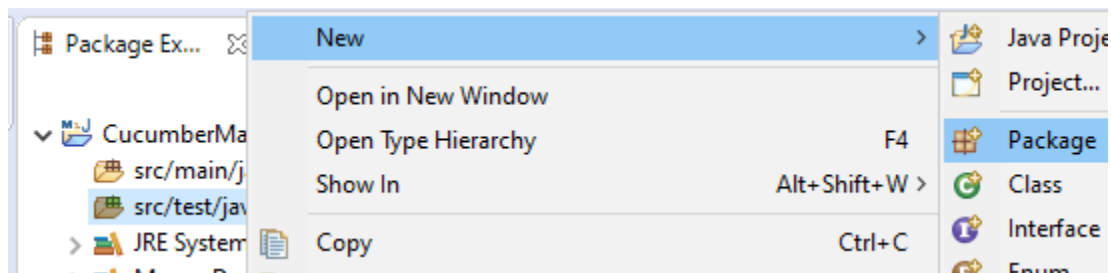# Way2Automation - Tutorial 3 – Create feature and Step definition file

What you will Learn :

- o Create a feature file
- o Configure the cucumber feature
- o Execute the feature file
- o Create Step Definition file

Create a feature file

Right click src/test/java > New > Package

Let us name the package as 'Features' (you can also name the package as 'AppFeatures' or any other name you wish to)

