

Background in Cucumber is used to define a step or series of steps that are common to all the tests in the feature file. It allows you to add some context to the scenarios for a feature where it is defined. A Background is much like a scenario containing a number of steps. But it runs before each and every scenario were for a feature in which it is defined.

For example, to purchase a product on any E-Commerce website, you need to do the following steps:

*Navigate to Login Page
Submit UserName and Password*

After these steps only you will be able to add a product to your *cart/basket* and able to perform the payment. Now as we are in a feature file where we will be testing only the *Add to Cart or Add to Bag* functionality, these tests become common for all tests. So instead of writing them again and again for all tests, we can move it under the *Background keyword*.

Background in Cucumber

Let's start with a simple exercise to build the understanding of Background usage in Cucumber test. If we create a feature file of the scenario we explained above, this is how it will look like:

Feature File

Feature: Test Background Feature

Description: The purpose of **this** feature is to test the Background keyword

Background: User is Logged In

Given I navigate to the login page
When I submit username and password
Then I should be logged in

Scenario: Search a product and add the first product to the User basket

Given User search **for** Lenovo Laptop
When Add the first laptop that appears in the search result to the basket
Then User basket should display with added item

Scenario: Navigate to a product and add the same to the User basket

Given User navigate **for** Lenovo Laptop
When Add the laptop to the basket
Then User basket should display with added item

In the above example, we have two different scenarios where a user is adding a product from search and directly from the product page. But the common step is to login to website for both the scenario. *This is why we create another Scenario for Login but named it as Background rather than a Scenario.* So that it executes for both the Scenarios

Step Definitions

```
package stepDefinition;

import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.When;

public class BackGround_Steps {

    @Given("^I navigate to the login page$")
    public void i_navigate_to_the_login_page() throws Throwable {
        System.out.println("I am at the LogIn Page");
    }

    @When("^I submit username and password$")
    public void i_submit_username_and_password() throws Throwable {
        System.out.println("I Submit my Username and Password");
    }

    @Then("^I should be logged in$")
    public void i_should_be_logged_in() throws Throwable {
        System.out.println("I am logged on to the website");
    }

    @Given("^User search for Lenovo Laptop$")
    public void user_searched_for_Lenovo_Laptop() throws Throwable {
        System.out.println("User searched for Lenovo Laptop");
    }

    @When("^Add the first laptop that appears in the search result to the basket$")
    public void add_the_first_laptop_that_appears_in_the_search_result_to_the_basket(
        System.out.println("First search result added to bag");
    }

    @Then("^User basket should display with added item$")
    public void user_basket_should_display_with_item() throws Throwable {
        System.out.println("Bag is now contains the added product");
    }

    @Given("^User navigate for Lenovo Laptop$")
    public void user_navigate_for_Lenovo_Laptop() throws Throwable {
        System.out.println("User navigated for Lenovo Laptop");
    }

    @When("^Add the laptop to the basket$")
    public void add_the_laptop_to_the_basket() throws Throwable {
        System.out.println("Laptop added to the basket");
    }

}
```

Just printing the appropriate logs in the steps. But we would bring the advance usage of the same in our next series of ***Designing Automation Framework with Cucumber.***

Output

```
Feature: Test Background Feature
  Description: The purpose of this feature is to test the Background keyword
```

```
I am at the LogIn Page
I Submit my Username and Password
I am logged on to the website
User searched for Lenovo Laptop
First search result added to bag
Bag is now contains the added product
```

```
I am at the LogIn Page
I Submit my Username and Password
I am logged on to the website
User navigated for Lenovo Laptop
Laptop added to the basket
Bag is now contains the added product
```

Note: Hope you noticed the impact. The background ran two times in the feature before each scenario.

Background with Hooks

This is so interesting to see the working of *Background with Hooks*. The background is run before each of your scenarios but after any of your **@Before hook**.

To get it straight, let's assign a task to the Before & After Hook in the same test.

```
@Before: Print the starting logs
@Before: Start browser and Clear the cookies
@After: Close the browser
@After: Print the closing logs
```

Hooks File

```
package utilities;

import cucumber.api.java.After;
import cucumber.api.java.Before;
```

```

public class Hooks {

    @Before(order=1)
    public void beforeScenario(){
        System.out.println("Start the browser and Clear the cookies");
    }

    @Before(order=0)
    public void beforeScenarioStart(){
        System.out.println("-----Start of Scenario-----");
    }

    @After(order=0)
    public void afterScenarioFinish(){
        System.out.println("-----End of Scenario-----");
    }

    @After(order=1)
    public void afterScenario(){
        System.out.println("Log out the user and close the browser");
    }

}

```

Output

Feature: Test Background Feature

Description: The purpose of **this** feature is to test the Background keyword

```

-----Start of Scenario-----
Start the browser and Clear the cookies
I am at the LogIn Page
I Submit my Username and Password
I am logged on to the website
User searched for Lenovo Laptop
First search result added to bag
Bag is now contains the added product
Log out the user and close the browser
-----End of Scenario-----

```

```

-----Start of Scenario-----
Start the browser and Clear the cookies
I am at the LogIn Page
I Submit my Username and Password
I am logged on to the website
User navigated for Lenovo Laptop
Laptop added to the basket
Bag is now contains the added product
Log out the user and close the browser
-----End of Scenario-----

```

Good practices for using Background

It is really necessary to understand the right usage of *Background*. As hooks as well give similar kind of functionality and moreover almost all the tasks can be done by hooks as well. This is why it is critical to use the background in the right place in the test.

Feature Dependency

Any feature level dependency should be tie with the background and any scenario level dependency should be tie with hooks.

Keep Background short.

You're expecting the user to actually remember this stuff when reading your scenarios. If the background is more than 4 lines long, can you move some of the irrelevant details into high-level steps? See Calling Steps from Step Definitions.

Make your Background section vivid.

*You should use colorful names and try to tell a story, because the human brain can keep track of stories much better than it can keep track of names like **"User A"**, **"User B"**, **"Site 1"**, and so on.*