Way2Automation - Tutorial 8 – DataTable (asLists) in Cucumber

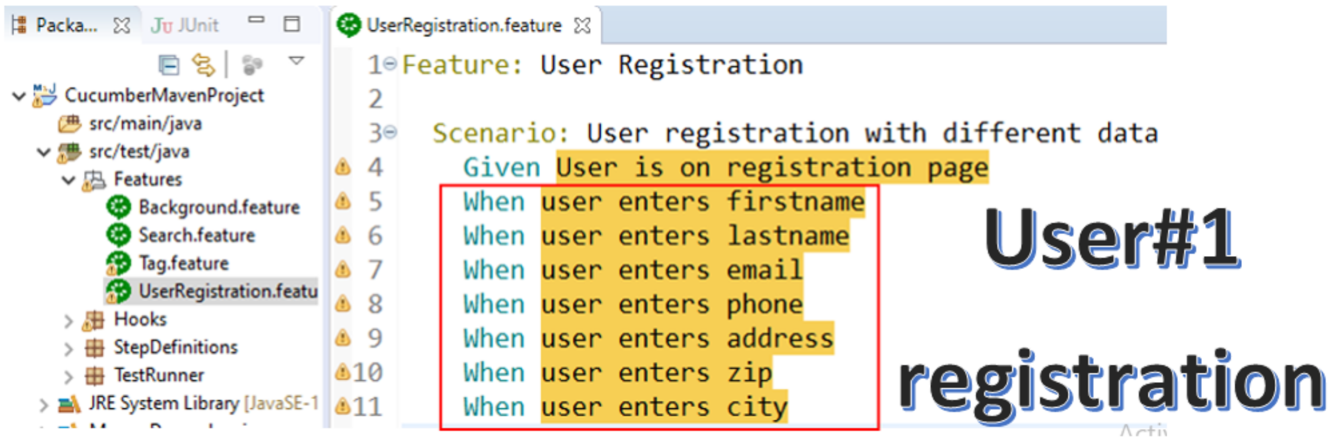
What you will Learn :

- o DataTable concept
- o DataTable asLists

DataTable concept

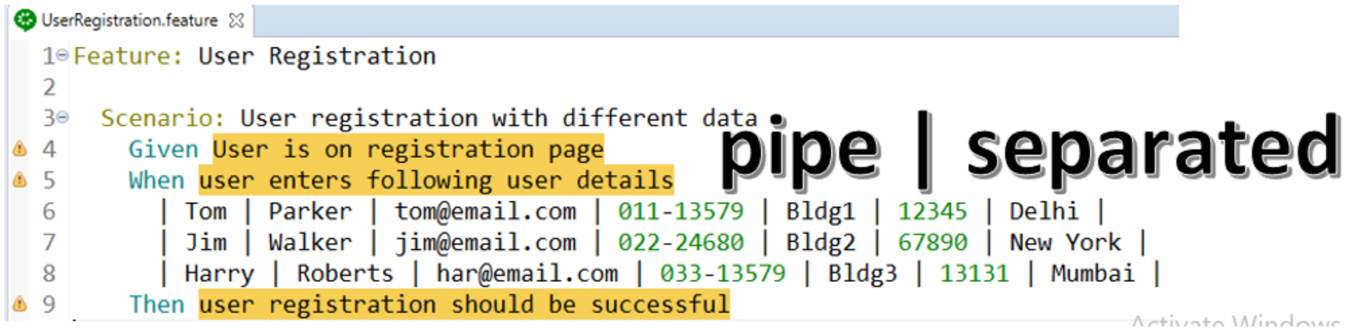
We will now study about Data Table concept in Cucumber BDD. We can parameterize a particular step in our feature file. For example, I might want to do a couple of user registrations with multiple users. Now, to register a user, we might need to enter the user's firstname, lastname, etc.

Consider the below block to register a single user. It would become cumbersome if we repeat the same block again and again to register all the users.



```
Feature: User Registration

  Scenario: User registration with different data
    Given User is on registration page
    When user enters firstname
    When user enters lastname
    When user enters email
    When user enters phone
    When user enters address
    When user enters zip
    When user enters city
```

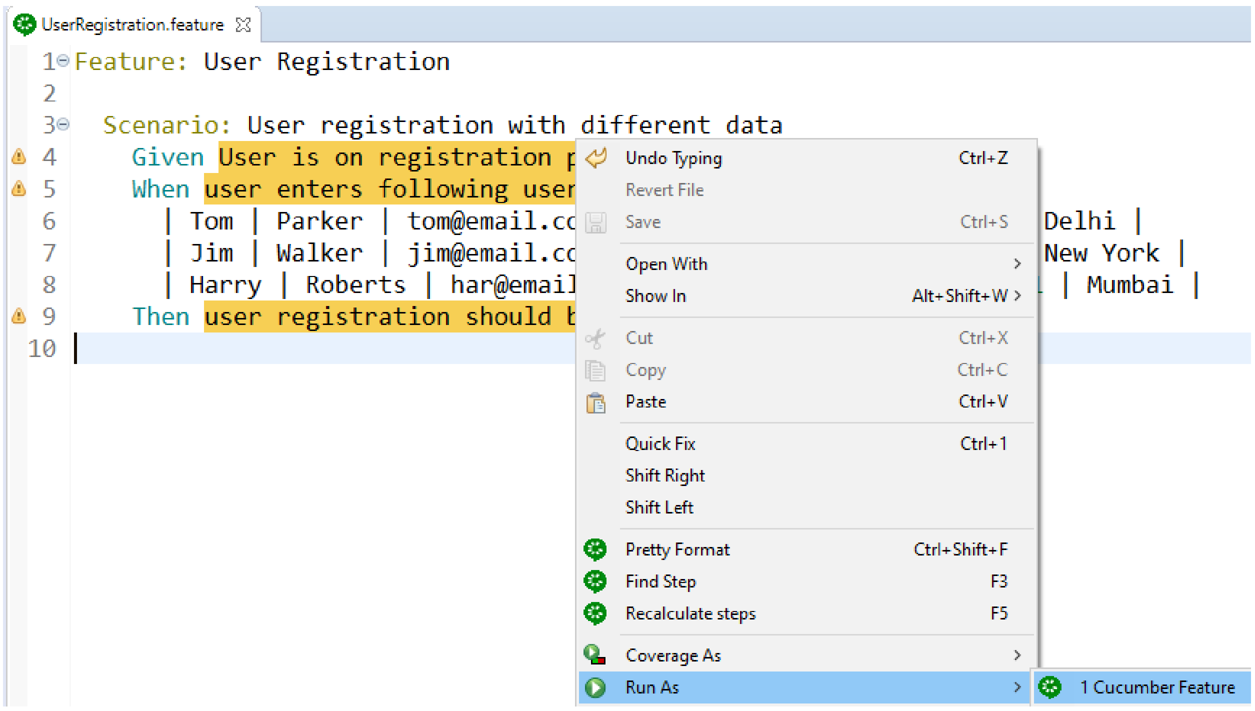
To make this simpler, we will use the DataTable feature in cucumber and write as follows. We write the pipe separated user data and notice that the feature file is more readable now. The file contains the data of 3 users.



```
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 | 12345 | Delhi |
      | Jim | Walker | jim@email.com | 022-24680 | Bldg2 | 67890 | New York |
      | Harry | Roberts | har@email.com | 033-13579 | Bldg3 | 13131 | Mumbai |
    Then user registration should be successful
```