Finish, the package gets created
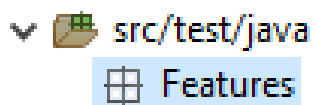
Next, right click 'Features' package > New > File
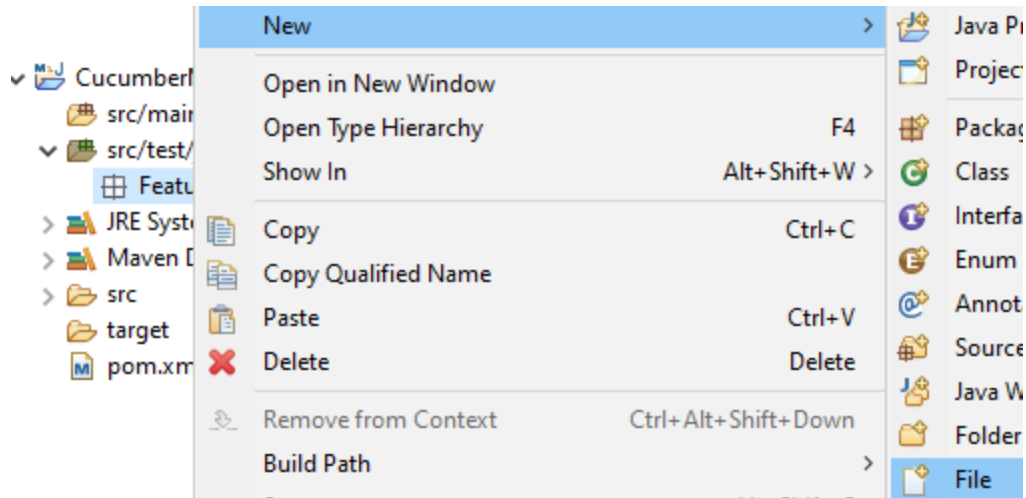
The filename should describe some feature (example: login, search, addtocart etc). So, let us mention a file name with **.feature** suffix as seen below
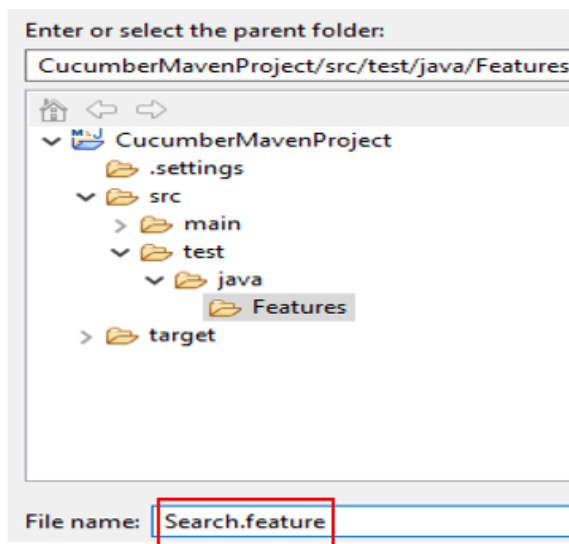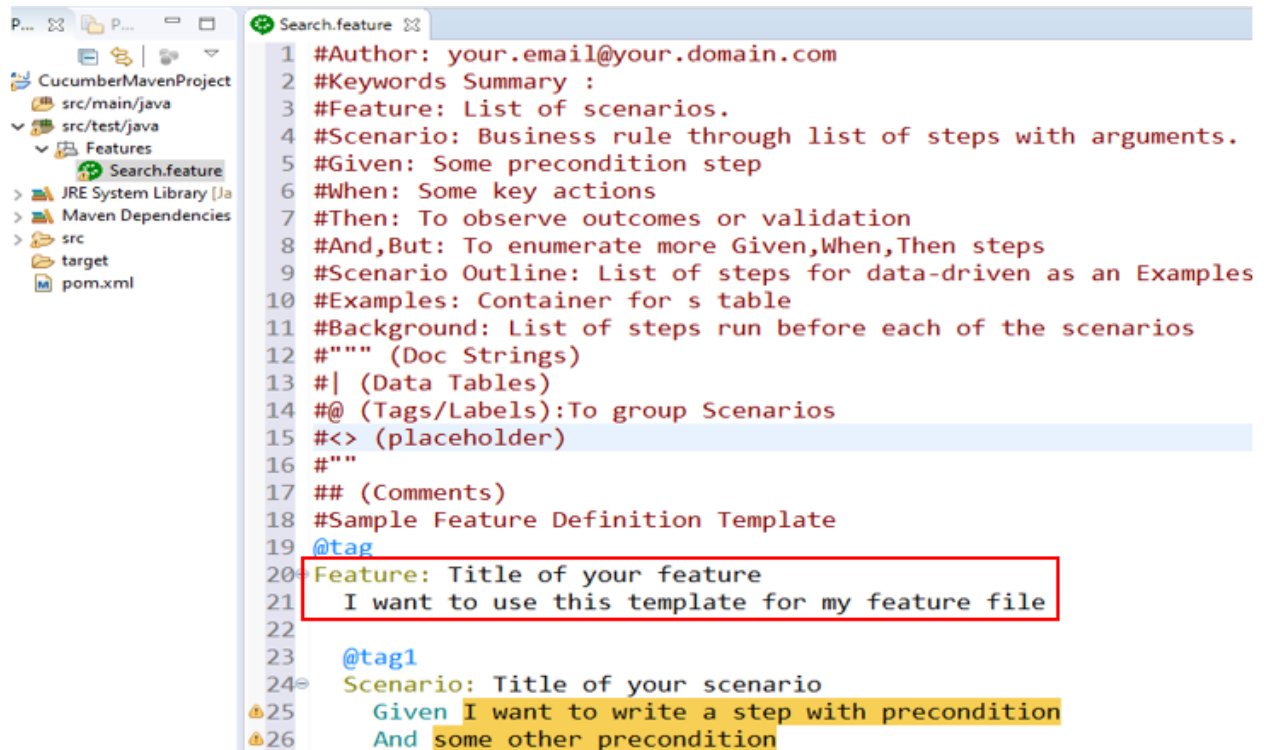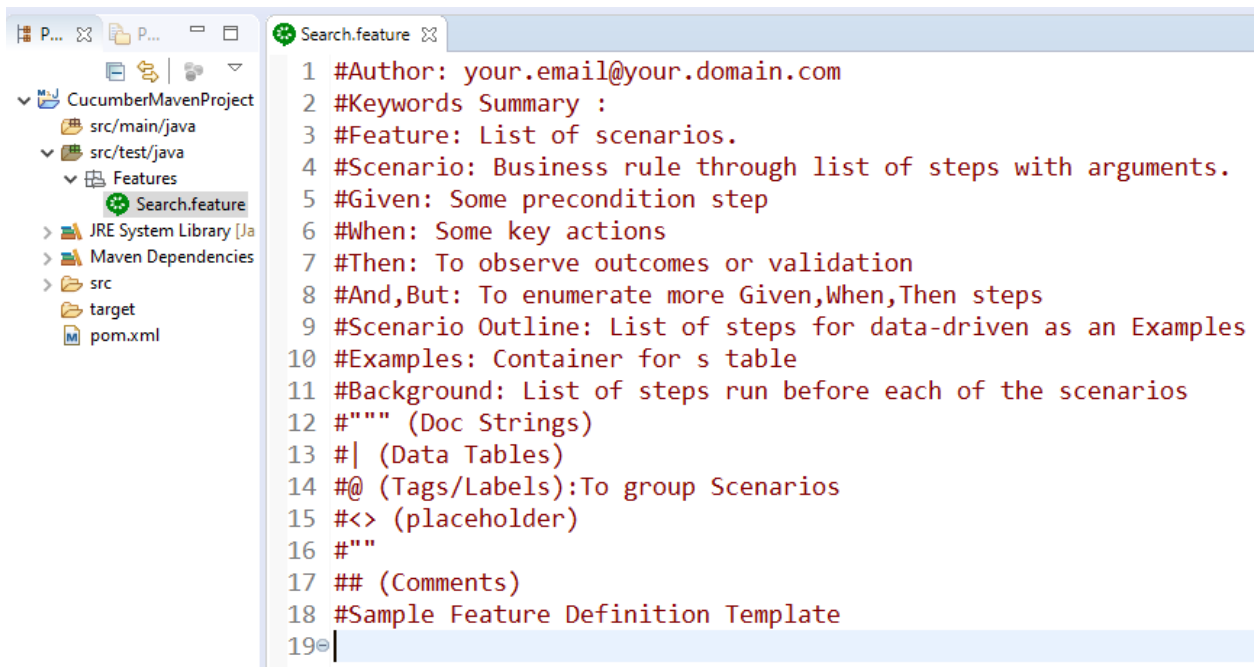

