Way2Automation - Tutorial 9 – DataTable (asMaps) in Cucumber

What you will Learn :

- DataTable concept
- Add column names
- DataTable asMaps

DataTable concept

Please read the DataTable concept explained in our previous tutorial. Please read the previous tutorial before you read this one.

Add column names

Let us add one more scenario to the feature that we created in our previous tutorial. Add one row that will have the column names

```
UserRegistration.feature   UserRegStepDef.java   RunnerTest.java
1  Feature: User Registration
2
3    Scenario: User registration with different data
4      Given User is on registration page
5      When user enters following user details
6        | Tom | Parker | tom@email.com | 011-13579 | Bldg1 | 12345 | Delhi |
7        | Jim | Walker | jim@email.com | 022-24680 | Bldg2 | 67890 | New York |
8        | Harry | Roberts | har@email.com | 033-13579 | Bldg3 | 13131 | Mumbai |
9      Then user registration should be successful
10
11     Scenario: User registration with different data having columns
12     Given User is on registration page
13     When user enters following user details having columns
14       | FirstName | LastName | Email | Phne | Address | Zip | City |
15       | Tom | Parker | tom@email.com | 011-13579 | Bldg1 | 12345 | Delhi |
16       | Jim | Walker | jim@email.com | 022-24680 | Bldg2 | 67890 | New York |
17       | Harry | Roberts | har@email.com | 033-13579 | Bldg3 | 13131 | Mumbai |
18     Then user registration should be successful
```

Column names

The first row will behave like a column for us. Other respective rows will behave as data for us.

Save the feature file.

Let us now run this file to generate the step definition methods for the new scenario

```
Feature: User Registration

  Scenario: User registration with different data
    Given User is on registration page
    When user enters following user details
      | Tom | Parker | tom@email.com | 011-13579 | Bldg1
      | Jim | Walker | jim@email.com | 022-24680 | Bldg2
      | Harry | Roberts | har@email.com | 033-13579 | Bld
    Then user registration should be successful

    Scenario: User registration with different data havin
    Given User is on registration page
    When user enters following user details having column
      | FirstName | LastName | Email | Phne | Address | Z
      | Tom | Parker | tom@email.com | 011-13579 | Bldg1
      | Jim | Walker | jim@email.com | 022-24680 | Bldg2
      | Harry | Roberts | har@email.com | 033-13579 | Bld
    Then user registration should be successful
```

Copy the code snippets

```
Console
<terminated> UserRegistration.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (03-Mar-2021, 10:45:10 PM)

Undefined scenarios:
file:///C:/Users/DELL/Desktop/Workspace/CucumberMavenProject/src/test/java/Features/UserRegistration.featur

2 Scenarios (1 undefined, 1 passed)
6 Steps (1 skipped, 1 undefined, 4 passed)
0m3.877s


You can implement missing steps with the snippets below:

@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(io.cucumber.datatable.DataTable dataTable) {
    // Write code here t                                  into concrete actions
    // For automatic tra                                  ble to one of
    // E, List<E>, List<                                  Map<K,V> or
    // Map<K, List<V>>.                                   teger, Float,
    // Double, Byte, Sho                                  gDecimal.
    //
    // For other transfo                                  a DataTableType.
    throw new io.cucumbe
}
```

Paste them in the same step definition file that was created in previous tutorial, see below

```
UserRegistration.feature   UserRegStepDef.java   RunnerTest.java
 8 public class UserRegStepDef {
 9     @Given("User is on registration page")
10     public void user_is_on_registration_page() {
11         System.out.println("User is on registration page");
12     }
13
14     @When("user enters following user details")
15     public void user_enters_following_user_details(DataTable dataTable) {
16     List<List<String>> userList = dataTable.asLists(String.class);
17     for(List<String> li : userList) {
18         System.out.println(li);
19         }
20     }
21
22     @When("user enters following user details having columns")
23     public void user_enters_following_user_details_having_columns(io.cucumber.datatable.DataTable dataTable) {
24         // Write code here that turns the phrase above into concrete actions
25         // For automatic transformation, change DataTable to one of
26         // E, List<E>, List<List<E>>, List<Map<K,V>>, Map<K,V> or
27         // Map<K, List<V>>. E,K,V must be a String, Integer, Float,
28         // Double, Byte, Short, Long, BigInteger or BigDecimal.
29         //
30         // For other transformations you can register a DataTableType.
31         throw new io.cucumber.java.PendingException();
32     }
```

