**TASK 2**

* **We assume that the host is 0.0.0.0:8000, followed by the various urls of the requests:**
  + **0.0.0.0:8000/api/v1/authentication/login for login requests**
  + **0.0.0.0:8000//api/v1/users/5ca5ddd23bf77546543e2c9f for update user requests**
* **We are using Windows 10 Pro 64bit**

Here, I have not written an excel file with test-cases, since the code logic is quite easier to follow. I conducted task2, using:

* **Python version: 3.9.4**
* **Python Pytest version: 7.1.2**
* **Python Requests version: 2.25.1**
* **Pycharm 2021.1**

**After installing python and setting it up in path, and then installing pycharm, installing the various libraries (pytest,requests) can be done through the pycharm interface.**

**To run my script, I opened cmd, use cd to navigate to the directory where my code exists, then use:**

**py.test -v -rP.**

**Program Flow:**

The program flow is as follows:

* I do the various imports (pytest, requests)
* Create 6 variables (3 for the requests given in the task, and another 3 for the respective responses), where I save the exact json formula and values, as described in the task.
* Then, I create one function that tests each case, each followed by its own helper function:
  + successful login (test\_successful\_login and check\_successful\_login\_response)
  + unsuccessful login (test\_unsuccessful\_login and check\_unsuccessful\_login\_response)
  + successful update user (test\_successful\_updateuser and check\_test\_successful\_updateuser\_response)
* In the test function, I make a request to the appropriate url, by sending the json request body (as well as header with token for update user case), and save the response.
* I save the json response in response\_body.
* I send response\_body in the helper function. The helper function checks (assert) whether the various keys of the json exist (basically if the json is in the exact form that is shown to us in the 3 examples), while also returning the status code, as well as some value from the json that we want:
  + The Login Token for successful login
  + The error message for the unsuccessful login
  + The role for the successful update user
* Lastly, in the function we check whether the status code was the one we wanted (assert), and if the other value we returned was in the form that we wanted (the token is a string, the error message is some certain message, or the role is either “admin”, “executor”, or “reporter”).

GitHub:

<https://github.com/Evanslearn/Automated-Test-for-Rest-Endpoints>