Finish

Notice below that feature file gets created with some default features. This is due to the cucumber plugin that we had installed in the previous tutorial

```
    1  #Author: your.email@your.domain.com
    2  #Keywords Summary :
    3  #Feature: List of scenarios.
    4  #Scenario: Business rule through list of steps with arguments.
    5  #Given: Some precondition step
    6  #When: Some key actions
    7  #Then: To observe outcomes or validation
    8  #And,But: To enumerate more Given,When,Then steps
    9  #Scenario Outline: List of steps for data-driven as an Examples
   10  #Examples: Container for s table
   11  #Background: List of steps run before each of the scenarios
   12  #""" (Doc Strings)
   13  #| (Data Tables)
   14  #@ (Tags/Labels):To group Scenarios
   15  #<> (placeholder)
   16  #""
   17  ## (Comments)
   18  #Sample Feature Definition Template
   19  @tag
   20  Feature: Title of your feature
   21     I want to use this template for my feature file
   22
   23     @tag1
   24     Scenario: Title of your scenario
   25        Given I want to write a step with precondition
   26        And some other precondition
```

You can also see some of the comments (lines 1-18) that describe some of the keywords for your ready reference.

Let us remove the sample features (line#19 onwards)


```
    1  #Author: your.email@your.domain.com
    2  #Keywords Summary :
    3  #Feature: List of scenarios.
    4  #Scenario: Business rule through list of steps with arguments.
    5  #Given: Some precondition step
    6  #When: Some key actions
    7  #Then: To observe outcomes or validation
    8  #And,But: To enumerate more Given,When,Then steps
    9  #Scenario Outline: List of steps for data-driven as an Examples
   10  #Examples: Container for s table
   11  #Background: List of steps run before each of the scenarios
   12  #""" (Doc Strings)
   13  #| (Data Tables)
   14  #@ (Tags/Labels):To group Scenarios
   15  #<> (placeholder)
   16  #""
   17  ## (Comments)
   18  #Sample Feature Definition Template
   19
```

We will create our own feature using the **Feature:** reserved keyword. So let us write **Feature:**

```
17  ## (Comments)
18  #Sample Feature Defir
19
20  Feature:
```

Now hit spacebar to create whitespace. Notice that the color changes

```
17  ## (Comments)
18  #Sample Feature Defi
19
20  Feature:
```

Let us give some name to our feature

```
18  #Sample Feature Definition Template
19
20  Feature: Way2Automation search
```

Let us now write scenario name using **Scenario:** keyword

```
20  Feature: Way2Automation search
21
22  Scenario: Search course
```

Let us now write the 'Given' keyword which defines some pre-condition

```
Search.feature ⊠

 5 #Given: Some precondition step
 6 #When: Some key actions
 7 #Then: To observe outcomes or validation
 8 #And,But: To enumerate more Given,When,Then steps
 9 #Scenario Outline: List of steps for data-driven as an Examples and <pl
10 #Examples: Container for s table
11 #Background: List of steps run before each of the scenarios
12 #""" (Doc Strings)
13 #| (Data Tables)
14 #@ (Tags/Labels):To group Scenarios
15 #<> (placeholder)
16 #""
17 ## (Comments)
18 #Sample Feature Definition Template
19 Feature: Way2Automation search
20
21    Scenario: Search course
22       Given Search field should be present on the Way2Automation website
```

Carefully see line#22 above. Do NOT add colon after 'Given' keyword. Thus **Given:**

is wrong. You should use **Given**

Next, we will write the 'When' step that defines some action. Since the name of the course is a string, it should be in double quotes, the price is numeric and hence without quotes

```
 6 #When: Some key actions
 7 #Then: To observe outcomes or validation
 8 #And,But: To enumerate more Given,When,Then steps
 9 #Scenario Outline: List of steps for data-driven as an Examples and <placehol
10 #Examples: Container for s table
11 #Background: List of steps run before each of the scenarios
12 #""" (Doc Strings)
13 #| (Data Tables)
14 #@ (Tags/Labels):To group Scenarios
15 #<> (placeholder)
16 #""
17 ## (Comments)
18 #Sample Feature Definition Template
19 Feature: Way2Automation search
20
21    Scenario: Search course
22       Given Search field should be present on the Way2Automation website
23       When I search for a course "cucumber BDD for Selenium" having price 1000
```

