**API Automation for WorkMarket**

**Notes**

1. The code was developed using Python 3.6/3.7 on a Windows machine, using Eclipse IDE
2. The code requires that the ***requests*** module be installed for the Python version in use.
3. The code can be run either using the command line or using an IDE, such as Eclipse
4. The code can be run as a standalone, where it executes a set of built-in tests. These tests include both functional and negative tests. The functional tests are executed first, followed by the negative tests.
5. The built-in test list is commented to list which tests pass and which fail. The negative tests also include those where a field name is missing.
6. The functional tests include tests that test any bounds associated with any of the field names
7. The code can also be run using a ‘-tc’ or ‘—test’ argument. The argument takes a user dictionary specified as a string. A sample format for using the argument would be :

* ***On Windows:***

*python.exe apitester.py -tc “firstName:sam,lastName:guy,username:samg,password:Pass1234!,email:samg@gmail.com,phoneNumber:111-222-0000”*

* **On Linux:**

*python apitester.py -tc “firstName:sam,lastName:guy,username:samg,password:Pass1234!,email:samg@gmail.com,phoneNumber:111-222-0000*”

1. I had originally included a function to validate the argument passed into the test and ensure that the argument contained all the requisite field names. However, after much deliberation, I removed it as the API would report on any errors in the field names passed in. Further, the function wasn’t consistent with the rest of the code, where I pass the built-in test dictionaries as-is without validation.
2. During testing, I noticed that the firstName field can be entirely skipped and the API does not report errors. But, if it is specified, it is validated by theAPI
3. Also, the password field does not mandate the criteria for at least one uppercase letter. The API accepts passwords of valid length containing only lowercase letters, numbers and special characters.
4. The API also validates user dictionaries with additional fields, as long as the mandated fields are present and valid.