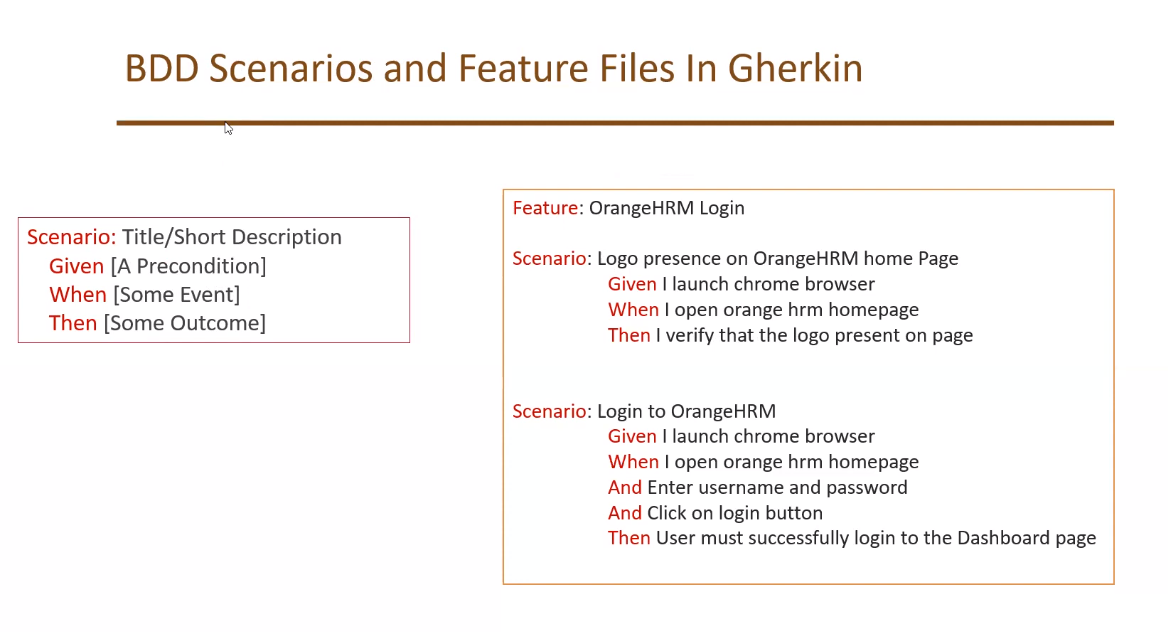
Introduction to BDD (Behavior-driven development):

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

**Except** BDD all other frame works are purely technical based.

What is behave?

* Behave is behavior-driven development, python style.
* Behave operates on directories containing.
* Feature files written by your business Analyst / sponsor / whoever with your behavior scenarios in it, and
* A steps directory with python step implementations for the scenarios.



BDD Scenarios and Feature Files in Gherkin Lang:

* Every Feature file has **.feature** extension.
* Every **feature** file consists multiple **scenarios**. i.e., A Feature is like module in an application.
* Every scenario contains multiple **steps**.
* **Feature, Scenario, Scenario Outline, Given, When, Then, And, But, Example & Background** are Gherkin keywords.
* Refer <https://pypi.org/project/behave/>.

**Given** – A Pre-Condition (ex: Launch browser)

**When, And** – Some Event/Actions (ex: enter URL, enter user name & password)

**Then** – Some outcome/Assertion/Expected result

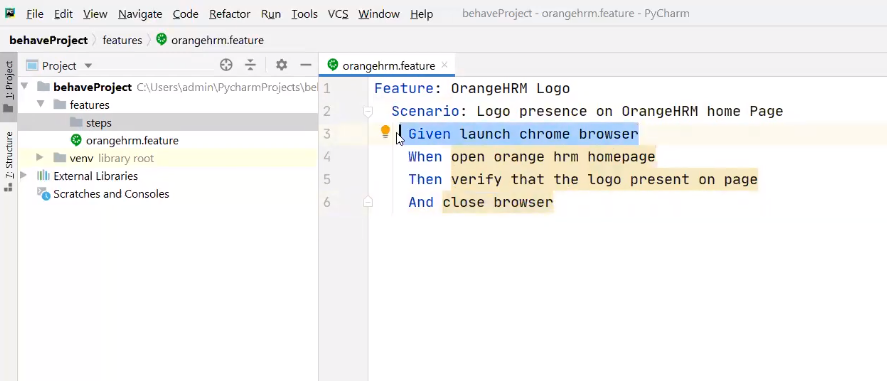
Environment Setup:

1. Install Python
2. Install PyCharm IDE
3. Install selenium library (pip install selenium)
4. Install behave (pip install behave)

Project Creation:

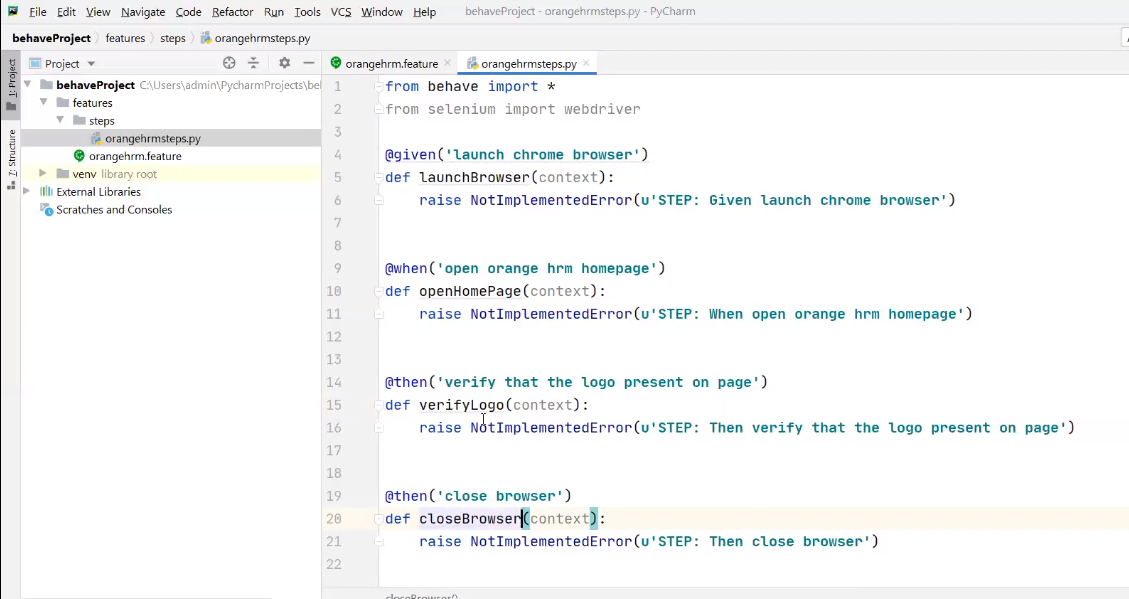
1. Create a project in PyCharm.
2. Add Selenium & Behave in project interpreter.
3. Download all browser drivers form <https://www.selenium.dev/downloads/> and place them in a folder of CWD.
4. Create a **Features** folder

Under this features folder create **steps** folder

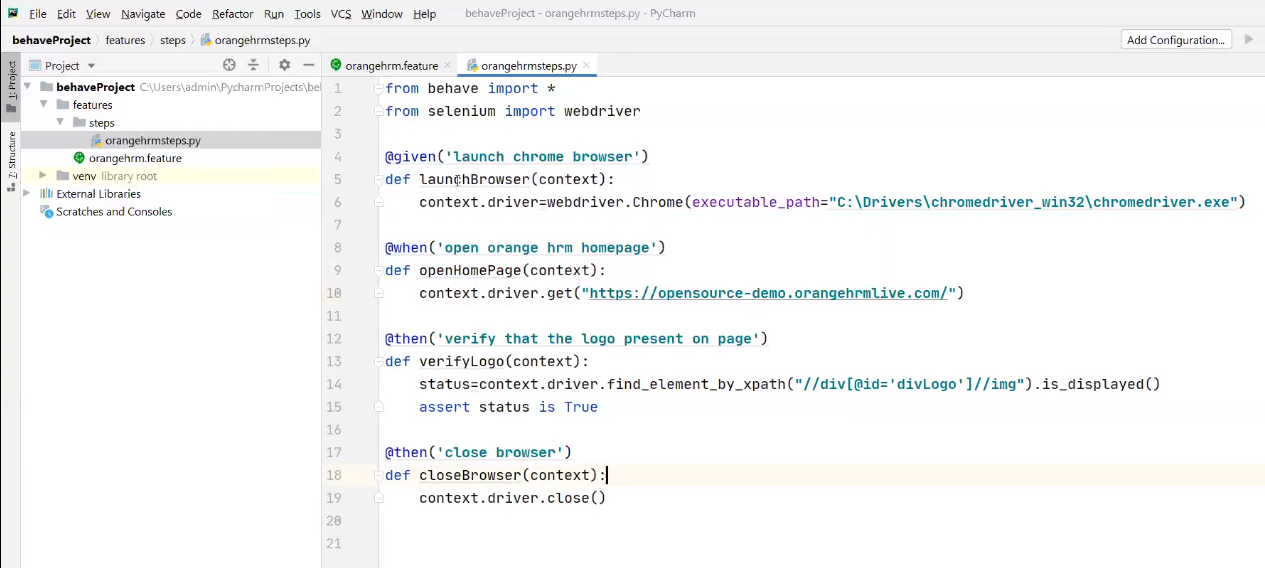
1. In feature folder create an example. feature file and create below scenario
2. Execute the above feature file by PyCharm terminal like **behave features\abc.feature.**