Remove the comments plus exceptions

```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(io.cucumber.datatable.DataTable dataTable) {

}
```

If you want, instead of writing **io.cucumber.datatable.DataTable**, you can simply write **DataTable** (see below)

```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {

}
```

If you get an error, you have to import the package **io.cucumber.datatable**

```
(DataTable dataTable) {
  DataTable cannot be resolved to a type
  16 quick fixes available:
    Import 'DataTable' (io.cucumber.datatable)
```

<u>DataTable asMaps</u>

As you can see below, there is a method **asMaps()** available

```
12      }
13
14      @When("user ent
15      public void use
16      List<List<Strin
17      for(List<String
18          System.out.
19          }
20      }
21
22      @When("user ent
23      public void use
24          dataTable.asMaps
25      }
```

asMaps() : List<Map<String,String>> - DataTable
asMaps(Type keyType, Type valueType) : List<Map<K,V>> - DataTable

Converts the table to a list of maps of strings. For each row in the body of the table a map is created containing a mapping of column headers to the column cell of that row.
Returns:
a list of maps

Press 'Ctrl+Space' to show Template Proposals
Press 'Tab' from proposal table or click for focu

Let us use that method

```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
    dataTable.asMaps();
}
```

Now, we know that, a map contains values on the basis of key viz key:value pair. Also, a map contains unique keys.

In our current case, both key and value would be of 'String' type

```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
    dataTable.asMaps(String.class, String.class);
}
```

Now, as you can see below, asMaps() method returns "List of Map"

```
dataTable.asMaps(
```

<Object, Object> List<Map<Object, Object>> io.cucumber.datatable.DataTable.asMaps(Type keyType, Type valueType)

This map will contain key-value pair of type 'String'

```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
List<Map<String, String>> userList =    dataTable.asMaps(String.class, String.class);
```

Next, let us import this map from java.util package

```
List<Map<
}
```

Map cannot be resolved to
13 quick fixes available:
Import 'Map' (java.util)

So we have

UserRegistration.feature | UserRegStepDef.java | RunnerTest.java

```java
package StepDefinitions;
import java.util.List;
import java.util.Map;
import io.cucumber.datatable.DataTable;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class UserRegStepDef {
    @Given("User is on registration page")
    public void user_is_on_registration_page() {
        System.out.println("User is on registration page");
    }

    @When("user enters following user details")
    public void user_enters_following_user_details(DataTable dataTable) {
    List<List<String>> userList = dataTable.asLists(String.class);
    for(List<String> li : userList) {
        System.out.println(li);
        }
    }

    @When("user enters following user details having columns")
    public void user_enters_following_user_details_having_columns(DataTable dataTable) {
    List<Map<String, String>> userList = dataTable.asMaps(String.class, String.class);
```

Next, let us print the list 'userList'

```java
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
List<Map<String, String>> userList = dataTable.asMaps(String.class, String.class);

System.out.println(userList);
}
```

Save the file

Let us now re-run the feature file

```
Feature: User Registration

  Scenario: User registration with different data
    Given User is on registration page
    When user enters following user details
      | Tom | Parker | tom@email.com | 011-13579 | Bldg1
      | Jim | Walker | jim@email.com | 022-24680 | Bldg2
      | Harry | Roberts | har@email.com | 033-13579 | Bld
    Then user registration should be successful

    Scenario: User registration with different data havin
    Given User is on registration page
    When user enters following user details having column
      | FirstName | LastName | Email | Phne | Address | Z
      | Tom | Parker | tom@email.com | 011-13579 | Bldg1
      | Jim | Walker | jim@email.com | 022-24680 | Bldg2
      | Harry | Roberts | har@email.com | 033-13579 | Bld
    Then user registration should be successful
```

