n the previous chapter of ***Data Tables in Cucumber***, we consider a very simple example of passing *UserName* and *Password* in the step. Let's take a little complex scenario where a good amount of data is required to pass in the step. Or what is there are multiple columns of test data is present. How would you handle it? The answer is to make a Use of ***Maps in Data Tables***.

Maps in Data Tables

Maps in Data Tables can be used if different ways. ***Headers*** can also be defined for the *data tables*. A same step can be executed multiple times with different set of test data using ***Maps***.

Maps in Data Tables with Header

In the previous chapter of ***Data Tables in Cucumber***, we pass *Username* & *Password* without Header, due to which the test was not much readable. What if there will be many columns. The basic funda of BDD test is to make the Test in Business readable format, so that business users can understand it easily. Setting Header in Test data is not a difficult task in Cucumber. take a look at a below Scenario.

Feature File Scenario

```
Scenario: Successful Login with Valid Credentials
        Given User is on Home Page
        When User Navigate to LogIn Page
        And User enters Credentials to LogIn
        | Username   | Password |
    | testuser 1 | Test@153 |
        Then Message displayed Login Successfully
```

The implementation of the above step will be like this:

```
@When("^User enters Credentials to LogIn$")
        public void user_enters_testuser_and_Test(DataTable usercredentials) throws Throw

                //Write the code to handle Data Table
                List<Map<String,String>> data = usercredentials.asMaps(String.class,Strin
                driver.findElement(By.id("log")).sendKeys(data.get(0).get("Username"));
            driver.findElement(By.id("pwd")).sendKeys(data.get(0).get("Password"));
            driver.findElement(By.id("login")).click();
```

```
}
```

Maps in Data Tables with Multiple Test Data

In this test we will pass *Username* and *Password* two times to the test step. So our test should enter *Username* & *Password* once, click on *LogIn* button and repeat the same steps again.

Feature File Scenario

```
Scenario: Successful Login with Valid Credentials
        Given User is on Home Page
        When User Navigate to LogIn Page
        And User enters Credentials to LogIn
        | Username   | Password |
    | testuser 1 | Test@153 |
    | testuser 2 | Test@154 |
        Then Message displayed Login Successfully
```

The implementation of the above step will be like this:

```
        @When("^User enters Credentials to LogIn$")
        public void user_enters_testuser_and_Test(DataTable usercredentials) throws Throw

                //Write the code to handle Data Table
                for (Map<String, String> data : usercredentials.asMaps(String.class, Stri
                        driver.findElement(By.id("log")).sendKeys(data.get("Username"));
                    driver.findElement(By.id("pwd")).sendKeys(data.get("Password"));
                    driver.findElement(By.id("login")).click();
                        }

        }
```

Map Data Tables to Class Objects

Luckily there are easier ways to access your data than *DataTable*. For instance you can create a Class-Object and have Cucumber map the data in a table to a list of these.

Feature File Scenario

```
Scenario: Successful Login with Valid Credentials
        Given User is on Home Page
        When User Navigate to LogIn Page
        And User enters Credentials to LogIn
        | Username   | Password |
    | testuser 1 | Test@153 |
    | testuser 2 | Test@154 |
        Then Message displayed Login Successfully
```

The implementation of the above step will be like this:

```
        @When("^User enters Credentials to LogIn$")
        public void user_enters_testuser_and_Test(List<Credentials>  usercredentials) thr

                //Write the code to handle Data Table
                for (Credentials credentials : usercredentials) {
                        driver.findElement(By.id("log")).sendKeys(credentials.getUsername
                    driver.findElement(By.id("pwd")).sendKeys(credentials.getPassword());
                    driver.findElement(By.id("login")).click();
                        }
        }
```

Class Credentials

```
package stepDefinition;

public class Credentials {
        private String username;
        private String password;

        public String getUsername() {
        return username;
    }
        public String getPassword() {
        return password;
    }
}
```