**Or** To execute all feature file under features folder we need execute command **behave .\features**

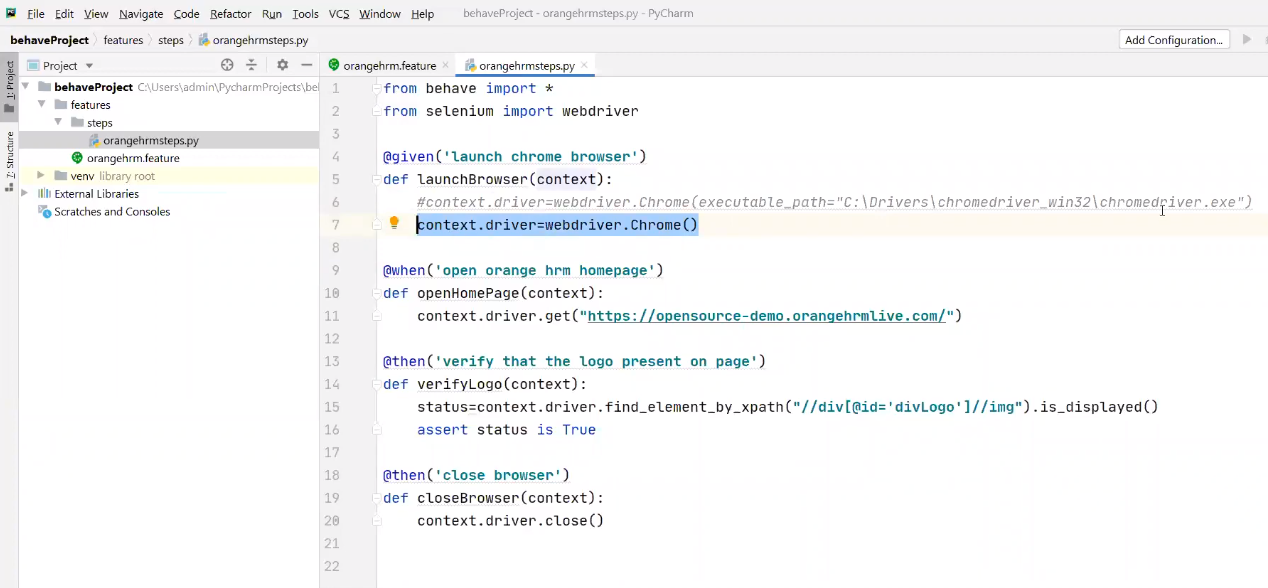
1. When we execute feature file by default it gives step definitions samples in terminal.
2. Create step definitions like below in example.py file under steps folder.



