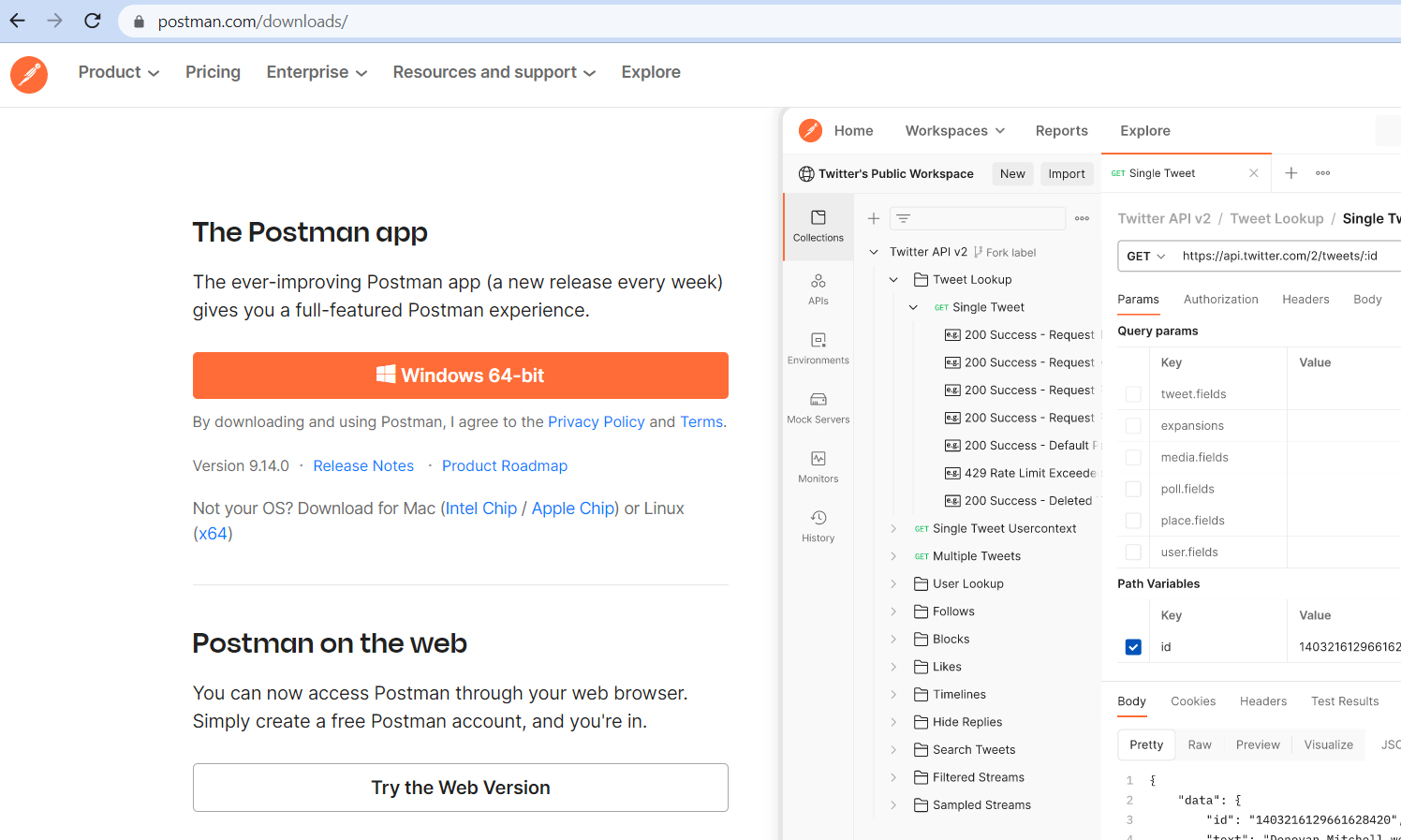
**Documentation for API Testing**

**Manual API Testing via Postman Tool:**

* Follow the steps in the Test document to get the Swagger UI tool documentation on your local server.
* Open Swagger UI documentation and try each every endpoint by following the instructions I the document. Prepare the test cases based on the documentation. Try with different positive and negative scenarios. Note down the results obtained.
* Install Postman on your system from official website



* Open Postman tool and try manually each endpoint by giving different inputs and parameters with various HTTP methods.
* Note down the results obtained.