Next we will write 'Then' statement that defines an outcome

```
 7  #Then: To observe outcomes or validation
 8  #And,But: To enumerate more Given,When,Then steps
 9  #Scenario Outline: List of steps for data-driven as an Examples and <placeholder>
10  #Examples: Container for s table
11  #Background: List of steps run before each of the scenarios
12  #""" (Doc Strings)
13  #| (Data Tables)
14  #@ (Tags/Labels):To group Scenarios
15  #<> (placeholder)
16  #""
17  ## (Comments)
18  #Sample Feature Definition Template
19  Feature: Way2Automation search
20
21    Scenario: Search course
22      Given Search field should be present on the Way2Automation website
23      When I search for a course "cucumber BDD for Selenium" having price 1000
24      Then Course "Cucumber BDD for Selenium & Appium with Live Projects" should be displayed
```

So this was our first scenario. Likewise, you can write any number of scenarios for search feature.
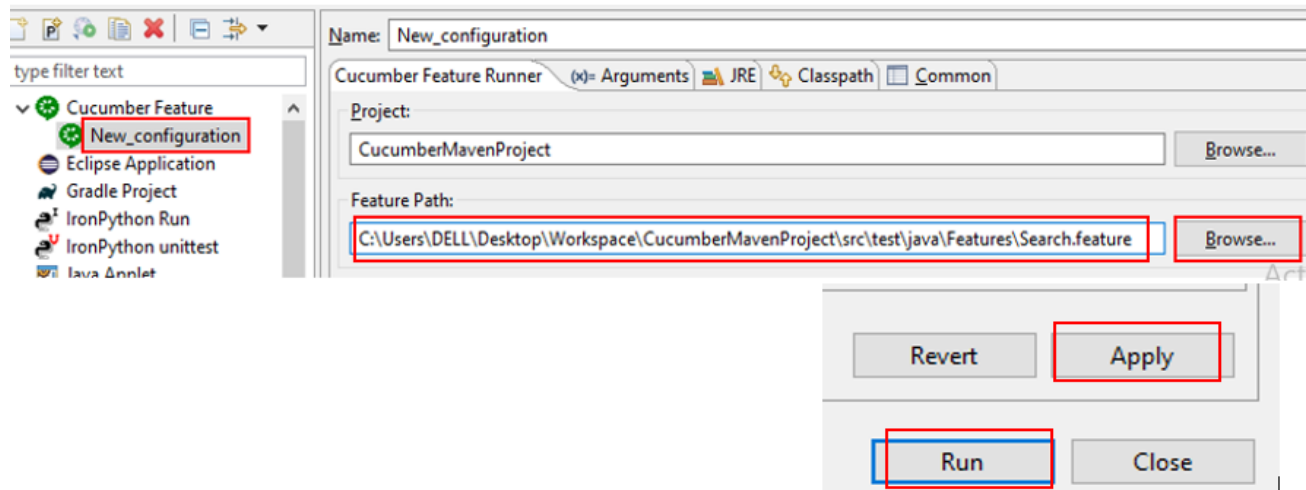
Configure the cucumber feature

