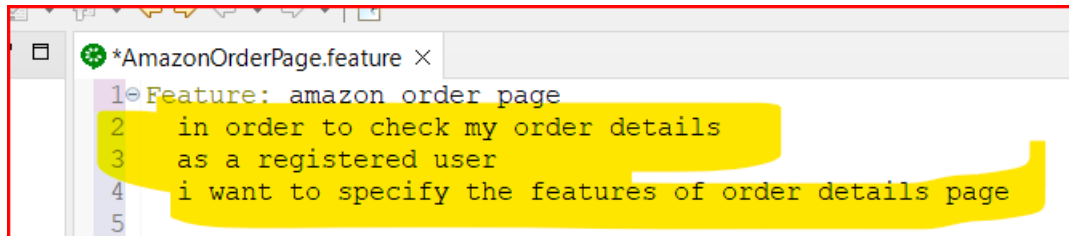


See the description below feature-

This is to give more information to the stakeholders.

We hit one tab key to make indentation more better and then add the description below features.



Sometimes you wont see this flagging-

```
4 i want to specify the features of order details page
5
6 Scenario: check previous order details
7   Given a registered user exists
8   Given user is on amazon login page
9   When user enters username
10  And user enters password
11  And user clicks on login button
12  Then user navigates to order page
13  Then user checks the previous order details
14
15 Scenario: check open order details
16   Given a registered user exists
17   Given user is on amazon login page
18   When user enters username
19   And user enters password
20   And user clicks on login button
21   Then user navigates to order page
22   Then user checks the open order details
23
24 Scenario: check cancelled order details
25   Given a registered user exists
26   Given user is on amazon login page
27   When user enters username
28   And user enters password
29   And user clicks on login button
30   Then user navigates to order page
31   Then user checks the cancelled order details
```

Then close and reopen the feature file.

If it does not work then right click on project -> maven -> update project -> select force update -> click ok.

Alignment and indentation is very important in feature file.

Background meaning-

Background

- Occasionally you'll find yourself repeating the same **Given** steps in all of the scenarios in a **Feature**.
- Since it is repeated in every scenario, this is an indication that those steps are not *essential* to describe the scenarios; they are *incidental details*. You can literally move such **Given** steps to the background, by grouping them under a **Background** section.
- A **Background** allows you to add some context to the scenarios that follow it. It can contain one or more **Given** steps, which are run before *each* scenario, but after any **Before hooks**.
- A **Background** is placed before the first **Scenario / Example**, at the same level of indentation.

Order Details
Order link
Previous order details

Order Details
Open Orders link
Open order details

Order Details

As per official
Cucumber
Doc

Background will run before each and every scenario runs. If there are ten scenarios, then background runs ten times.

Run the order feature file-

```
Feature: amazon order page
  in order to check my order details
  as a registered user
  i want to specify the features of order details page

Background:
  Given a registered user exists
  Given user is on amazon login page
  When user enters username
  And user enters password
  And user clicks on login button
  Then user navigates to order page

Scenario: check previous order details
  When user clicks on orders link
  Then user checks the previous order details

Scenario: check open order details
  When user clicks on open orders link
  Then user checks the open order details

Scenario: check cancelled order details
  When user clicks on cancelled orders link
  Then user checks the cancelled order details
```

You will get the code for step definition file in the console.

Create step def file.

Selenium is just a library to automate the browser.

Where is the location of .class file which is compiled from eclipse-

E:\Naveen Java Training\Cucumber\CucumberPractices\target\test-classes

Run sample dummy code-

Feature file:

```
Feature: amazon order page
  in order to check my order details
  as a registered user
  i want to specify the features of order details page

Background:
  Given a registered user exists
  Given user is on amazon login page
  When user enters username
  And user enters password
  And user clicks on login button
  Then user navigates to order page

Scenario: check previous order details
  When user clicks on orders link
  Then user checks the previous order details

Scenario: check open order details
  When user clicks on open orders link
  Then user checks the open order details

Scenario: check cancelled order details
  When user clicks on cancelled orders link
  Then user checks the cancelled order details
```