**API Automation using Requests tool with Python:**

* Python and PIP installed on system during accessing Swagger documentation.
* Check for Python and Pip versions on your system command prompt with following commands:

**Python –version**

**Pip –version**

**Pip list**

* Install Requests tool using following command in command prompt

**Pip install requests**

* Open Pycharm IDE and create a new project with desired name and select the new environment option with Python Interpreter and click create
* Select the project and go to file and click on settings🡪Project name🡪Interpreter🡪click on the “+” icon and search for Requests package and JsonPath respectively and click on install package🡪close window and click on Ok.
* Go to Project, right click on Project🡪New🡪package🡪Api testing
* Right click on Api testing package🡪New🡪Python file🡪Give the test name
* Import Requests Package to the test and continue with the Requests and Python code.
* Included all the possible assertions and try except blocks for smooth execution.
* Right click on the test and select the test name to run/click on the green triangle button right side of the IDE to run test.
* Results can be seen on the Run window below. Check for the following commands on Run window.

Process finished with exit code 0 – no execution errors.

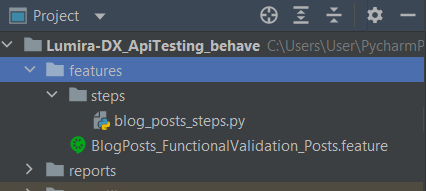
Process finished with exit code 1-execution error. Code need to be corrected.

**API Automation using Requests Tool with Python in Behave Framework:**

Behaviour-driven development (or BDD) is an agile software development technique that encourages collaboration between developers, QA and non-technical or business participants in a software project.

[Behave](https://pypi.python.org/pypi/behave) uses tests written in a natural language style, backed up by Python code.

* Create a new Project
* Install cucumber plugin along with Gherkin plugin in to the project
* Add Request and JsonPath to the project.
* Right click on project🡪New🡪directory(folder)🡪features
* Right click on the features🡪New🡪directory🡪steps
* Right click on features🡪New🡪File-->name of the feature file with “.feature” extension-will be saved as Cucumber feature file. Eg: BlogPosts\_FunctionalValidation\_Posts.feature



* Write the scenarios using gherkin sentences.
* Go to terminal give following command:

**behave features\.BlogPosts\_FunctionalValidation\_Posts.feature**

To run the feature file.

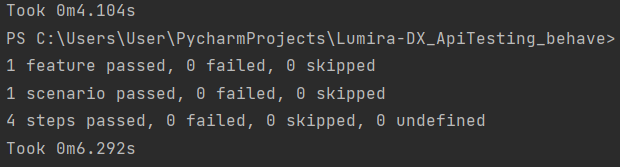
* In terminal all the unimplemented steps are displayed.
* Copy the steps and right click on steps directory🡪New🡪PythonFile🡪name of the file.

Eg: blog\_posts\_steps.py

* Paste the unimplemented methods and write the Requests Python code
* Run the feature file with below command.

**behave features\.BlogPosts\_FunctionalValidation\_Posts.feature**

* Check in the terminal for the code errors and result.

****

**Allure reports:**

* Used Allure reports for reporting in Behave.
* Go to cmd prompt and type

**Pip install allure-behave**

* Add the allure-behave to the project.
* Go to terminal/Command prompt and type the following commands:

**behave -f allure\_behave.formatter:AllureFormatter -o reports/** features

(To run the all feature files under features package. To run specific feature file /feature file name)

* This will create a reports folder under project with Json files.
* To convert Json into HTML reports use following command.

**allure serve reports/**

* Opens in browser**.**

**Note:** Allure reports doesn’t work for community versions on python. To make it work please download the zip file of command line tool of allure and add the bin path to the environmental variables. Execute below commands on command prompt/Power shell command prompt.

**behave -f allure\_behave.formatter:AllureFormatter -o reports/** features

**allure serve reports/**

**================================ END =====================================**