Save the file

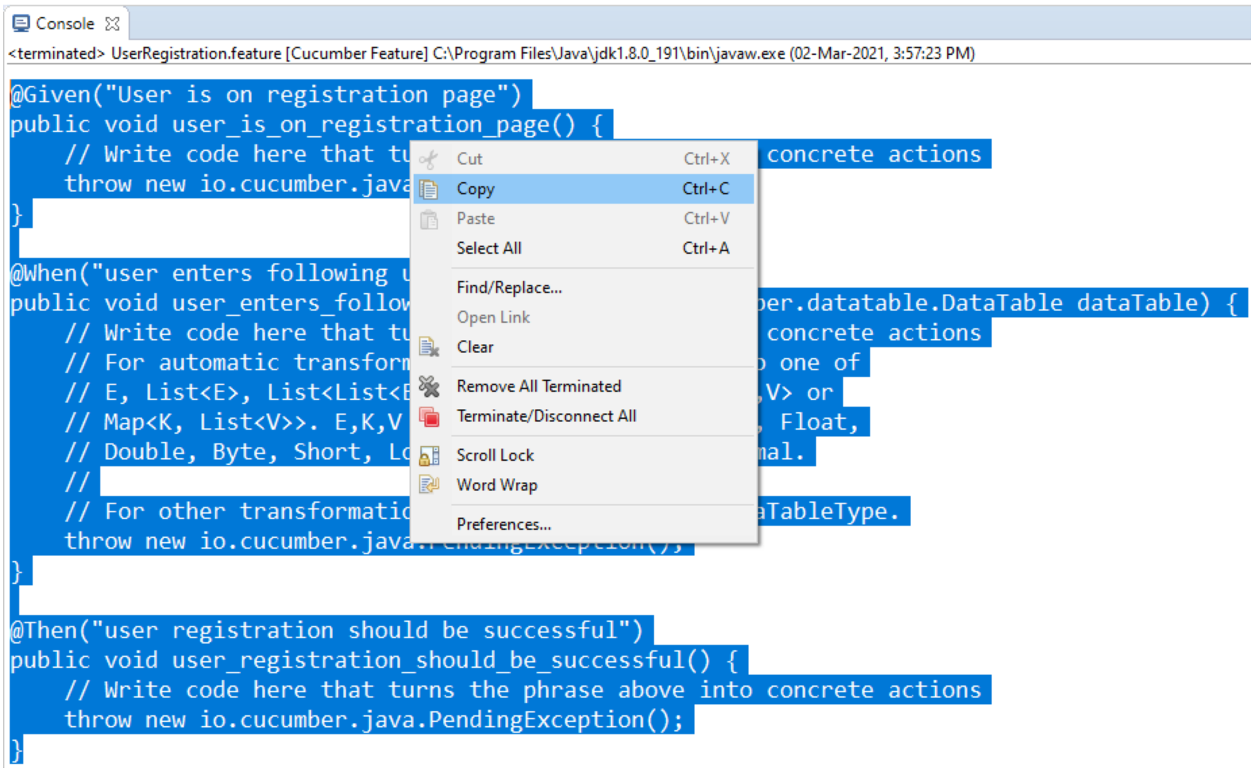
Let us now run this file to generate the step definition methods



UserRegistration.feature

```
Feature: User Registration

  Scenario: User registration with different data
    Given User is on registration p
    When user enters following user
      | Tom | Parker | tom@email.co                Delhi |
      | Jim | Walker | jim@email.co                New York |
      | Harry | Roberts | har@email                | Mumbai |
    Then user registration should b
```

Copy the code snippets

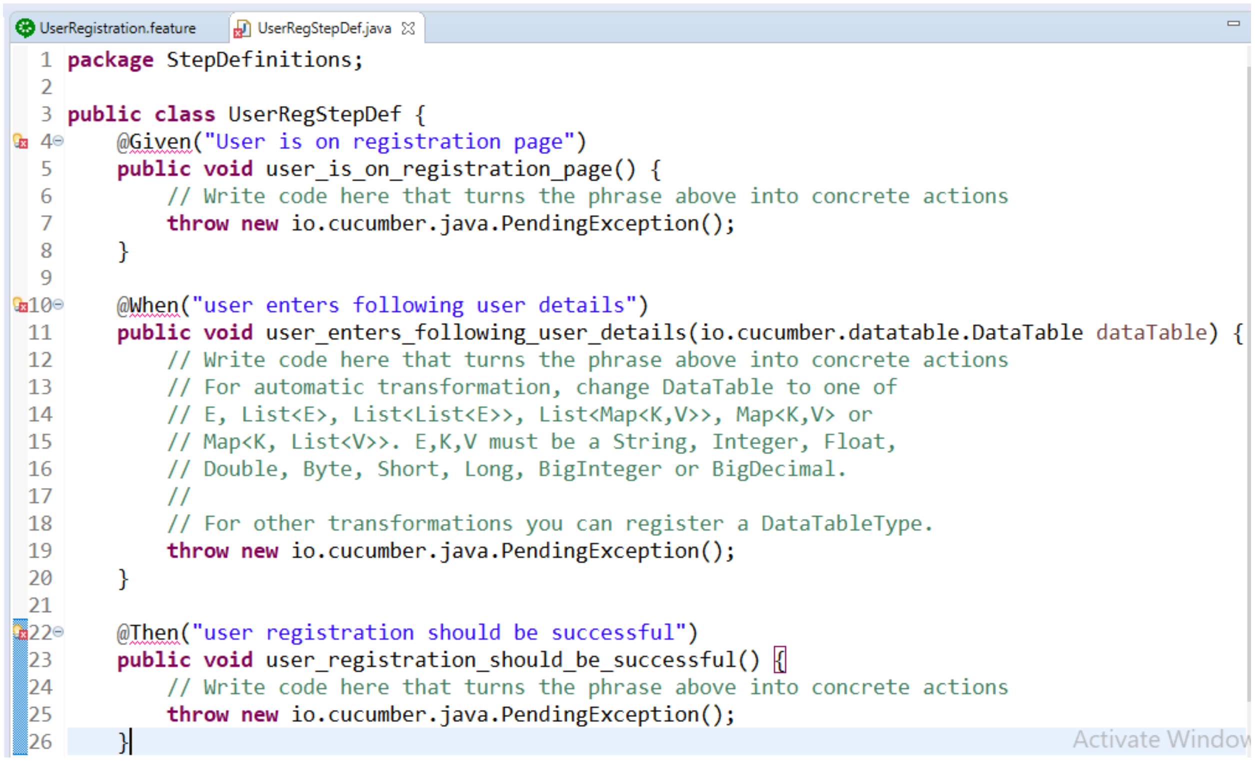


Console

```
<terminated> UserRegistration.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (02-Mar-2021, 3:57:23 PM)
```

