

1. Assignment Description

This assignment will require that you write code to interface with an external REST-based APIs. We could have used almost any external APIs, but for this assignment we chose GitHub because many of its APIs are public and do not require any authorization or API Keys. This simplifies both the use and setup. This assignment entails making an API request to GitHub to view a user's repositories and the amount of commits each has.

2. Author- Ryan Davis

3. Summary

Results:

- Implemented a function that interacts with the GitHub REST API to retrieve a user's repositories and the number of commits per repository.
- Developed unit tests for the GitHub API interactions.
- Integrated CircleCI for continuous integration and added a build badge in the README.

Reflection:

- What I learned:
 - How to interact with APIs in Python using requests.
 - The importance of writing unit tests before implementation.
 - How to set up CircleCI for continuous integration.
 - What worked well:
 - Unit tests confirmed that the function handles various cases.
 - Using CircleCI helped automate testing and ensured the code was always working.
 - Challenges faced:
 - Encountered GitHub API rate limits, requiring token authentication.
 - Travis-CI not displaying GitHub as an option, leading to using CircleCI instead.
 - Had issues with package installations (requests, dotenv) on CircleCI.
-

4. Detailed Results

GitHub Repository- <https://github.com/ryry91021/ssw567/tree/main/hw/hw4/GitHubApi567-hw4a>

GitHub API Calls

- Retrieve repositories:
 - `https://api.github.com/users/{username}/repos`
- Retrieve commits for a repository:
 - `https://api.github.com/repos/{username}/{repo}/commits`

Implemented Code

- The program retrieves repositories and counts commits using requests and JSON parsing.

Example Output:

Repo: ProjectA, Number of commits: 14

Repo: ProjectB, Number of commits: 32

5. Honor Pledge

"I pledge my honor that I have abided by the Stevens Honor System."

Signed: *Ryan Davis*