**Note:** Every variable which we declared under step definition methods we need to prefix with context i.e., **context.variable\_name**

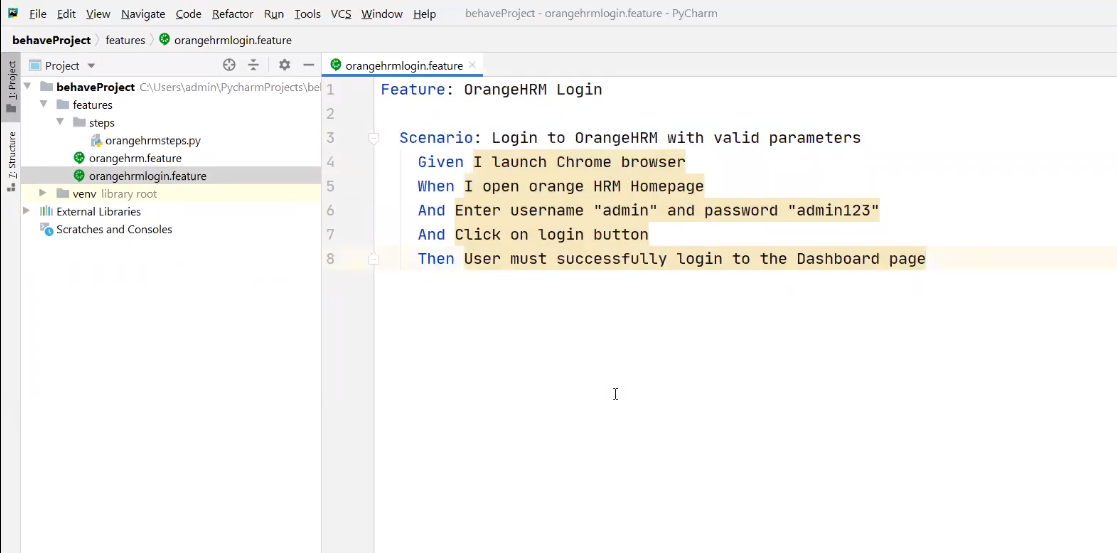


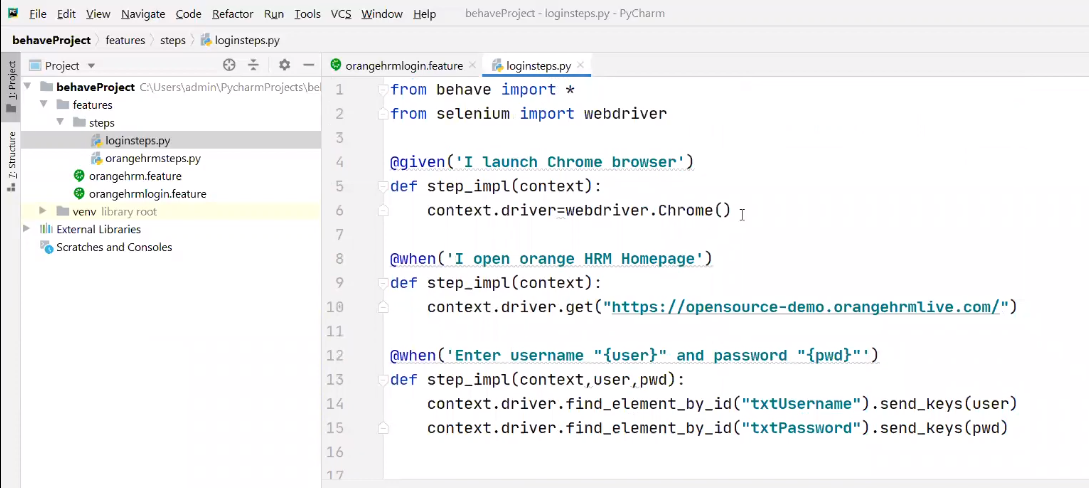
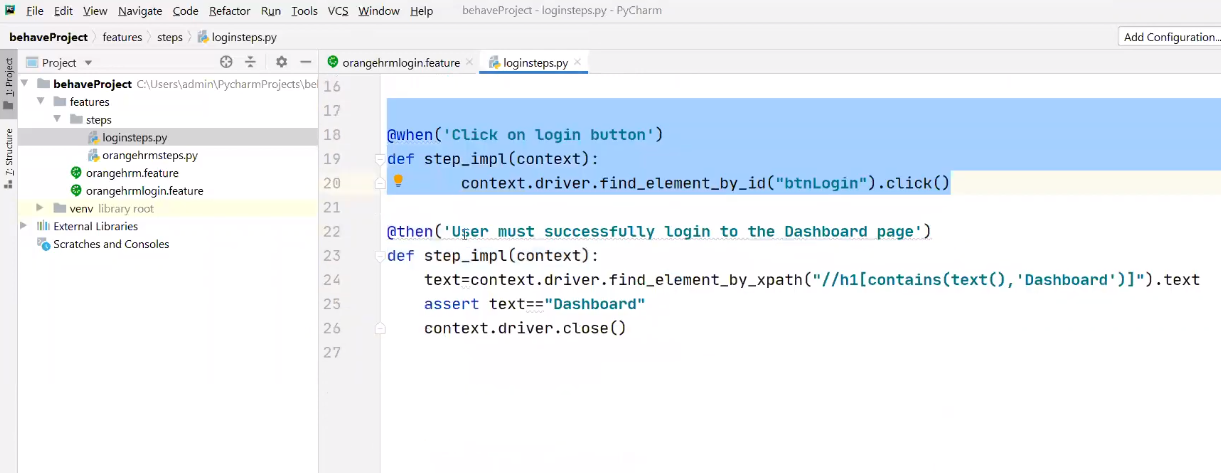
**Note:** If we place all our browser driver under python\scrips folder (this folder should add in environmental path) then we no need to mention path of driver explicitly **i.e.,**