Notice the console below and see that our map is getting printed

```
<terminated> UserRegistration.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (04-Mar-2021, 8:01:21 PM)
Scenario: User registration with different data # src/test/java/Features/UserRegistration.feature:3
User is on registration page
  Given User is on registration page           # StepDefinitions.UserRegStepDef.user_is_on_registration_page()
[Tom, Parker, tom@email.com, 011-13579, Bldg1, 12345, Delhi]
[Jim, Walker, jim@email.com, 022-24680, Bldg2, 67890, New York]
[Harry, Roberts, har@email.com, 033-13579, Bldg3, 13131, Mumbai]
  When user enters following user details       # StepDefinitions.UserRegStepDef.user_enters_following_user_details(io.cucumber.datata
User registration successful
  Then user registration should be successful   # StepDefinitions.UserRegStepDef.user_registration_should_be_successful()

Scenario: User registration with different data having columns # src/test/java/Features/UserRegistration.feature:11
User is on registration page
  Given User is on registration page                          # StepDefinitions.UserRegStepDef.user_is_on_registration_page()
[{FirstName=Tom, LastName=Parker, Email=tom@email.com, Phne=011-13579, Address=Bldg1, Zip=12345, City=Delhi}, {FirstName=Jim, LastName
  When user enters following user details having columns       # StepDefinitions.UserRegStepDef.user_enters_following_user_details_hav
```

Let us see how this list looks like. Just select and copy the the entire list from console.
Paste it in notepad and arrange it a little bit:

So we have a list containing 3 maps. Each map has key/value pairs. For example, in the first map, one of the keys is ‘FirstName’ and the corresponding value is ‘Tom’. Another key is ‘Zip’ and value is 12345, and so on.

Also, each and every row value will be mapped to respective column headers (FirstName, LastName etc).

Now, a list is order based, it maintains the indexing or the order. The indexing starts from 0. So see below. From this userList, let us try to print the first map. The first map has index 0 and hence we can say userList.get(0)

```
List<List<String>> userList =
for(List<String> li : userLis
    System.out.println(li);
    }
}

@When("user enters following
public void user_enters_follo
List<Map<String, String>> use

System.out.println(userList);
System.out.println(userList.get);
}
```

```
get(int index) : Map<String,String> - List
getClass() : Class<?> - Object
```

So we can say

```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
List<Map<String, String>> userList = dataTable.asMaps(String.class, String.class);

System.out.println(userList);
System.out.println(userList.get(0));
}
```