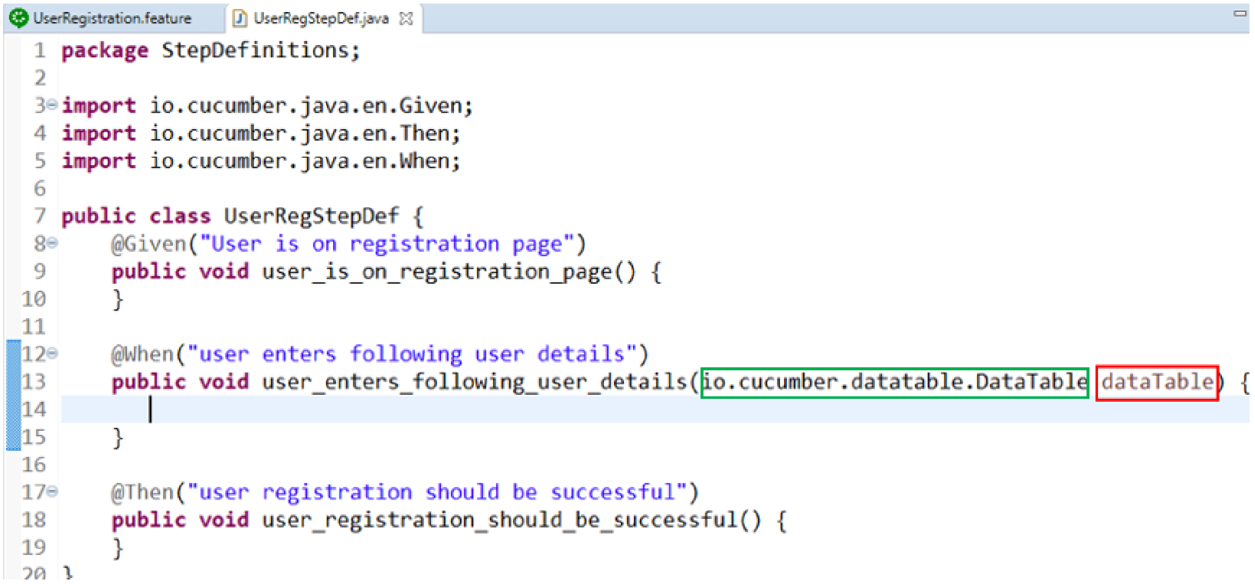
```
@Given("User is on registration page")
public void user_is_on_registration_page() {
    // Write code here that tu                          concrete actions
    throw new io.cucumber.java
}

@When("user enters following u
public void user_enters_follow                          per.datatable.DataTable dataTable) {
    // Write code here that tu                          concrete actions
    // For automatic transform                          o one of
    // E, List<E>, List<List<E                          ,V> or
    // Map<K, List<V>>. E,K,V                           , Float,
    // Double, Byte, Short, Lo                          mal.
    //
    // For other transformatic                          aTableType.
    throw new io.cucumber.java.PendingException();
}

@Then("user registration should be successful")
public void user_registration_should_be_successful() {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}
```

Paste them in a step definition file



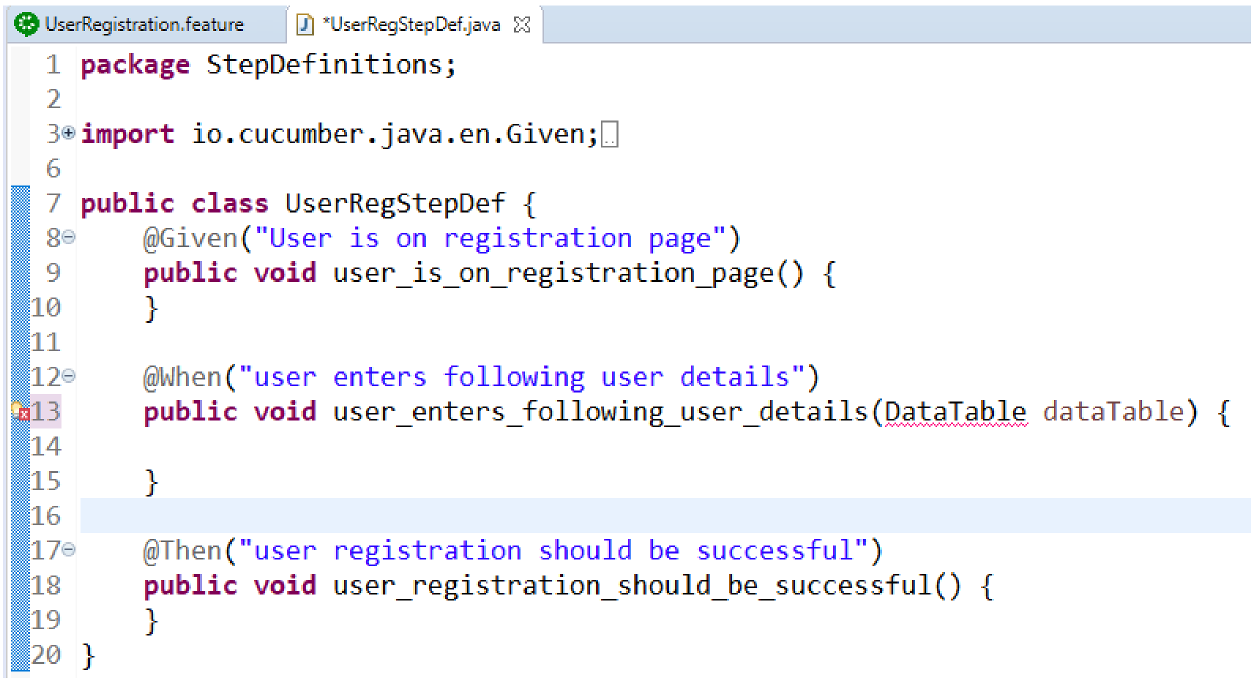
```java
package StepDefinitions;

public class UserRegStepDef {
    @Given("User is on registration page")
    public void user_is_on_registration_page() {
        // Write code here that turns the phrase above into concrete actions
        throw new io.cucumber.java.PendingException();
    }

    @When("user enters following user details")
    public void user_enters_following_user_details(io.cucumber.datatable.DataTable dataTable) {
        // Write code here that turns the phrase above into concrete actions
        // For automatic transformation, change DataTable to one of
        // E, List<E>, List<List<E>>, List<Map<K,V>>, Map<K,V> or
        // Map<K, List<V>>. E,K,V must be a String, Integer, Float,
        // Double, Byte, Short, Long, BigInteger or BigDecimal.
        //
        // For other transformations you can register a DataTableType.
        throw new io.cucumber.java.PendingException();
    }

    @Then("user registration should be successful")
    public void user_registration_should_be_successful() {
        // Write code here that turns the phrase above into concrete actions
        throw new io.cucumber.java.PendingException();
    }
```