Let us see how to configure cucumber feature

Right click feature file > Run As > Run Configurations
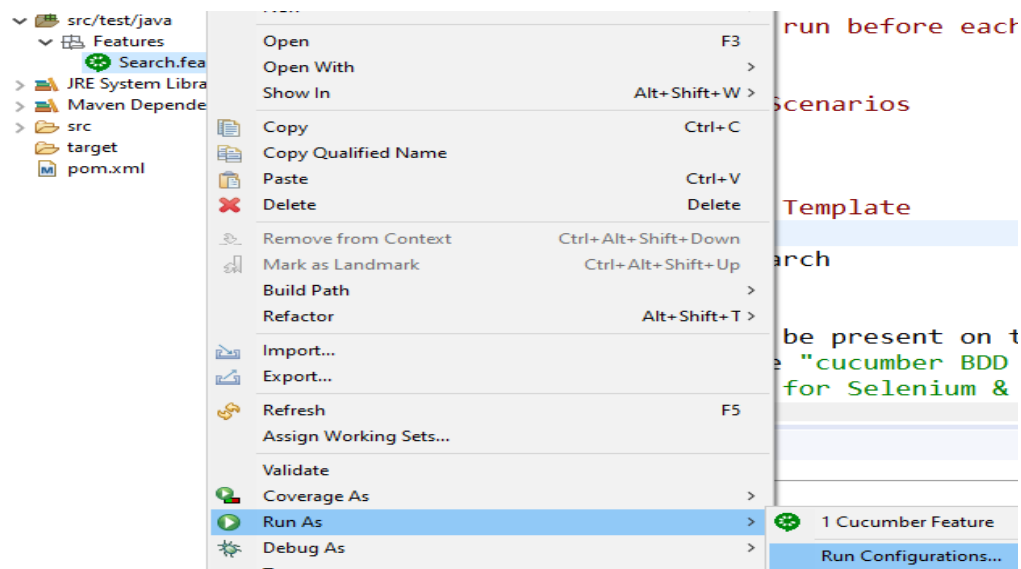
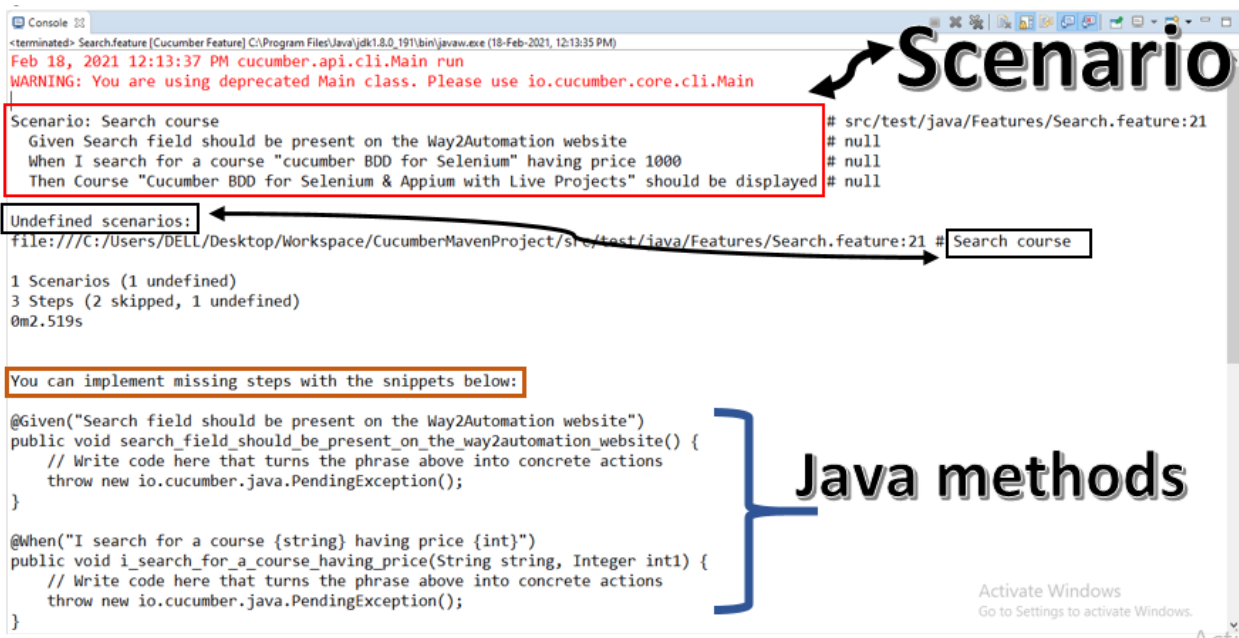See below. Expand 'Cucumber Feature' and select New_configuration.

Click 'Browse' button and select the path of the feature file

Execute the feature file

Click Apply, click Run

The below message would be seen in console:

```
@Then("Course {string} should be displayed")
public void course_should_be_displayed(String string) {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}


???????????????????????????????????????????????????????????????????????????????????
? Share your Cucumber Report with your team at https://reports.cucumber.io         ?
? Activate publishing with one of the following:                                   ?
?                                                                                  ?
? src/test/resources/cucumber.properties:          cucumber.publish.enabled=true   ?
? src/test/resources/junit-platform.properties:    cucumber.publish.enabled=true   ?
? Environment variable:                            CUCUMBER_PUBLISH_ENABLED=true   ?
? JUnit:                                           @CucumberOptions(publish = true) ?
?                                                                                  ?
? More information at https://reports.cucumber.io/docs/cucumber-jvm                ?
?                                                                                  ?
? Disable this message with one of the following:                                  ?
?                                                                                  ?
? src/test/resources/cucumber.properties:          cucumber.publish.quiet=true     ?
? src/test/resources/junit-platform.properties:    cucumber.publish.quiet=true     ?
???????????????????????????????????????????????????????????????????????????????????
```

Activate Windows
Go to Settings to activate Windows

The console has suggested us 3 java methods (code snippets) that need to be implemented. So basically, we have not yet implemented the 3 steps in our feature file. The 3 steps are: Given, When, Then.
The feature file currently is in plain English language.

You can also execute the feature file by right clicking the feature file > Run As > Cucumber Feature, see below.

Create Step Definition

Recall the cucumber framework design that we had discussed in previous tutorial. In a step definition file, we map the Give/When/Then steps using @ annotations