Save the file

Let us now re-run the feature file

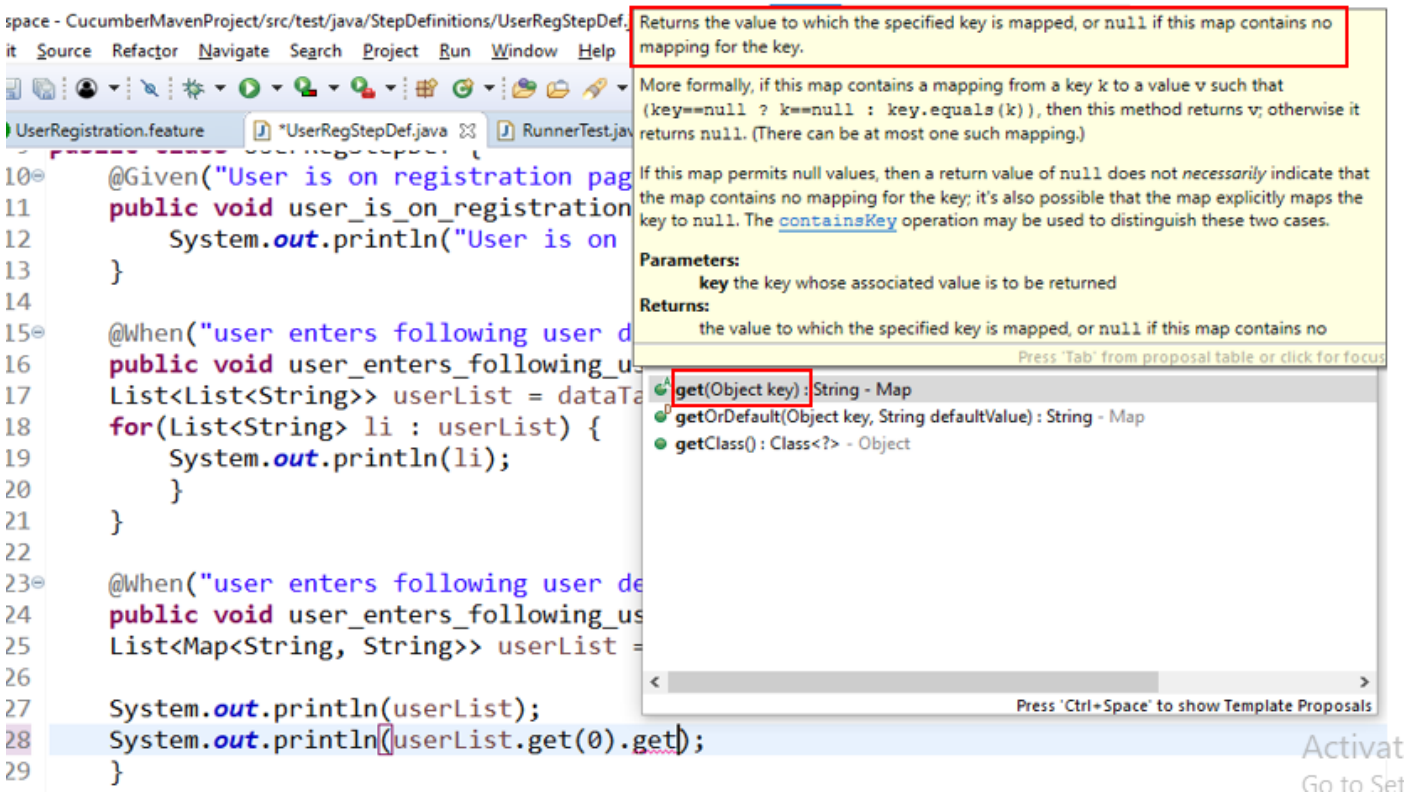
```
Feature: User Registration

  Scenario: User registration with different data
    Given User is on registration page
    When user enters following user details
      | Tom | Parker | tom@email.com | 011-13579 | Bldg1
      | Jim | Walker | jim@email.com | 022-24680 | Bldg2
      | Harry | Roberts | har@email.com | 033-13579 | Bld
    Then user registration should be successful

    Scenario: User registration with different data havin
    Given User is on registration page
    When user enters following user details having column
      | FirstName | LastName | Email | Phne | Address | Z
      | Tom | Parker | tom@email.com | 011-13579 | Bldg1
      | Jim | Walker | jim@email.com | 022-24680 | Bldg2
      | Harry | Roberts | har@email.com | 033-13579 | Bld
    Then user registration should be successful
```

Notice the console below and see that entire first map (index 0) is getting printed

```
Console
<terminated> UserRegistration.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (06-Mar-2021, 4:03:44 PM)

Scenario: User registration with different data having columns # src/test/java/Features/UserRegistration.feature:11
User is on registration page
  Given User is on registration page                          # StepDefinitions.UserRegStepDef.user_is_on_registration_page()
[{FirstName=Tom, LastName=Parker, Email=tom@email.com, Phne=011-13579, Address=Bldg1, Zip=12345, City=Delhi}, {FirstName=Jim, LastNa
{FirstName=Tom, LastName=Parker, Email=tom@email.com, Phne=011-13579, Address=Bldg1, Zip=12345, City=Delhi}
  When user enters following user details having columns       # StepDefinitions.UserRegStepDef.user_enters_following_user_details_h
```

Next, how to get a specific value from the first map?

See below. We can use the get(key) method



So we have

```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
List<Map<String, String>> userList = dataTable.asMaps(String.class, String.class);

System.out.println(userList);
System.out.println(userList.get(0).get(key));
}
```

Let us now provide a specific key to get its value, see below. We will now try to get the value of key 'FirstName' from the 0th row or 1st map

```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
List<Map<String, String>> userList = dataTable.asMaps(String.class, String.class);

System.out.println(userList);
System.out.println(userList.get(0).get("FirstName"));
}
```

Save the file

Remember that, the keys are column headers that we have defined in our feature file, see line#14 below

```
UserRegistration.feature   UserRegStepDef.java   RunnerTest.java
1  Feature: User Registration
2
3    Scenario: User registration with different data
4      Given User is on registration page
5      When user enters following user details
6        | Tom | Parker | tom@email.com | 011-13579 | Bldg1 | 12345 | Delhi |
7        | Jim | Walker | jim@email.com | 022-24680 | Bldg2 | 67890 | New York |
8        | Harry | Roberts | har@email.com | 033-13579 | Bldg3 | 13131 | Mumbai |
9      Then user registration should be successful
10
11     Scenario: User registration with different data having columns
12     Given User is on registration page
13     When user enters following user details having columns
14       | FirstName | LastName | Email | Phne | Address | Zip | City |
15       | Tom | Parker | tom@email.com | 011-13579 | Bldg1 | 12345 | Delhi |
16       | Jim | Walker | jim@email.com | 022-24680 | Bldg2 | 67890 | New York |
17       | Harry | Roberts | har@email.com | 033-13579 | Bldg3 | 13131 | Mumbai |
18     Then user registration should be successful
```

