## 5.10 Tagged Hooks



This section will guide you to understand :

* What is a tagged hook in Cucumber
* How a tagged hook works in Cucumber

**Development Environment:**

* JRE: OpenJDK Runtime Environment 11.0.2
* Eclipse IDE for Enterprise Java Developers v2019-03 (4.11.0)
* TestNG
* Selenium jars
* Cucumber jars

This guide has two subsections, namely:

## 5.10.1 Steps involved in adding tagged hooks in Cucumber

5.10.2 Pushing code to your GitHub repositories

## 

## **Step 5.10.1:** Tagged Hooks in Cucumber

* Tagged Hooks are used where you need to perform different tasks before and after scenarios.
* The first step is to annotate the required scenarios using ***@ + AnyName*** at the top of the Scenario. For this example, you just annotate each scenario with the sequence order of it, like ***@First, @Second & @Third***.

FEATURE FILE

|  |
| --- |
| Feature: Test Tagged Hooks    @First  Scenario: This is First Scenario  Given **this** is the first step  When **this** is the second step  Then **this** is the third step    @Second  Scenario: This is Second Scenario  Given **this** is the first step  When **this** is the second step  Then **this** is the third step    @Third  Scenario: This is Third Scenario  Given **this** is the first step  When **this** is the second step  Then **this** is the third step |

* Create a step definition file and print the execution order of the steps in the console.

STEP DEFINITION

|  |
| --- |
| package stepDefinition;    **import** cucumber.api.java.en.Given;  **import** cucumber.api.java.en.Then;  **import** cucumber.api.java.en.When;    public class Hooks\_Steps {    @Given("^this is the first step$")  public **void** This\_Is\_The\_First\_Step(){  System.out.println("This is the first step");  }    @When("^this is the second step$")  public **void** This\_Is\_The\_Second\_Step(){  System.out.println("This is the second step");  }    @Then("^this is the third step$")  public **void** This\_Is\_The\_Third\_Step(){  System.out.println("This is the third step");  }    }  Given **this** is the first step  When **this** is the second step  Then **this** is the third step |

* Define *tagged hooks* in Hooks class file. Hooks can be used like ***@Before(“@TagName”)***. Create before and after hooks for every scenario.

HOOKS

|  |
| --- |
| **import cucumber.api.java.After;**  **import cucumber.api.java.Before;**    **public class Hooks {**    **@Before**  **public void beforeScenario(){**  **System.out.println("This will run before the every Scenario");**  **}**    **@After**  **public void afterScenario(){**  **System.out.println("This will run after the every Scenario");**  **}**    **@Before("@First")**  **public void beforeFirst(){**  **System.out.println("This will run only before the First Scenario");**  **}**    **@Before("@Second")**  **public void beforeSecond(){**  **System.out.println("This will run only before the Second Scenario");**  **}**    **@Before("@Third")**  **public void beforeThird(){**  **System.out.println("This will run only before the Third Scenario");**  **}**    **@After("@First")**  **public void afterFirst(){**  **System.out.println("This will run only after the First Scenario");**  **}**    **@After("@Second")**  **public void afterSecond(){**  **System.out.println("This will run only after the Second Scenario");**  **}**    **@After("@Third")**  **public void afterThird(){**  **System.out.println("This will run only after the Third Scenario");**  **}**  **}** |

* Run the feature file

**Step 5.10.2:** Pushing code to your Git repositories

Open your command prompt and navigate to the folder where you have created your files

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master