Step def file:

```
package StepDefinitions;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class AmazonOrderPageStepDef {

    @Given("a registered user exists")
    public void a_registered_user_exists() {
    }

    @Given("user is on amazon login page")
    public void user_is_on_amazon_login_page() {
    }

    @When("user enters username")
    public void user_enters_username() {
    }

    @When("user enters password")
    public void user_enters_password() {
    }
}
```

```
@When("user clicks on login button")
public void user_clicks_on_login_button() {

}

@Then("user navigates to order page")
public void user_navigates_to_order_page() {

}

@When("user clicks on orders link")
public void user_clicks_on_orders_link() {

}

@Then("user checks the previous order details")
public void user_checks_the_previous_order_details() {

}

@When("user clicks on open orders link")
public void user_clicks_on_open_orders_link() {

}

@Then("user checks the open order details")
public void user_checks_the_open_order_details() {

}

@When("user clicks on cancelled orders link")
public void user_clicks_on_cancelled_orders_link() {

}

@Then("user checks the cancelled order details")
public void user_checks_the_cancelled_order_details() {

}

}
```

Output:

```
Console x Terminal
<terminated> New_configuration [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe (20-Jun-2023, 7:55:00 am - 7:55:01 am) [pid: 10616]
Jun 20, 2023 7:55:00 AM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main

Scenario: check previous order details # src/test/resources/AppFeatures/AmazonOrderPage.feature:14
  Given a registered user exists # StepDefinitions.AmazonOrderPageStepDef.a_registered_user_exists()
  Given user is on amazon login page # StepDefinitions.AmazonOrderPageStepDef.user_is_on_amazon_login_page()
  When user enters username # StepDefinitions.AmazonOrderPageStepDef.user_enters_username()
  And user enters password # StepDefinitions.AmazonOrderPageStepDef.user_enters_password()
  And user clicks on login button # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_login_button()
  Then user navigates to order page # StepDefinitions.AmazonOrderPageStepDef.user_navigates_to_order_page()
  When user clicks on orders link # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_orders_link()
  Then user checks the previous order details # StepDefinitions.AmazonOrderPageStepDef.user_checks_the_previous_order_details()

Scenario: check open order details # src/test/resources/AppFeatures/AmazonOrderPage.feature:18
  Given a registered user exists # StepDefinitions.AmazonOrderPageStepDef.a_registered_user_exists()
  Given user is on amazon login page # StepDefinitions.AmazonOrderPageStepDef.user_is_on_amazon_login_page()
  When user enters username # StepDefinitions.AmazonOrderPageStepDef.user_enters_username()
  And user enters password # StepDefinitions.AmazonOrderPageStepDef.user_enters_password()
  And user clicks on login button # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_login_button()
  Then user navigates to order page # StepDefinitions.AmazonOrderPageStepDef.user_navigates_to_order_page()
  When user clicks on open orders link # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_open_orders_link()
  Then user checks the open order details # StepDefinitions.AmazonOrderPageStepDef.user_checks_the_open_order_details()

Scenario: check cancelled order details # src/test/resources/AppFeatures/AmazonOrderPage.feature:22
  Given a registered user exists # StepDefinitions.AmazonOrderPageStepDef.a_registered_user_exists()
  Given user is on amazon login page # StepDefinitions.AmazonOrderPageStepDef.user_is_on_amazon_login_page()
  When user enters username # StepDefinitions.AmazonOrderPageStepDef.user_enters_username()
  And user enters password # StepDefinitions.AmazonOrderPageStepDef.user_enters_password()
  And user clicks on login button # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_login_button()
  Then user navigates to order page # StepDefinitions.AmazonOrderPageStepDef.user_navigates_to_order_page()
  When user clicks on cancelled orders link # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_cancelled_orders_link()
  Then user checks the cancelled order details # StepDefinitions.AmazonOrderPageStepDef.user_checks_the_cancelled_order_details()

3 Scenarios (3 passed)
24 Steps (24 passed)
0m0.442s
```

Background should be short and crisp.

Don't write lengthy.

Don't write complex steps.

Created runner and running as junit test using runner-

```
package testRunners;

import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;
import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(plugin = { "pretty" }, features = {
    "src/test/resources/AppFeatures/AmazonOrderPage.feature" },
glue = { "StepDefinitions" })

public class AmazonOrdersPageTest {

}
```

Output:

Package Explorer JUnit x

Finished after 0.25 seconds

Runs: 3/3 Errors: 0 Failures: 0

AmazonOrdersPageTest [Runner: JUnit 5] (0.001 s)

- amazon order page (0.001 s)
 - check previous order details (0.001 s)
 - check open order details (0.000 s)
 - check cancelled order details (0.000 s)