Business requirement or user story#1

Feature file#1 containing scenario(s) steps in Gherkin syntax (Given, When, Then..)
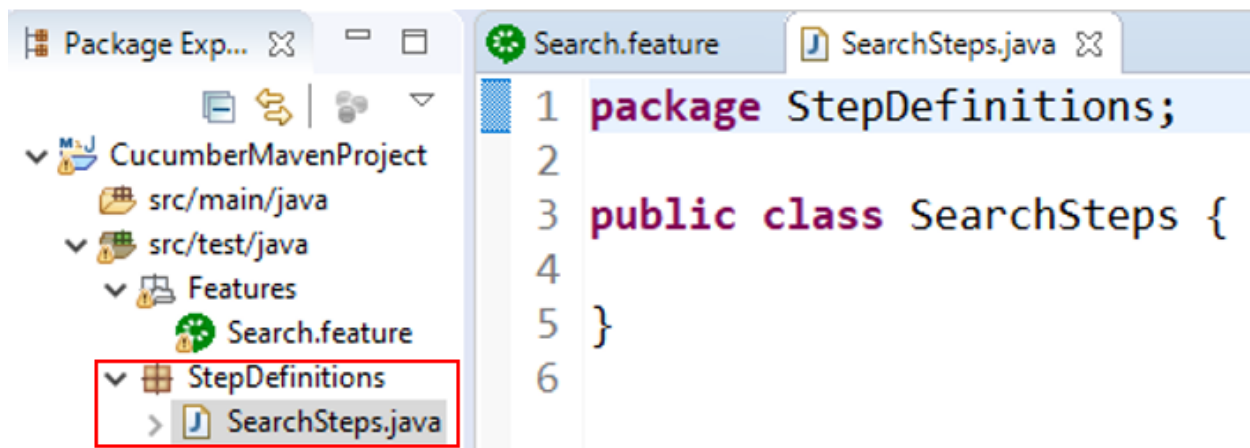
search.feature

Write a StepDefinition file in Java for search feature. Given/When/Then steps will be mapped to respective annotations, example @Given

StepDefinition file may use common utilities,e.g: Excelutility.java

TestRunner.java file will execute our step definition file. This java file would have -Junit runner, Cucumber options like report format, path of your step def and feature file, etc

Now, under src/test/java, create a new package 'StepDefinitions' and under this package, create a class

Next, go to the console and copy the code snippets

<terminated> Search.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (18-Feb-2021, 1:49:38 PM)

```
0m1.705s


You can implement missing steps with the snippets below:

@Given("Search field should be present on the Way2Automation website")
public void search_field_should_be_present_on_the_way2automation_website() {
    // Write code here that turns the phrase                      Cut            Ctrl+X    s
    throw new io.cucumber.java.PendingExcepti          Copy          Ctrl+C
}                                                                   Paste         Ctrl+V

                                                                    Select All     Ctrl+A
@When("I search for a course {string} having
public void i_search_for_a_course_having_pric                       Find/Replace...               nt1) {
    // Write code here that turns the phrase                        Open Link                      s
    throw new io.cucumber.java.PendingExcepti          Clear
}
                                                                    Remove All Terminated
                                                                    Terminate/Disconnect All
@Then("Course {string} should be displayed")
public void course_should_be_displayed(String                       Scroll Lock
    // Write code here that turns the phrase                        Word Wrap
    throw new io.cucumber.java.PendingException();                  s
}                                                                   Preferences...
```

Paste these methods in step definition file:

```
  Search.feature      SearchSteps.java ⊠
  1  package StepDefinitions;
  2
  3  public class SearchSteps {
  4⊖      @Given("Search field should be present on the Way2Automation website")
  5      public void search_field_should_be_present_on_the_way2automation_website() {
  6          // Write code here that turns the phrase above into concrete actions
  7          throw new io.cucumber.java.PendingException();
  8      }
  9
 10⊖      @When("I search for a course {string} having price {int}")
 11      public void i_search_for_a_course_having_price(String string, Integer int1) {
 12          // Write code here that turns the phrase above into concrete actions
 13          throw new io.cucumber.java.PendingException();
 14      }
 15
 16⊖      @Then("Course {string} should be displayed")
 17      public void course_should_be_displayed(String string) {
 18          // Write code here that turns the phrase above into concrete actions
 19          throw new io.cucumber.java.PendingException();
 20      }
 21  }
```

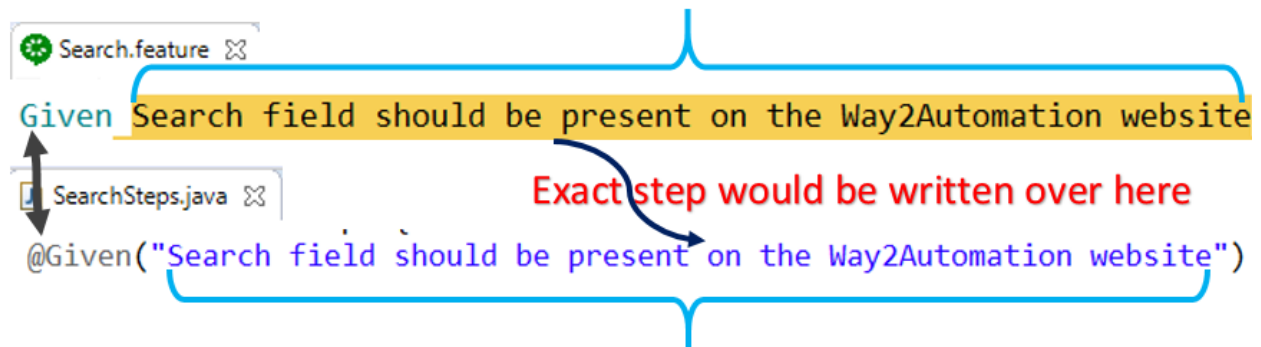Right now there are 3 errors. Press Ctrl+Shift+o to import Given/When/Then

The errors are resolved now.