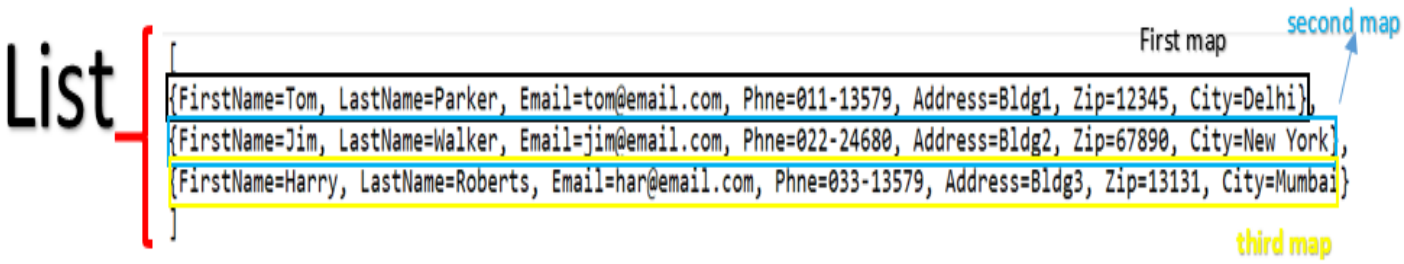
Let us now re-run the feature file

```
UserRegistration.feature   UserRegStepDef.java   RunnerTest.java
1  Feature: User Registration
2
3    Scenario: User registration with different data
4      Given User is on registration page
5      When user enters following user details
6        | Tom | Parker | tom@email.com | 011-13579 | Bldg1
7        | Jim | Walker | jim@email.com | 022-24680 | Bldg2
8        | Harry | Roberts | har@email.com | 033-13579 | Bld
9      Then user registration should be successful
10
11     Scenario: User registration with different data havin
12     Given User is on registration page
13     When user enters following user details having column
14       | FirstName | LastName | Email | Phne | Address | Z
15       | Tom | Parker | tom@email.com | 011-13579 | Bldg1
16       | Jim | Walker | jim@email.com | 022-24680 | Bldg2
17       | Harry | Roberts | har@email.com | 033-13579 | Bld
18     Then user registration should be successful

Undo Typing                Ctrl+Z
Revert File
Save                       Ctrl+S
Open With                  >
Show In                    Alt+Shift+W >
Cut                        Ctrl+X
Copy                       Ctrl+C
Paste                      Ctrl+V
Quick Fix                  Ctrl+1
Shift Right
Shift Left
Pretty Format              Ctrl+Shift+F
Find Step                  F3
Recalculate steps          F5
Coverage As                >
Run As                     >   1 Cucumber Feature
```

Notice the console below and see that the value ‘Tom’ gets printed

```
Console
<terminated> UserRegistration.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (06-Mar-2021, 4:17:32 PM)

Scenario: User registration with different data having columns # src/test/java/Features/UserRegistration.feature:11
User is on registration page
  Given User is on registration page                          # StepDefinitions.UserRegStepDef.user_is_on_registration_page(
[{FirstName=Tom, LastName=Parker, Email=tom@email.com, Phne=011-13579, Address=Bldg1, Zip=12345, City=Delhi}, {FirstName=Jim,
Tom
  When user enters following user details having columns       # StepDefinitions.UserRegStepDef.user_enters_following_user_de
User registration successful
  Then user registration should be successful                 # StepDefinitions.UserRegStepDef.user_registration_should_be_s
```

This value is correct as seen in below



Similarly you can print the values of other keys. These keys behave like columns for us.

Comment the 2 sops

```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
List<Map<String, String>> userList = dataTable.asMaps(String.class, String.class);

//System.out.println(userList);
//System.out.println(userList.get(0).get("FirstName"));
}
```

Next, we will create a 'for' loop so that we can traverse this particular userList.