Console:

```
<terminated> AmazonOrdersPageTest [JUnit] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe (20-Jun-2023, 8:01:23 am - 8:01:24 am) [pid: 3672]

Scenario: check previous order details # src/test/resources/AppFeatures/AmazonOrderPage.feature:14
  Given a registered user exists # StepDefinitions.AmazonOrderPageStepDef.a_registered_user_exists()
  Given user is on amazon login page # StepDefinitions.AmazonOrderPageStepDef.user_is_on_amazon_login_page()
  When user enters username # StepDefinitions.AmazonOrderPageStepDef.user_enters_username()
  And user enters password # StepDefinitions.AmazonOrderPageStepDef.user_enters_password()
  And user clicks on login button # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_login_button()
  Then user navigates to order page # StepDefinitions.AmazonOrderPageStepDef.user_navigates_to_order_page()
  When user clicks on orders link # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_orders_link()
  Then user checks the previous order details # StepDefinitions.AmazonOrderPageStepDef.user_checks_the_previous_order_details()

Scenario: check open order details # src/test/resources/AppFeatures/AmazonOrderPage.feature:18
  Given a registered user exists # StepDefinitions.AmazonOrderPageStepDef.a_registered_user_exists()
  Given user is on amazon login page # StepDefinitions.AmazonOrderPageStepDef.user_is_on_amazon_login_page()
  When user enters username # StepDefinitions.AmazonOrderPageStepDef.user_enters_username()
  And user enters password # StepDefinitions.AmazonOrderPageStepDef.user_enters_password()
  And user clicks on login button # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_login_button()
  Then user navigates to order page # StepDefinitions.AmazonOrderPageStepDef.user_navigates_to_order_page()
  When user clicks on open orders link # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_open_orders_link()
  Then user checks the open order details # StepDefinitions.AmazonOrderPageStepDef.user_checks_the_open_order_details()

Scenario: check cancelled order details # src/test/resources/AppFeatures/AmazonOrderPage.feature:22
  Given a registered user exists # StepDefinitions.AmazonOrderPageStepDef.a_registered_user_exists()
  Given user is on amazon login page # StepDefinitions.AmazonOrderPageStepDef.user_is_on_amazon_login_page()
  When user enters username # StepDefinitions.AmazonOrderPageStepDef.user_enters_username()
  And user enters password # StepDefinitions.AmazonOrderPageStepDef.user_enters_password()
  And user clicks on login button # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_login_button()
  Then user navigates to order page # StepDefinitions.AmazonOrderPageStepDef.user_navigates_to_order_page()
  When user clicks on cancelled orders link # StepDefinitions.AmazonOrderPageStepDef.user_clicks_on_cancelled_orders_link()
  Then user checks the cancelled order details # StepDefinitions.AmazonOrderPageStepDef.user_checks_the_cancelled_order_details()

Share your Cucumber Report with your team at https://reports.cucumber.io
```