Import the missing packages and remove the comments plus exceptions. If you notice, for the second method, the data is coming in the form of dataTable. Also, it is coming from library **io.cucumber.datatable.DataTable**



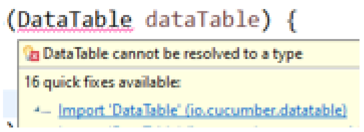
```java
package StepDefinitions;

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() {
    }

    @When("user enters following user details")
    public void user_enters_following_user_details(io.cucumber.datatable.DataTable dataTable) {
        
    }

    @Then("user registration should be successful")
    public void user_registration_should_be_successful() {
    }
}
```

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



```
package StepDefinitions;

import io.cucumber.java.en.Given;

public class UserRegStepDef {
    @Given("User is on registration page")
    public void user_is_on_registration_page() {
    }

    @When("user enters following user details")
    public void user_enters_following_user_details(DataTable dataTable) {

    }

    @Then("user registration should be successful")
    public void user_registration_should_be_successful() {
    }
}
```

However you have to import the same package **io.cucumber.datatable**



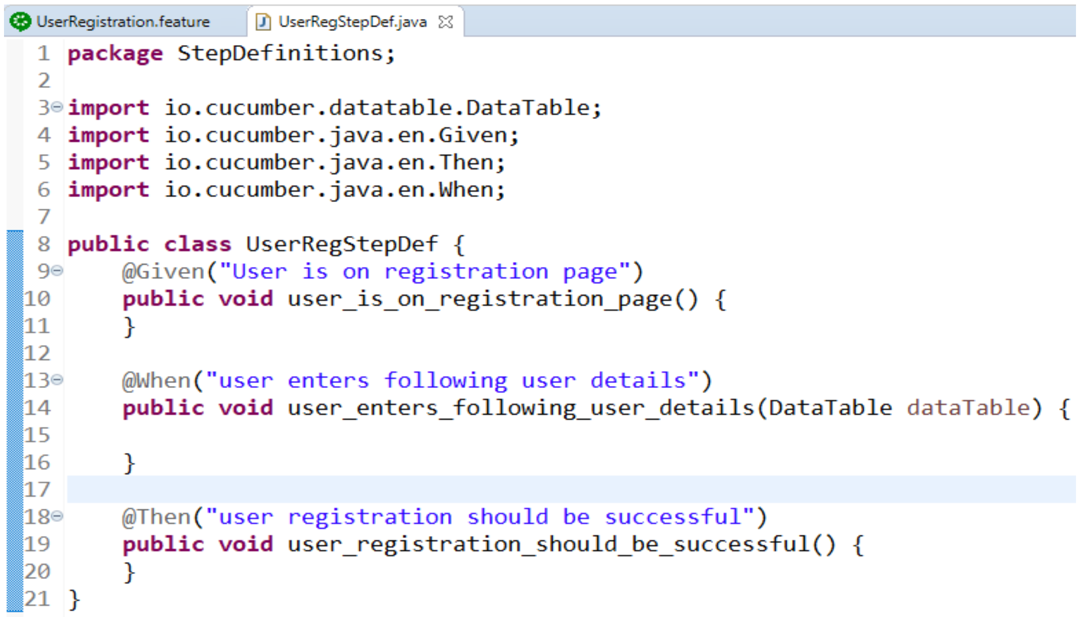
```
(DataTable dataTable) {
```

DataTable cannot be resolved to a type

16 quick fixes available:

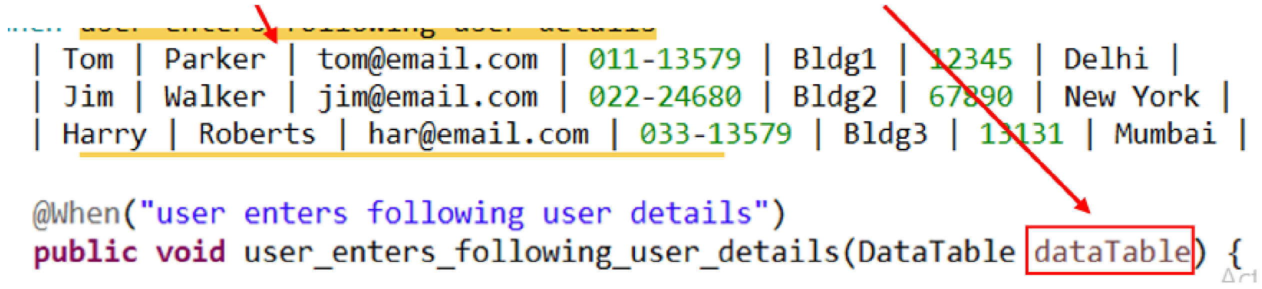
Import 'DataTable' (io.cucumber.datatable)

So let us do that



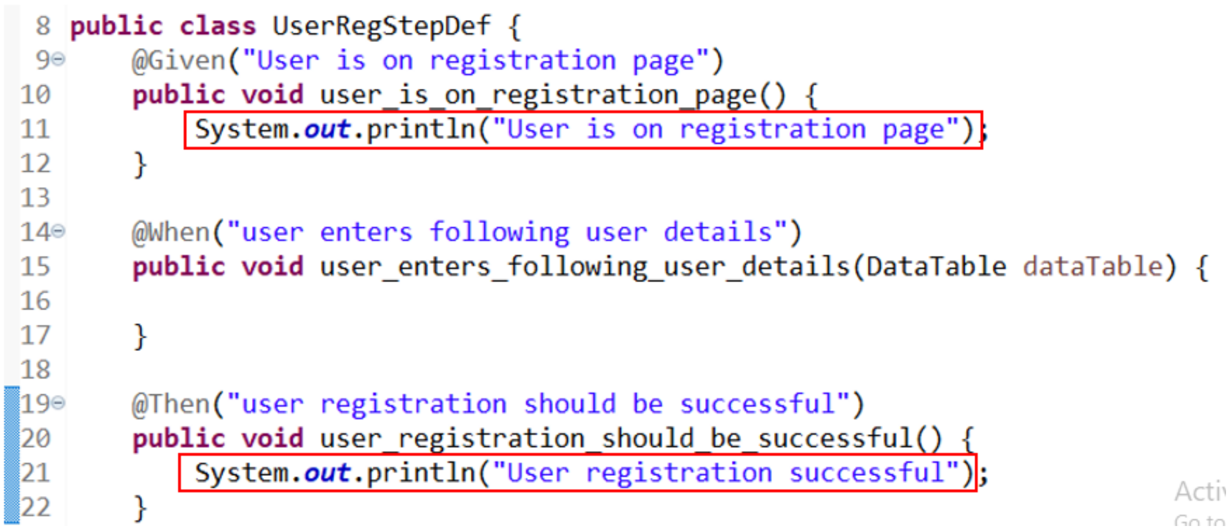
```
package StepDefinitions;

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() {
    }

    @When("user enters following user details")
    public void user_enters_following_user_details(DataTable dataTable) {

    }

    @Then("user registration should be successful")
    public void user_registration_should_be_successful() {
    }
}
```