So, we want to traverse through a 'Map of string' viz Map<String, String>

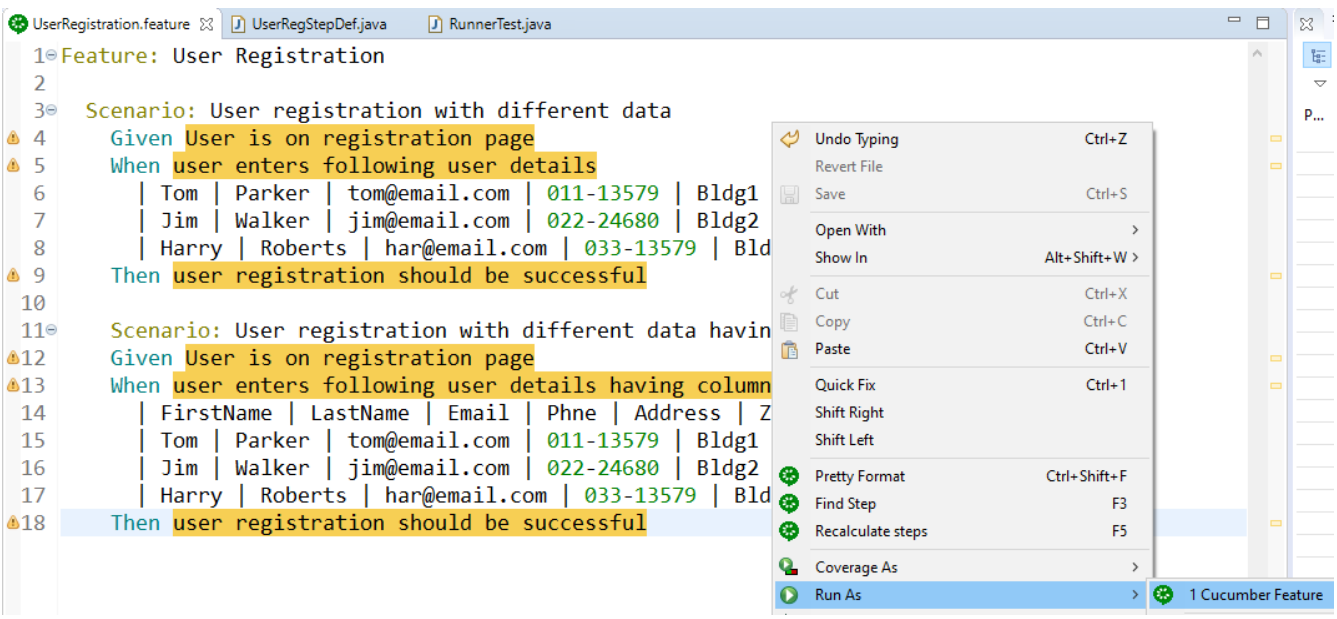
```
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
List<Map<String, String>> userList = dataTable.asMaps(String.class, String.class);

//System.out.println(userList);
//System.out.println(userList.get(0).get("FirstName"));
for(Map<String, String> m : userList) {

    }
```

We will now print the values of all the keys from all the 3 maps, see below

```java
@When("user enters following user details having columns")
public void user_enters_following_user_details_having_columns(DataTable dataTable) {
List<Map<String, String>> userList = dataTable.asMaps(String.class, String.class);

//System.out.println(userList);
//System.out.println(userList.get(0).get("FirstName"));
for(Map<String, String> m : userList) {
    System.out.println(m.get("FirstName"));
    System.out.println(m.get("LastName"));
    System.out.println(m.get("Email"));
    System.out.println(m.get("Phne"));
    System.out.println(m.get("Address"));
    System.out.println(m.get("Zip"));
    System.out.println(m.get("City"));
    }
```

Save the file

Let us now re-run the feature file



Notice the console below and see that all the values from all the maps get printed

```
Scenario: User registration with different data having columns
User is on registration page
  Given User is on registration page
Tom
Parker
tom@email.com
011-13579
Bldg1
12345
Delhi
Jim
Walker
jim@email.com
022-24680
Bldg2
67890
New York
Harry
Roberts
har@email.com
033-13579
Bldg3
13131
Mumbai
```

So this is how we can use DataTable asMaps feature.

Thank you for reading!