See below. The 'Given' step in feature file is mapped to @Given in step definition file. Also, the **exact step name** written in feature file "Search field should be present on the Way2Automation website" is mapped to the step of @Given method in the step def file

Next, whatever is the step name "Search field should be present on the Way2Automation website", cucumber will generate the same method name

Note: You can change the method name to any other desired name, example search_field_should_be_present_on_the_website(). However, the step name CANNOT be changed. So, you **cannot** change the step name to @Given("Search field should be present on the website").

Next, the string that we mentioned within double quotes in the 'When' step, translates to {string} in the step definition file. Similalrly, the integer we mentioned in the 'When' step, translates to {int} in the step definition file. Also, cucumber automatically generates 2 method parameters: String, Integer
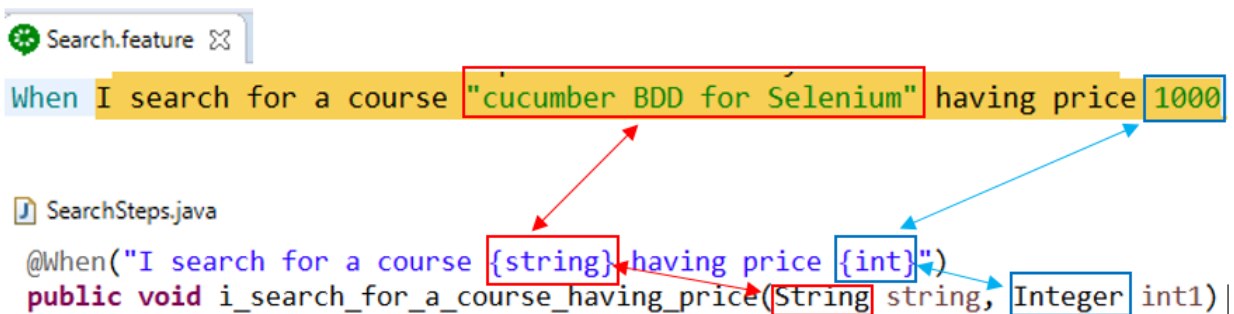
Next, you can change the variable names as seen below

Similarly, you can change the variable name in @Then method

One thing that you notice in the step definition file is that, cucumber automatically generates an exception. The reason being, we have not yet written any java code for the respective steps.

Next, run the feature file

```
#""
## (Comments)
#Sample Feature Definition Template
Feature: Way2Automation search

  Scenario: Search course
    Given Search field should be present on the Way2Aut
    When I search for a course "cucumber BDD for Seleni
    Then Course "Cucumber BDD for Selenium & Appium wit
```
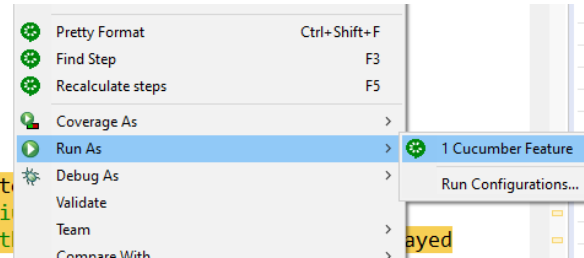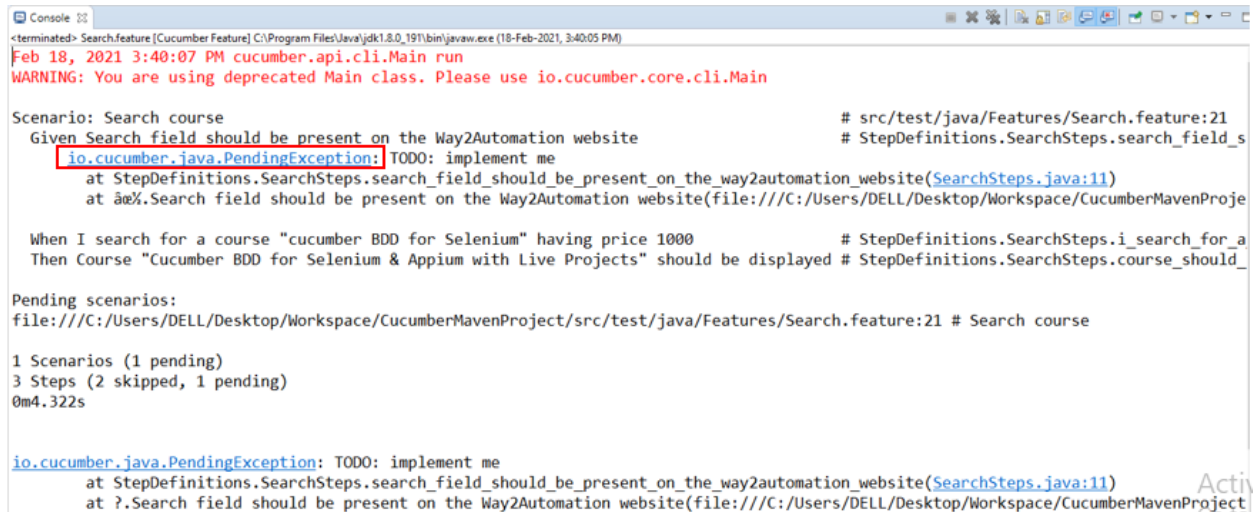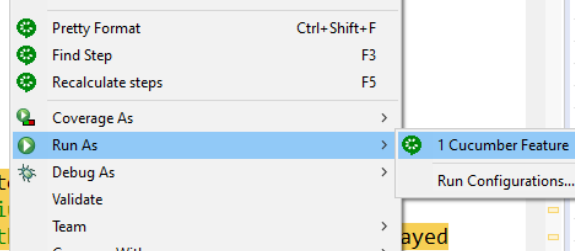
⊕ Pretty Format	Ctrl+Shift+F	
⊕ Find Step	F3	
⊕ Recalculate steps	F5	
⚐ Coverage As	>	
▶ Run As	>	⊕ 1 Cucumber Feature
⭑ Debug As	>	Run Configurations...
Validate		
Team	>	ayed
Compare With		

