**Cucumber**

Cucumber is a testing tool that supports Behaviour Driven Development (BDD) framework. It defines application behaviour using simple English text, defined by a language called Gherkin.

Cucumber allows automation functional validation that is easily read and understood. Cucumber was initially implemented in Ruby and then extended to Java framework. Both the tools support native JUnit.

In order to get better advantage of the software testing, organizations are nowadays taking a step forward. They implement important acceptance test scenarios while development is in-progress. This approach is commonly known as **Behaviour Driven Development** (BDD).

Behaviour Driven Development gives us an opportunity to create test scripts from both the developer’s and the customer’s prospective as well. So, in the beginning, developers, project managers, QAs, user acceptance testers and the product owner (stockholder), all get together and brainstorm about which test scenarios should be passed in order to call this software/application successful. This way they come up with a set of test scenarios. All these test scripts are in simple English language, so it serves the purpose of documentation also.

Example

If we are developing a user authentication feature, then the following can be few key test scenarios, which needs to get passed in order to call it a success.

* The user should be able to login with correct username and correct password.
* The user should not be able to login with incorrect username and correct password.
* The user should not be able to login with correct username and incorrect password.

## Advantages of Cucumber Over Other Tools

* Cucumber supports different languages like Java.net and Ruby.
* It acts as a bridge between the business and technical language. We can accomplish this by creating a test case in plain English text.
* It allows the test script to be written without knowledge of any code, it allows the involvement of non-programmers as well.
* It serves the purpose of end-to-end test framework unlike other tools.
* Due to simple test script architecture, Cucumber provides code reusability.

**Gherkin**

The language, in which this executable feature files is written, is known as **Gherkin**. Gherkin is a plain English text language, which helps the tool - Cucumber to interpret and execute the test scripts.

**Feature**

A **Feature** can be defined as a standalone unit or functionality of a project.

## Feature Files

The file, in which Cucumber tests are written, is known as **feature files**. It is advisable that there should be a separate feature file, for each feature under test. The extension of the feature file needs to be “. feature”.

One can create as many feature files as needed. To have an organized structure, each feature should have one feature file.

For Example −

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Feature** | **Feature File name** |
| 1 | User Login | userLogin.feature |
| 2 | Registration | registration. feature |
| 3 | Add Product | addProduct.feature |
| 4 | Delete Account | deleteAccount.feature |

The naming convention to be used for feature name, feature file name depends on the individual’s choice. There is no ground rule in Cucumber about names.

A simple feature file consists of the following keywords/parts −

* **Feature** − Name of the feature under test.
* **Description** (optional) − Describe about feature under test.
* **Scenario** − What is the test scenario.
* **Given** − Prerequisite before the test steps get executed.
* **When** − Specific condition which should match in order to execute the next step.
* **Then** − What should happen if the condition mentioned in WHEN is satisfied.