.readthedocs.io/en/master/api/#requests.Response) object. They are: requests.request(), requests.head(), requests.get(), requests.post(), requests.put(), requests.patch(), requests.delete. It also covers exceptions, sessions and other classes.

* Show screenshots of your added functionality at work.

I did not use this functionality in the assignment, but I do use at work. These are two screenshots of how I import it and use its method:



Referenced link:

<https://rapidapi.com/blog/best-python-api-frameworks/>