

# CO324 Lab 3: Exploring the Github API with Requests

## Objective

Use command line tools to make HTTP requests to various APIs. At the end of the lab you should be able to

- Use APIs via the Requests 3 Python module
- Set up authentication and sessions.
- Understand the difference between various HTTP request methods (verbs.)
- Be able to interpret HTTP responses and status codes.
- Awareness of common HTTP headers.

## Preparation

Install Python version 3.7 or newer on your computer. These instructions assume a Linux environment. If you prefer to use Windows, follow these [instructions to install WSL2](#). Run the following commands in a shell in this lab's working directory. Change the python version number to suit.

1. [Refer to this for more information about pipenv.](#)

```
pip3 install pipenv
pipenv install requests --python 3.8
pipenv shell
```

2. Create a Github account if you have not already done so. Log in and create a Personal API token (**PAT**) that you will use to authenticate yourself:  
<https://github.com/settings/tokens>. Copy the generated token to a text file.
3. [Install a REST client.](#)

# Instructions

- Use the **Requests 3 module only** with standard built-in modules in Python to complete these exercises.
- Put the solutions to coding exercises in files named ex1a.py, ex2b.py, etc.
- Solutions to short answer questions should be in a single plain text file named answers.txt. State your E-number at the top of answers.txt.
- Zip all your files and submit them (please don't use other formats.)

**Submissions that do not follow instructions will incur a 10% penalty.**

## References

- <https://www.pythonforbeginners.com/requests/using-requests-in-python/>
- <https://realpython.com/python-requests/>
- <https://3.python-requests.org/>
- <https://docs.github.com/en/rest/guides>

## Exercises

1.
  - a. What do you see in the response when calling the **API root endpoint** <https://api.github.com>.
  - b. Get your Github profile information from the endpoint (replace username with your's) <https://api.github.com/users/username>
  - c. What is the purpose of the various X-Ratelimit- headers that you see in the response?
  - d. To access restricted operations via the API, you need to authenticate yourself first. Explain what effect the following code has at the HTTP protocol level, when a request is made using that session.

```
with requests.Session() as session:  
    session.headers['Authorization'] = 'token YOUR_PAT_STRING'
```

- e. Use the session you created to create a repository owned by you by making a POST request to <https://api.github.com/user/repos>. Include the following request body, to create a repo named "test." (Hint: use the json keyword argument when calling the method requests.post)

```
{'name':'test', 'description':'some test repo'}
```

2.

- a. Write a function called github\_superstar that does the following

```
from typing import List, Tuple
def github_superstars(organization: str) -> List[Tuple]:
    """Sirasa superstar for Computer engineers. :)
    1. Get a list of members in the Github organization
    2. For each member, find the repo they own with the most stars.
    3. Add the repo name and the number of stars it has to a list.
    4. Return the list sorted in descending order of stars.
    """
```

- b. Write code to watch the repo that wins github\_superstars for <https://github.com/cepdnackl>.
- c. Github has two versions of its API currently available: v3 and v4. How do you inform the API which version you want to use?
- d. Explain the difference between the two API versions. Is one "better" than the other? Why?