Run using maven-

Go to the project path in command line.

```
Command Prompt

Apache Maven 3.9.2 (c9616018c7a021c1c39be70fb2843d6f5f9b8a1c)
Maven home: C:\Program Files\ApacheMaven\apache-maven-3.9.2-bin\apache-maven-3.9.2
Java version: 1.8.0_202, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_202\jre
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

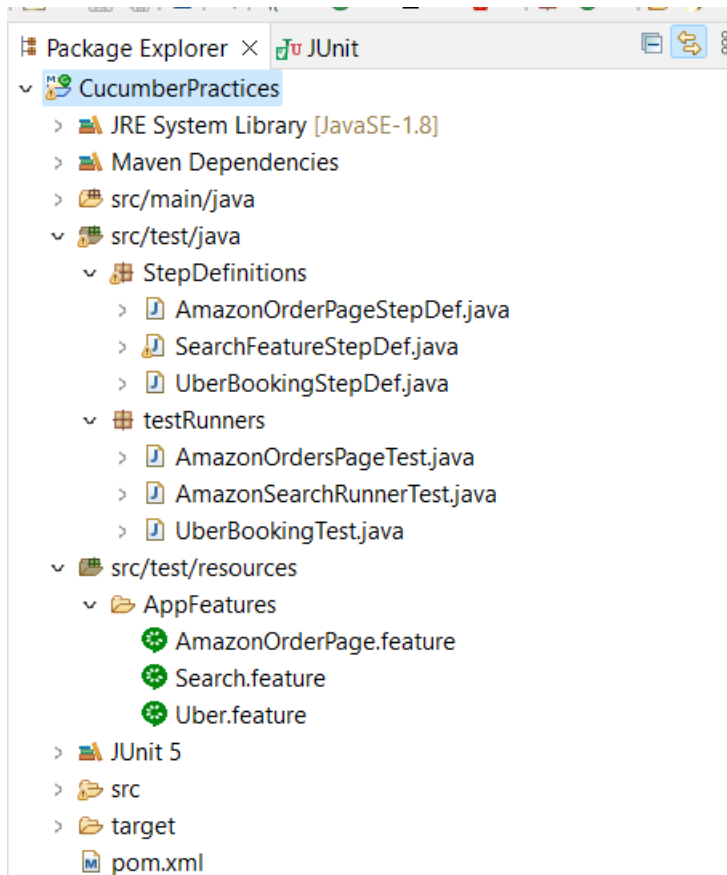
E:\Naveen Java Training\Cucumber\CucumberPractices>mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< CucumberYouTubeSeries:CucumberPractices >-----
[INFO] Building CucumberPractices 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/io/cucumber/messages/maven-metadata.xml
Downloaded from central: https://repo.maven.apache.org/maven2/io/cucumber/messages/maven-metadata.xml (2.4 kB at 5.2 kB/s)
[INFO]
[INFO] --- resources:3.3.0:resources (default-resources) @ CucumberPractices ---
[INFO] skip non existing resourceDirectory E:\Naveen Java Training\Cucumber\CucumberPractices\src\main\resources
[INFO]
[INFO] --- compiler:3.8.1:compile (default-compile) @ CucumberPractices ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- resources:3.3.0:testResources (default-testResources) @ CucumberPractices ---
[INFO] Copying 3 resources
[INFO]
```

```

? src/test/resources/junit-platform.properties:  cucumber.publish.quiet=true  ?
????????????????????????????????????????????????????????????????????????????????????
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.006 s - in testRunners.UberBookingTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.784 s
[INFO] Finished at: 2023-06-20T08:04:32+02:00
[INFO] -----
[WARNING]
[WARNING] Plugin validation issues were detected in 3 plugin(s)
[WARNING]
[WARNING] * org.apache.maven.plugins:maven-compiler-plugin:3.8.1
[WARNING] * org.apache.maven.plugins:maven-surefire-plugin:2.22.2
[WARNING] * org.apache.maven.plugins:maven-resources-plugin:3.3.0
[WARNING]
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the values (case insensitive): [BRIEF, DEFAULT, VERBOSE]
[WARNING]
E:\Naveen Java Training\Cucumber\CucumberPractices>

```

Project structure-



Codes-

Output for codes already mentioned earlier.

Feature file for amazon order:

```

Feature: amazon order page
  in order to check my order details
  as a registered user
  i want to specify the features of order details page

Background:
  Given a registered user exists
  Given user is on amazon login page

```

```

    When user enters username
    And user enters password
    And user clicks on login button
    Then user navigates to order page

Scenario: check previous order details
    When user clicks on orders link
    Then user checks the previous order details

Scenario: check open order details
    When user clicks on open orders link
    Then user checks the open order details

Scenario: check cancelled order details
    When user clicks on cancelled orders link
    Then user checks the cancelled order details

```

Amazon order step def:

```

package StepDefinitions;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class AmazonOrderPageStepDef {

    @Given("a registered user exists")
    public void a_registered_user_exists() {
    }

    @Given("user is on amazon login page")
    public void user_is_on_amazon_login_page() {
    }

    @When("user enters username")
    public void user_enters_username() {
    }

    @When("user enters password")
    public void user_enters_password() {
    }

    @When("user clicks on login button")
    public void user_clicks_on_login_button() {
    }

    @Then("user navigates to order page")
    public void user_navigates_to_order_page() {
    }

    @When("user clicks on orders link")
    public void user_clicks_on_orders_link() {
    }

    @Then("user checks the previous order details")

```



```

    public void user_checks_the_previous_order_details() {

    }

    @When("user clicks on open orders link")
    public void user_clicks_on_open_orders_link() {

    }

    @Then("user checks the open order details")
    public void user_checks_the_open_order_details() {

    }

    @When("user clicks on cancelled orders link")
    public void user_clicks_on_cancelled_orders_link() {

    }

    @Then("user checks the cancelled order details")
    public void user_checks_the_cancelled_order_details() {

    }

}

```

Runner:

```

package testRunners;

import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;
import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(plugin = { "pretty" }, features = {
    "src/test/resources/AppFeatures/AmazonOrderPage.feature" },
glue = { "StepDefinitions" })

public class AmazonOrdersPageTest {

}

```