So when the 2nd step @When will be executed, the multiple data sets that you have given in the feature file will be coming to this particular dataTable



```
| 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")
public void user_enters_following_user_details(DataTable dataTable) {
```

Next, let us write simple SOPs for first and third step

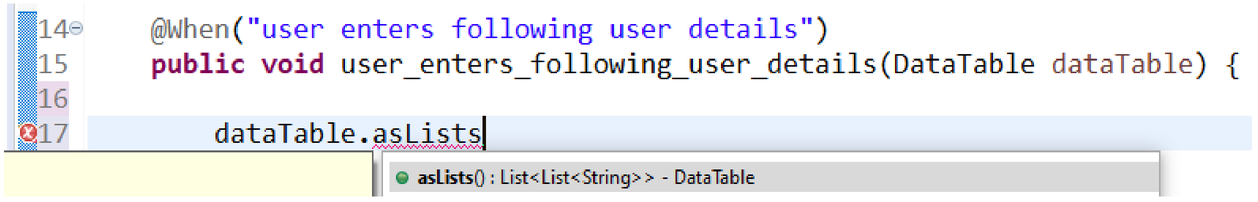


```
public class UserRegStepDef {
    @Given("User is on registration page")
    public void user_is_on_registration_page() {
        System.out.println("User is on registration page");
    }

    @When("user enters following user details")
    public void user_enters_following_user_details(DataTable dataTable) {

    }

    @Then("user registration should be successful")
    public void user_registration_should_be_successful() {
        System.out.println("User registration successful");
    }
```

Next, let us now concentrate on second step @When, see below

<u>DataTable asLists</u>

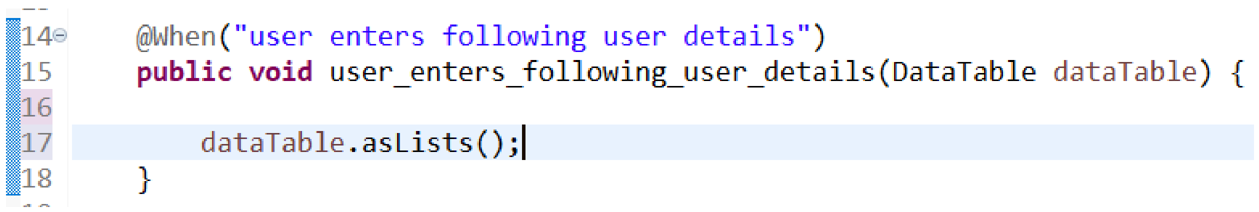
As you can see below, there is a method **asLists()** available



```
    @When("user enters following user details")
    public void user_enters_following_user_details(DataTable dataTable) {

        dataTable.asLists
```

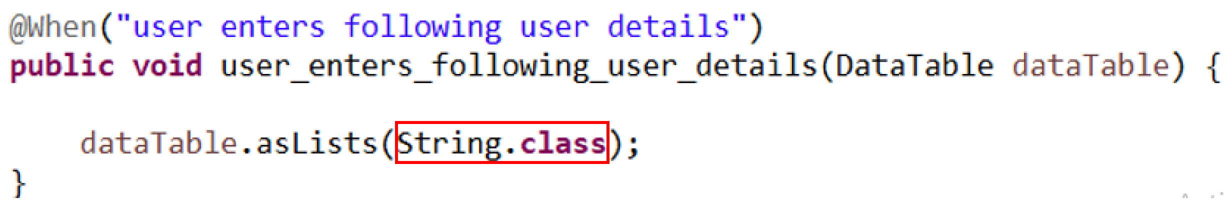
asLists() : List<List<String>> - DataTable

Let us use that method



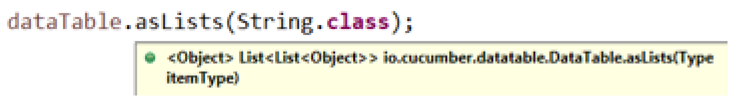
```
    @When("user enters following user details")
    public void user_enters_following_user_details(DataTable dataTable) {

        dataTable.asLists();
    }
```

Now, what kind of list you want to get? We want to get ‘String’ type of list, because the dataset in the feature file is in string format. So we will simply write String.class



```
@When("user enters following user details")
public void user_enters_following_user_details(DataTable dataTable) {

    dataTable.asLists(String.class);
}
```

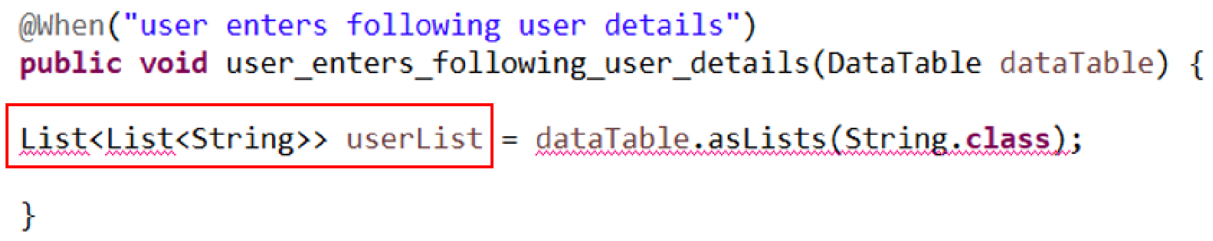
Now, as you can see below, this asLists() method returns “List of List of object”. Again, this list will again be of type ‘String’



```
dataTable.asLists(String.class);
```

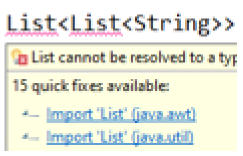
<Object> List<List<Object>> io.cucumber.datatable.DataTable.asLists(Type itemType)

So we can say



```
@When("user enters following user details")
public void user_enters_following_user_details(DataTable dataTable) {

List<List<String>> userList = dataTable.asLists(String.class);

}
```

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



```
List<List<String>>
```