So, when you now run the feature file, the console shows the same Exception

```
Console ⊠
<terminated> Search.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (18-Feb-2021, 3:40:05 PM)
Feb 18, 2021 3:40:07 PM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main

Scenario: Search course                                                # src/test/java/Features/Search.feature:21
  Given Search field should be present on the Way2Automation website    # StepDefinitions.SearchSteps.search_field_s
    io.cucumber.java.PendingException: TODO: implement me
        at StepDefinitions.SearchSteps.search_field_should_be_present_on_the_way2automation_website(SearchSteps.java:11)
        at âœ%.Search field should be present on the Way2Automation website(file:///C:/Users/DELL/Desktop/Workspace/CucumberMavenProje

  When I search for a course "cucumber BDD for Selenium" having price 1000    # StepDefinitions.SearchSteps.i_search_for_a
  Then Course "Cucumber BDD for Selenium & Appium with Live Projects" should be displayed # StepDefinitions.SearchSteps.course_should_

Pending scenarios:
file:///C:/Users/DELL/Desktop/Workspace/CucumberMavenProject/src/test/java/Features/Search.feature:21 # Search course

1 Scenarios (1 pending)
3 Steps (2 skipped, 1 pending)
0m4.322s


io.cucumber.java.PendingException: TODO: implement me
        at StepDefinitions.SearchSteps.search_field_should_be_present_on_the_way2automation_website(SearchSteps.java:11)
        at ?.Search field should be present on the Way2Automation website(file:///C:/Users/DELL/Desktop/Workspace/CucumberMavenProject
```

To resolve this, let us remove the exception from the step def file, see below

```
 7 public class SearchSteps {
 8⊝    @Given("Search field should be present on the Way2Automation website")
 9     public void search_field_should_be_present_on_the_way2automation_website() {
10
11     }
12
13⊝    @When("I search for a course {string} having price {int}")
14     public void i_search_for_a_course_having_price(String courseName, Integer price) {
15
16     }
17
18⊝    @Then("Course {string} should be displayed")
19     public void course_should_be_displayed(String courseName) {
20         |
21     }
```

Next, we will write some simple java code SOPs

```
 7 public class SearchSteps {
 8⊖     @Given("Search field should be present on the Way2Automation website")
 9      public void search_field_should_be_present_on_the_way2automation_website() {
10          System.out.println("Step 1: I landed on search page");
11      }
12
13⊖     @When("I search for a course {string} having price {int}")
14      public void i_search_for_a_course_having_price(String courseName, Integer price) {
15          System.out.println("Step 2: Search the course with name: " + courseName + " price: " + price);
16      }
17
18⊖     @Then("Course {string} should be displayed")
19      public void course_should_be_displayed(String courseName) {
20          System.out.println("Step 3: course " + courseName + " is displayed");
21      }
22 }
```

Whatever course name and price you have written in the feature file, the same gets passed to 'courseName' and 'price' variables respectively in step def file.

Next, let us again run the feature file

Notice the console output, all 3 steps got passed this time. There are no exceptions.

Also notice that the respective course name and price gets passed to the SOPs, see below. You can also see the respective Give/When/Then steps in the console below. The course name mentioned in When/Then steps matches with the course names in respective SOPs

```
Step 1: I landed on search page
   Given Search field should be present on the Way2Automation website
Step 2: Search the course with name: cucumber BDD for Selenium price: 1000
   When I search for a course "cucumber BDD for Selenium" having price 1000
Step 3: course Cucumber BDD for Selenium & Appium with Live Projects is displayed
   Then Course "Cucumber BDD for Selenium & Appium with Live Projects" should be displayed
```

So, this is how we implement the step definition file.

Thank you for reading!