**Step Parameters:**

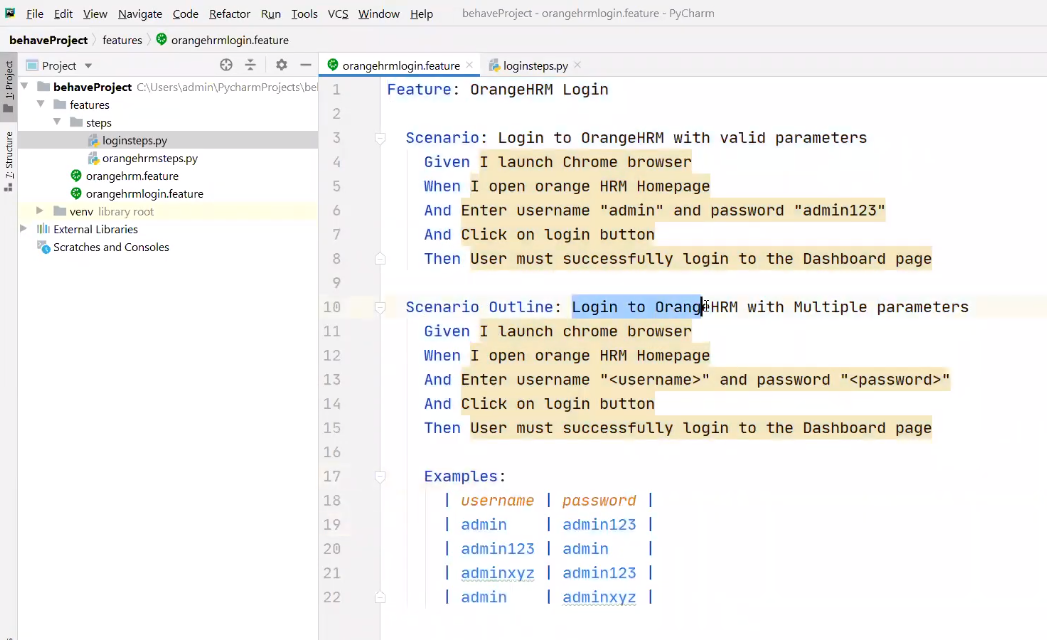
**Note:** When every we want to pass parameters into a step definition, first we need to mention those parameters with double codes in scenarios like below.



Once we write scenarios with parameters in steps, then we need to define step definition by double codes & parenthesis of parameters with any given name **i.e.,**  