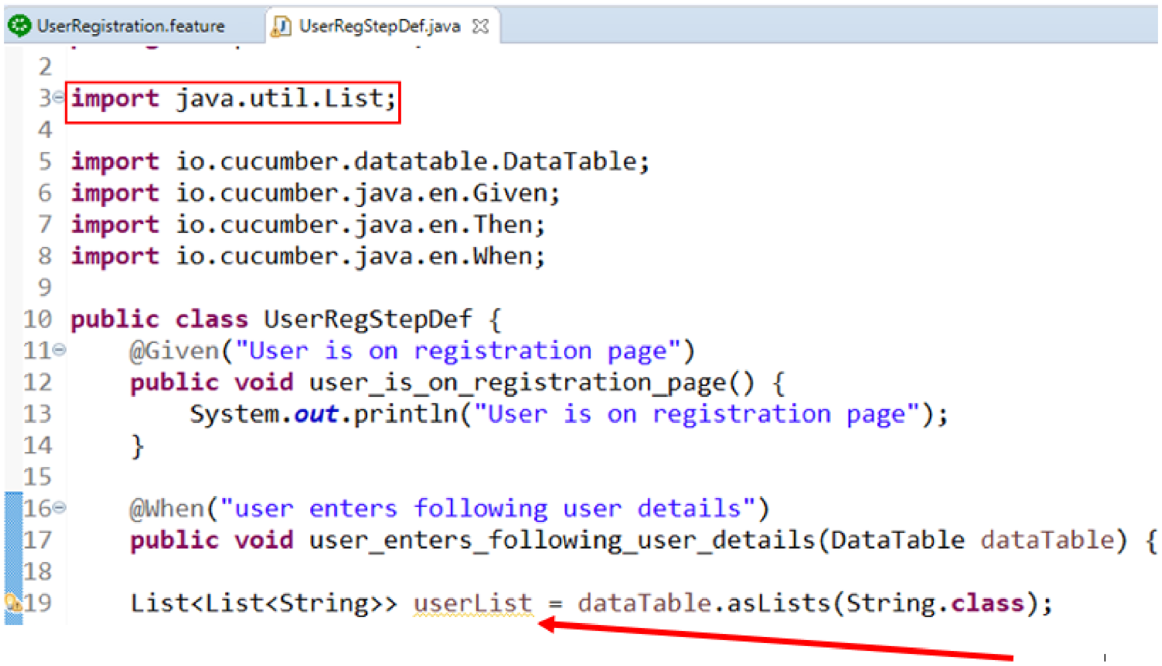
List cannot be resolved to a typ

15 quick fixes available:

Import 'List' (java.awt)

Import 'List' (java.util)

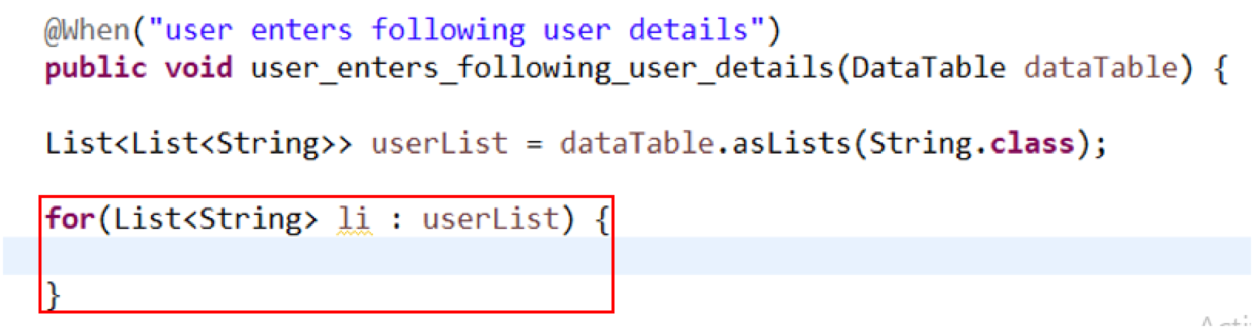
So we have



```java

import java.util.List;

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);
```

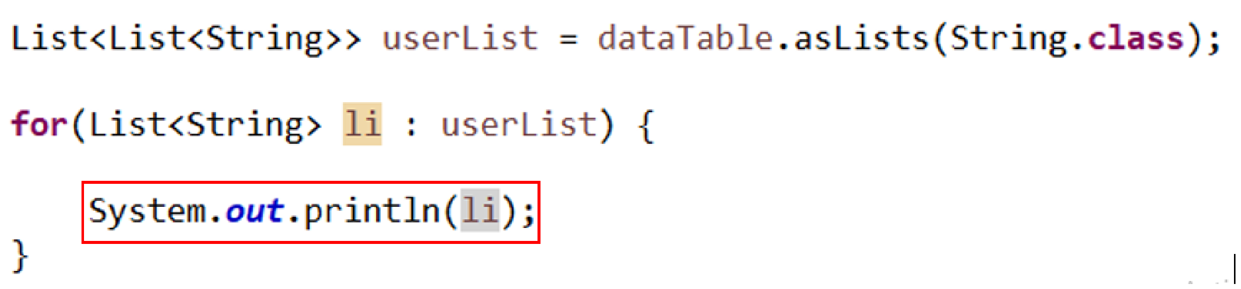
Next, we will create a ‘for’ loop so that we can traverse this particular userList.

So, we want to traverse through a ‘List of string’ viz List<String>



```java
@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) {

}
```

We will next print the value of ‘li’



```java
List<List<String>> userList = dataTable.asLists(String.class);

for(List<String> li : userList) {

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

It means, print each and every list. So in the first iteration, we will be getting the first list viz first set of data

```
| Tom | Parker | tom@email.com | 011-13579 | Bldg1 | 12345 | Delhi |
```

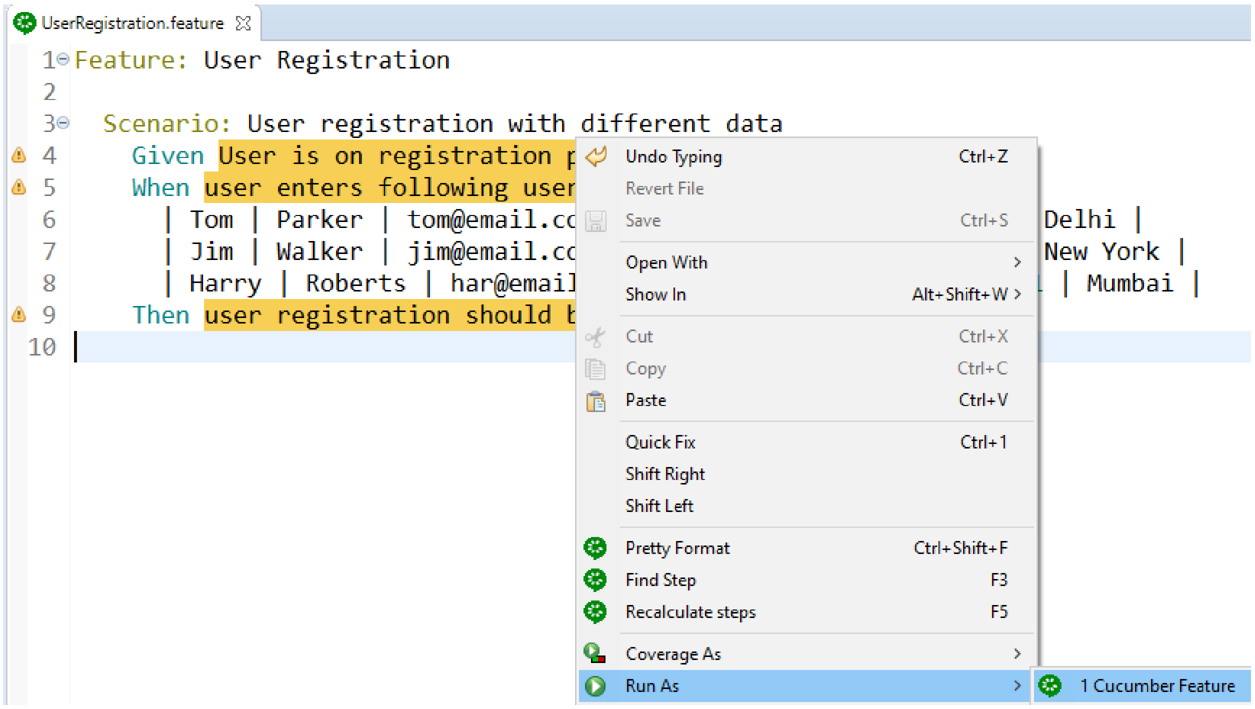
In the second iteration, we will get

