



Cucumber

Class 3

Agenda

What is Cucumber data driven testing?

Data Driver testing with and Without Examples Keyword

What is Data Table in Cucumber?

Parameterization in Cucumber?

In Cucumber we can easily parameterize the tests.
To parameterize the values, put them in double quotes

```
1 #Author: syntax team
2 Feature: Add Employee
3
4 Background:
5     Given I logged in into OrangeHrm
6
7 Scenario: Add Employee
8     When I click on PIM link
9     And I click on Add Employee link
10    And I add Employees "John", "Smith" and "HQ"
11    And I click on save button
12    Then I see employee "John", "Smith" is added successfully
```

```
@Then("I see employee {string}, {string} is added successfully")
public void i_see_employee_is_added_successfully(String fName, String lName) {
    waitForElementBeClickable(addEmployee.empCheck, 20);
    String actualName=addEmployee.empCheck.getText();
    Assert.assertEquals(fName+" "+lName, actualName);
}
```

What is Cucumber data driven testing?

Data Driven Testing, which allows to automatically run a test case multiple times with different input and validation values.

In simple terms, cucumber data driven testing means that we can pass data from cucumber feature files to our test cases.

There are different ways to use the data insertion within Cucumber and outside the Cucumber with external files.

- **Data Driven Testing in Cucumber using Scenario Outline with Examples keyword**
- **Data Driven Testing in Cucumber using Cucumber DataTable**
- **Data Driven Testing in Cucumber using External Files**

Data Driven Testing Using Examples Keyword

Cucumber supports Data Driven testing by the use of the Scenario Outline and Examples section.

With these keywords cucumber allows us easy Data Driven testing to be completed where no changes need to be made to the Java file.

Example keyword can **only** be used with the **Scenario Outline** Keyword.

Scenario Outline – This is used to run the same scenario for 2 or more different set of test data.

Examples – All scenario outlines have to be followed with the Examples section. This contains the data that has to be passed on to the scenario.

Data Driven Testing Using Examples Keyword

```
1 #Author: syntax team
2 Feature: Add Employee
3
4 Background:
5     Given I logged in into OrangeHrm
6
7 Scenario Outline: Add Employee
8     When I click on PIM link
9     And I click on Add Employee link
10    And I add Employees "<FirstName>", "<LastName>" and "<Location>"
11    And I click on save button
12    Then I see employee "<FirstName>", "<LastName>" is added successfully
13
14 Examples:
15     | FirstName | LastName | Location |
16     | Jane      | North   | Tech Office |
17     | Jack      | South   | Smart Office |
18     | James     | West    | HQ          |
```

Data Driven Testing Using Data Tables

With Cucumber data tables, we can pass parameters from feature files in tabular format. And we can then use this data in step definition methods in the form of Lists and Maps.

To use DataTable we do not need to specify any keyword and parameters that will be passed under specific step are applicable only for the that step.

A separate code is needed to understand the test data and then it can be run single or multiple times but again just for the single step, not for the complete test

In Cucumber, we can add data tables in two different formats :

- Data table without a header
- Data table with a header

Data table without a header

The complete scenario is same as what we have done earlier. But the only difference is in this, we are not passing parameters in the step line and even we are not using Examples test data. We declared the data under the step only. So we are using Tables as arguments to Steps.

```
@temp
> Scenario: Add Employee Labels Verification
  Then I see following labels
    | First Name |
    | Middle Name |
    | Last Name |
    | Employee Id |
    | Location |
```

```
@Then("I see following labels")
public void i_see_following_labels(DataTable addEmpLabels) {

    List<String> labels = addEmpLabels.asList();
    System.out.println("----Printing labels from cucumber dataTable----");
    for (String label : labels) {
        System.out.println(label);
    }
}
```


Data table with a header

We add the header to the Data table when we have more columns in the data table, so that we can improve the readability of our cucumber scenarios and scripts. Whenever we have multiple columns of test data present the best approach will be to use of Maps in Data Tables.

@temp

Scenario: Error message Validation

When I enter invalid username and password I see errorMessage

UserName	Password	ErrorMessage
Admin	Admin123	Invalid Credentials
Admin1	Syntax123!	errorMessage

```
@When("I enter invalid username and password and see error message")
public void i_enter_invalid_username_and_password_and_see_error_message(DataTable wrongCredentials) {
    List<Map<String, String>> maps = wrongCredentials.asMaps(String.class, String.class);
    for (Map<String, String> map : maps) {
        sendText(login.userName, map.get("UserName"));
        sendText(login.password, map.get("Password"));
        click(login.loginBtn);
        String actualError=login.message.getText().trim();
        String expectedError=map.get("ErrorMessage");
        Assert.assertEquals(expectedError, actualError);
    }
}
```