**Note:** If we have multiple parameters to pass into step definition then we must go for scenario outline.

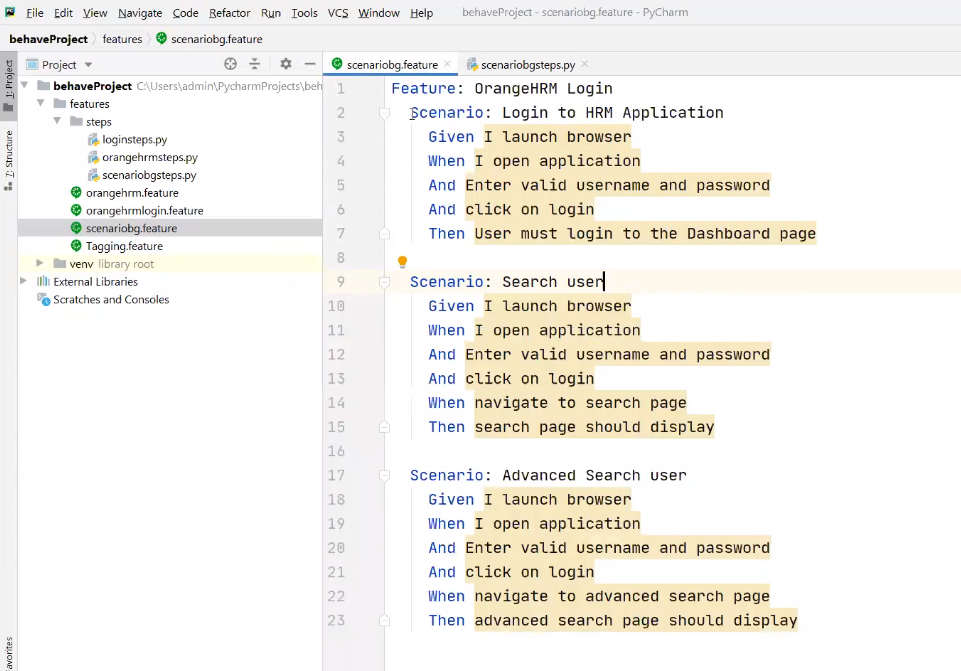
**Scenario Outline:**

Create a Scenario Outline with Example like below, 

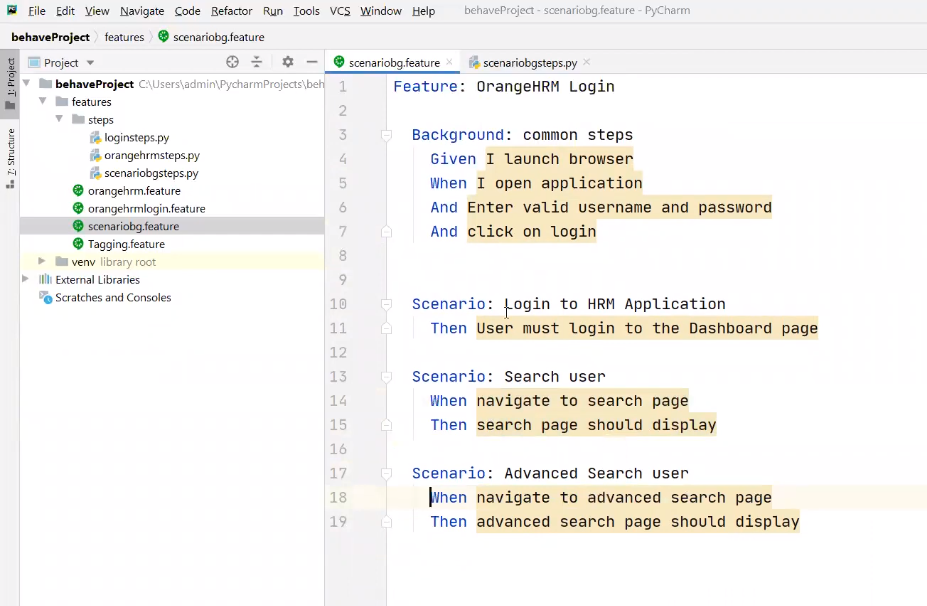
Then create step definition and execute.

**Background:** Executing number of steps **before** each scenario.

Create a feature file with multiple scenarios like below.



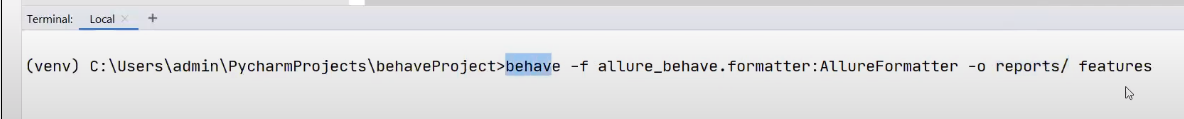
Now write all common steps into a background i.e.,



Allure Reports:

1. Create a folder called Reports in CWD of our project
2. Install allure-behave (pip install allure-behave)
3. Add allure-behave in project interpreter
4. Execute below command

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



1. Once we execute above command, it will create all json report files in report folder
2. Now we need to convert those json report files into html by executing below command

**allure serve reports/**

1. Once we execute above command, we will get all html reports in temp directory (we can get temp directory path in terminal)