```
| Jim | Walker | jim@email.com | 022-24680 | Bldg2 | 67890 | New York |
```

And so on…

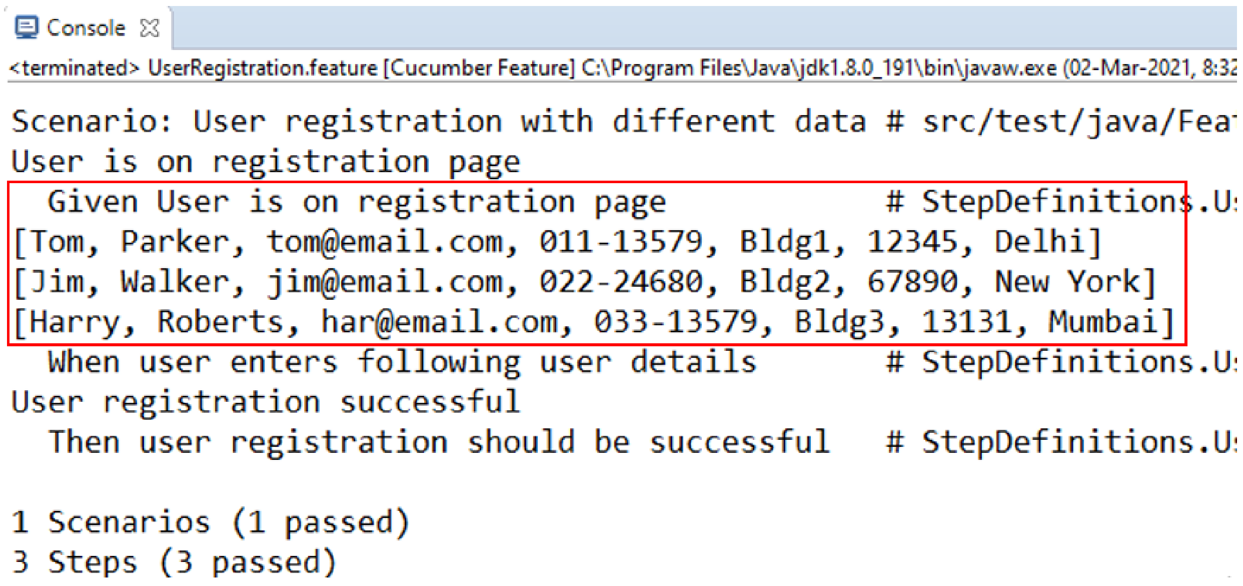
Save the step definition file.

Let us now re-run our feature file



```
Feature: User Registration

  Scenario: User registration with different data
    Given User is on registration p
    When user enters following user
      | Tom | Parker | tom@email.cc                          Delhi |
      | Jim | Walker | jim@email.cc                          New York |
      | Harry | Roberts | har@email                          | Mumbai |
    Then user registration should b
```

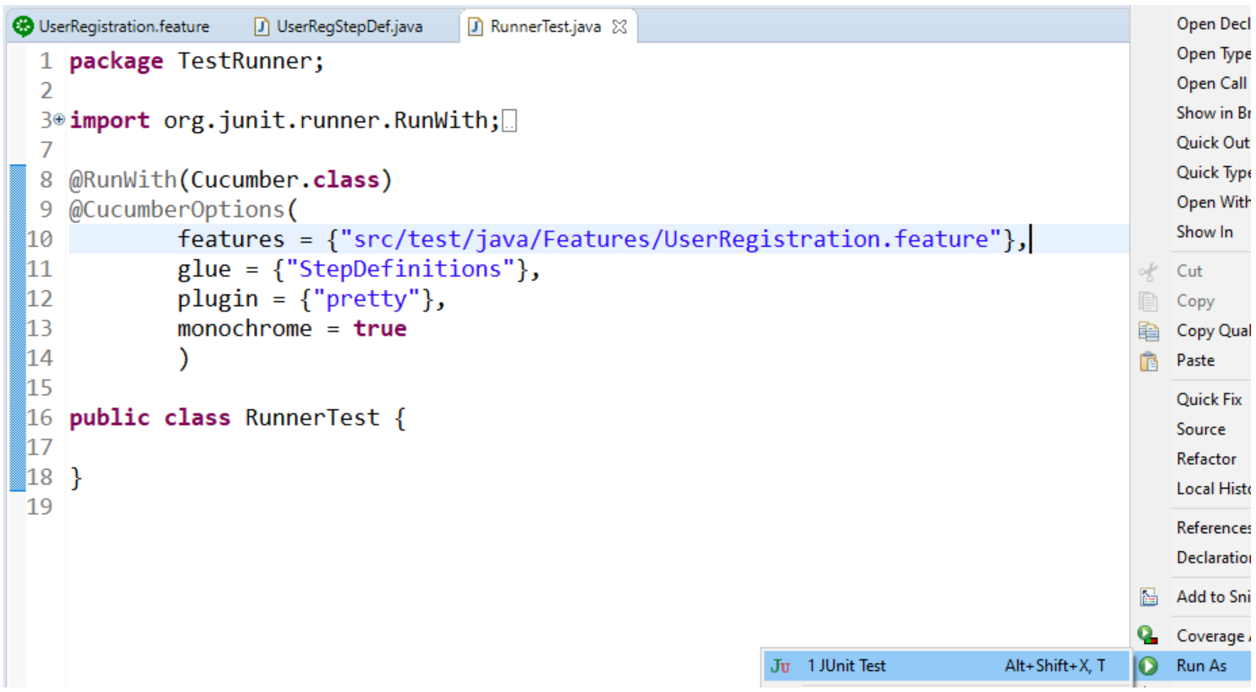
See the console. When the user is on registration page, we are entering the 3 data set of values



```
Console ⊠
<terminated> UserRegistration.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (02-Mar-2021, 8:32
Scenario: User registration with different data # src/test/java/Fea
User is on registration page
  Given User is on registration page          # StepDefinitions.U
[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.U
User registration successful
  Then user registration should be successful  # StepDefinitions.U

