**Cucumber Advanced -**

So far we are good just because we have only a few couple of scenarios in the feature file. However, when you work on any real-life project, you would have many scenarios in one feature file and there will be many other feature files full of scenarios. As the name suggests ***Feature***, we tend to create a separate feature file for every other feature or functionality in an application.

We also try to keep all the related scenarios with in the same feature file and this is the reason we end up having many scenarios in it.

Every scenario comes with it's own prerequisites. To handle the same cucumber gives us many useful functionalities like

* ***Tags***
* ***Hooks & Tagged Hooks***
* ***Execution Order of Hooks***
* ***Background***

Things work absolutely fine till the time we run every feature and all the scenarios under it as all together.

But what if you do not want to run all the scenarios together and like to run a few selected scenarios from different feature files together.

**What are Cucumber Tags?**

## Suppose we have many different feature files that cover all the different functionality of the application.

## Now there can be a certain situation in the project where we have to execute just a **SmokeTests**, **RegressionTests, functional Tests** etc

## One approach is that you start creating new feature files with the name of the type like **SmokeTests.features** or **End2EndTests.feature** and copy-paste your existing tests in the same

## But this would make the project filthy and would require more maintenance in future.

## Organize your scenario execution by using **tags** in feature file. We can define each scenario with a useful tag.

## Later, in the runner file, we can decide which specific tag (and so as the scenario(s)) we want Cucumber to execute. Tag starts with “**@**”. After “**@**” you can have any relevant text to define your tag like **@SmokeTests**

## **And just** specify the tags names in the **CucumberOptions** as

## **tags = "@SmokeTests"**

**Things to Note:**

1. Few scenarios are part of the Smoke Test, Regression Test, and End2End Test.
2. Few scenarios are part of two or more Test Types. For example, the first test is considered as Smoke as well as Regression.
3. Few scenarios are not at all tagged

## **So to achieve Logically ANDing and ORing Tags**

## **Execute all tests tagged as @SmokeTest OR @RegressionTest**

## OR means scenarios that are tagged either @SmokeTest OR @RegressionTest.

## **Execute all tests tagged as @SmokeTest AND @RegressionTest**

## **AND** means scenarios that are tagged with both tag @SmokeTest **AND** @RegressionTest.

## **How to Ignore Cucumber Tests**

## **Special Character ~** is used to skip the tags.

## This also works both for Scenarios and Features. And this can also works in conjunction with AND or OR.

## Eg. **Execute all tests of the feature tagged as @FunctionalTests but skip scenarios tagged as @SmokeTest**

## **What are Hooks in Cucumber?**

## Cucumber **hooks** are blocks of code that run **before** or **after** each scenario.

## We can define them anywhere in the project using the methods **@Before** and **@After**. (Step Definitions)

* As the name suggests, @before hook gets executed before any other test scenario, and @after hook gets executed after executing the scenario.
* An important thing to note about the after hook is that even in case of test fail, after hook will execute for sure.

## **Cucumber Hooks** allows us to better manage the code workflow and helps us to reduce the code redundancy.

## **Why Cucumber Hooks?**

## need to perform the prerequisite steps before testing any test scenario.

* Starting a webdriver
* Setting up DB connections
* Setting up test data
* Setting up browser cookies
* Navigating to certain page
* or anything before the test

In the same way, there are always after steps as well of the tests like:

* Killing the webdriver
* Closing DB connections
* Clearing the test data
* Clearing browser cookies
* Logging out from the application
* Printing reports or logs
* Taking screenshots on error
* or anything after the test

So Cucumber hooks are the best choice to use.cucumber supports only two hooks (Before & After)

### **How to set the Order or Priority of Cucumber Hooks?**

* **@Before(order = int)** : This runs in increment order, means value 0 would run first and 1 would be after 0.
* **@After(order = int)** : This runs in decrements order, means apposite of @Before. Value 1 would run first and 0 would be after 1.