**Steps to Trigger Test Scenarios**

To run the tests from command line please use the following steps:

* Open the command line
* Copy the command and execute “*behave -f pretty <path\_of\_the\_feature\_files\_directory>*”

This command will execute all the feature files, subsequently covering all the scenarios and running all the steps.

By running above command, you will be able to see all the steps being executed sequentially along with the output of each scenario and the results summary.

In order to see the graphical representation of the above execution, execute the following command using Allure:

* Copy the command and execute “*behave -f allure\_behave.formatter:AllureFormatter -o <allure result folder> <Path of the Feature Files Directory>*” to create the reports in json format
* Copy the command and execute “*allure generate --clean <allure result directory> -o <allure report directory>*” in order to create the allure reports summary.
* Copy the command and execute “*allure open <allure report directory>”* to view the graphical representation

**Note: In order to run the above commands successfully, underlying system should have python3.7, behave, behave-allure along with cucumber framework preinstalled.**

**Test Cases Creation**

For this assignment, we have implemented cucumber framework with python language.

We have created two separate feature files as per two routes /albums and /users ,covering following scenarios to check reliability testing of Rest API.

* **GET**: To check JSON response and status code through get method
* **POST**: To check added JSON is getting updated and verified status code
* **PUT**: To check value is getting updated and verified status code
* **DELETE**: To check JSON is getting deleted and verified status code
* **FILTER**: To check values are getting filtered as per given input and verified status code
* **NESTED**: To check values are getting fetched as per given input and verified status code

We have created multiple use cases which cover Sunny and rainy-day scenarios as follows.

* **stepVerifyid** : To verify if album’s id is unique.
* **stepVerifyTitle**: To verify if title is unique.
* **stepInvalidId**: To verify Id is invalid
* **stepInvalidTitle**: To verify provided Title is invalid
* **step\_UniqueEmail**: To verify if Email-id is unique
* **step\_InvalidEmail**: To verify if provided Email-id is invalid