1 Scenarios (1 passed)
3 Steps (3 passed)
```

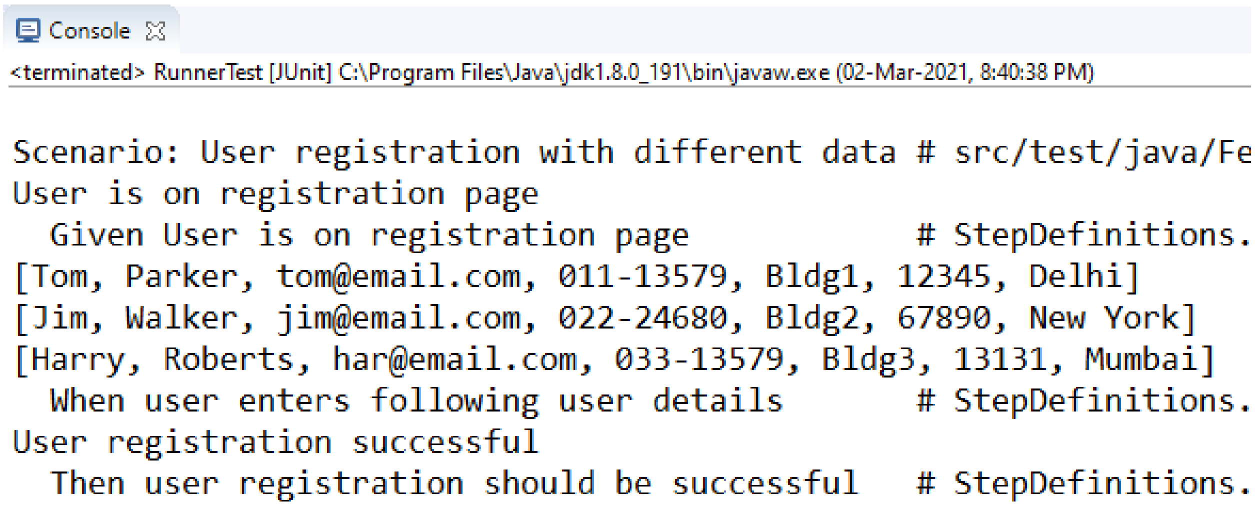
So like this way, you can use the concept of datatable in cucumber. The datatable can be applied to any type of step Given/When/Then etc..

Let us create a RunnerTest file and run it as Junit test



```java
package TestRunner;

import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(
        features = {"src/test/java/Features/UserRegistration.feature"},
        glue = {"StepDefinitions"},
        plugin = {"pretty"},
        monochrome = true
        )

public class RunnerTest {

}
```

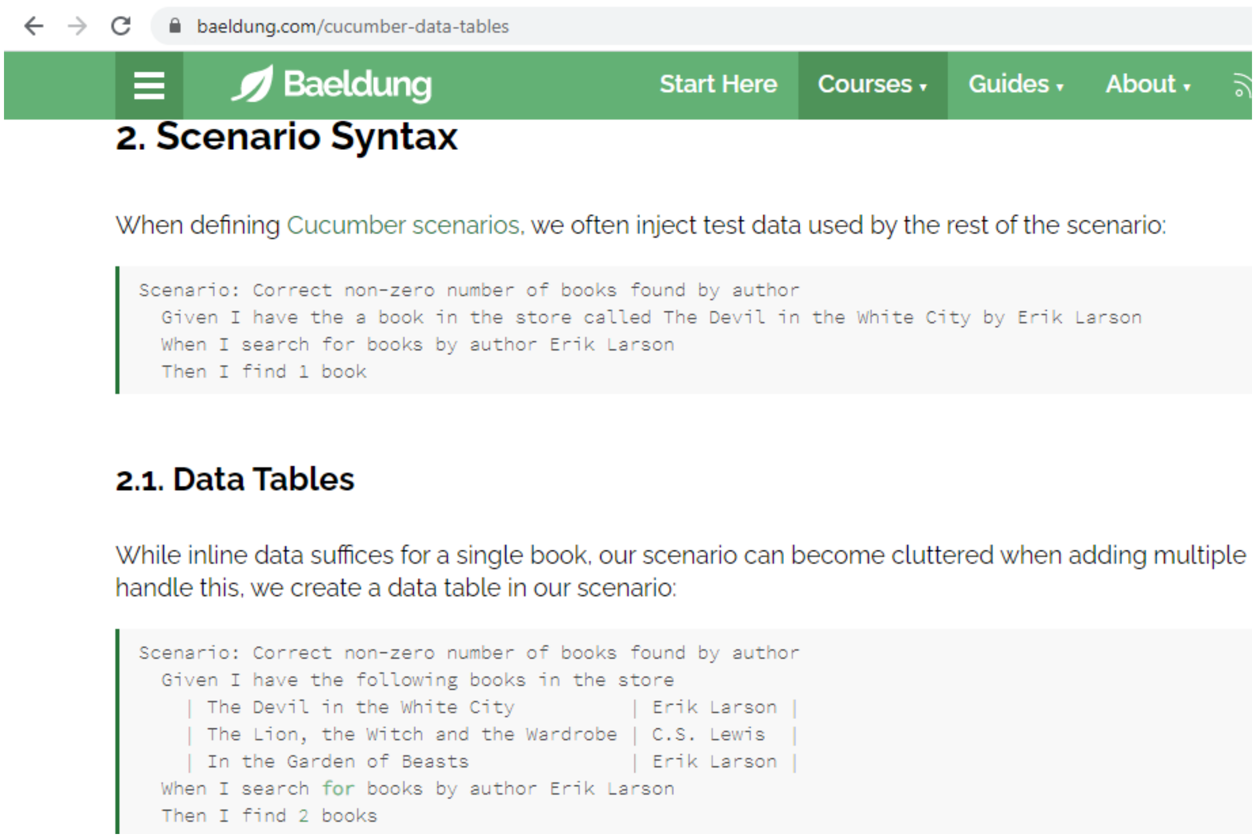
See the console o/p below. We get exactly the same thing



```
<terminated> RunnerTest [JUnit] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (02-Mar-2021, 8:40:38 PM)

Scenario: User registration with different data # src/test/java/Fe
User is on registration page
  Given User is on registration page            # StepDefinitions.
[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.
User registration successful
  Then user registration should be successful   # StepDefinitions.
```

Reference

You can further refer www.baeldung.com/cucumber-data-tables



2. Scenario Syntax

When defining Cucumber scenarios, we often inject test data used by the rest of the scenario:

```
Scenario: Correct non-zero number of books found by author
  Given I have the a book in the store called The Devil in the White City by Erik Larson
  When I search for books by author Erik Larson
  Then I find 1 book
```

2.1. Data Tables

While inline data suffices for a single book, our scenario can become cluttered when adding multiple
handle this, we create a data table in our scenario:

```
Scenario: Correct non-zero number of books found by author
  Given I have the following books in the store
    | The Devil in the White City          | Erik Larson |
    | The Lion, the Witch and the Wardrobe | C.S. Lewis  |
    | In the Garden of Beasts              | Erik Larson |
  When I search for books by author Erik Larson
  Then I find 2 books
```

Thank you for reading!