

Way2Automation - Tutorial 6 – Background keyword in Cucumber

What you will Learn :

- o About 'Background' keyword in cucumber
- o User story
- o Create feature file to capture scenarios
- o Background keyword
- o Reference

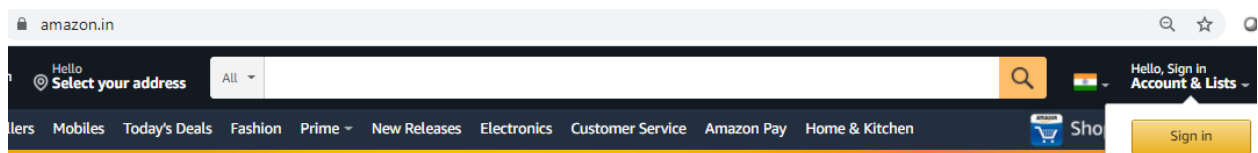
About 'Background' keyword in cucumber

Background keyword is used to define some pre-conditions for all the scenarios. Many a times, we repeat the same 'Given' step in all the scenarios. Since 'Given' step is repeated in every scenario, we can move such repeated steps to the background, by grouping them under a 'Background' section.

User story

The user story or the requirement is that:

We have to launch www.amazon.in and click 'Sign in'



Next we will enter our user id and click 'Continue'

Sign-In

Email or mobile phone number

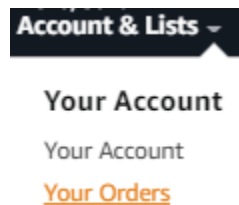
Continue

Next we enter our password and click 'Sign-in'

Password

Sign-In

Once we login, we will click 'Your Orders'



We will see our 'Orders', 'Open Orders', 'Cancelled Orders'

Your Orders

Orders Buy Again Open Orders Cancelled Orders

So we will write 3 scenarios over here:

When I log in and go to order details page, I want to see my previous orders

When I log in and go to order details page, I want to see my open orders

When I log in and go to order details page, I want to see my cancelled orders

Create feature file to capture scenarios

Create a feature file and mention the 3 scenarios, see below

```
Background.feature
1 Feature: My Amazon Orders
2
3 Scenario: Check Previous order details
4 Given a registered user exists
5 Given user is on Amazon login page
6 When user enters username
7 And user enters password
8 And user clicks on sign-in button
9 Then user navigates to order page
10 When user clicks on Orders link
11 Then user checks the previous order details
12
13 Scenario: Check Open order details
14 Given a registered user exists
15 Given user is on Amazon login page
16 When user enters username
17 And user enters password
18 And user clicks on sign-in button
19 Then user navigates to order page
20 When user clicks on Open Orders link
21 Then user checks the open order details
22
23 Scenario: Check Cancelled order details
24 Given a registered user exists
25 Given user is on Amazon login page
26 When user enters username
27 And user enters password
28 And user clicks on sign-in button
29 Then user navigates to order page
30 When user clicks on Cancelled Orders link
31 Then user checks the cancelled order details
```

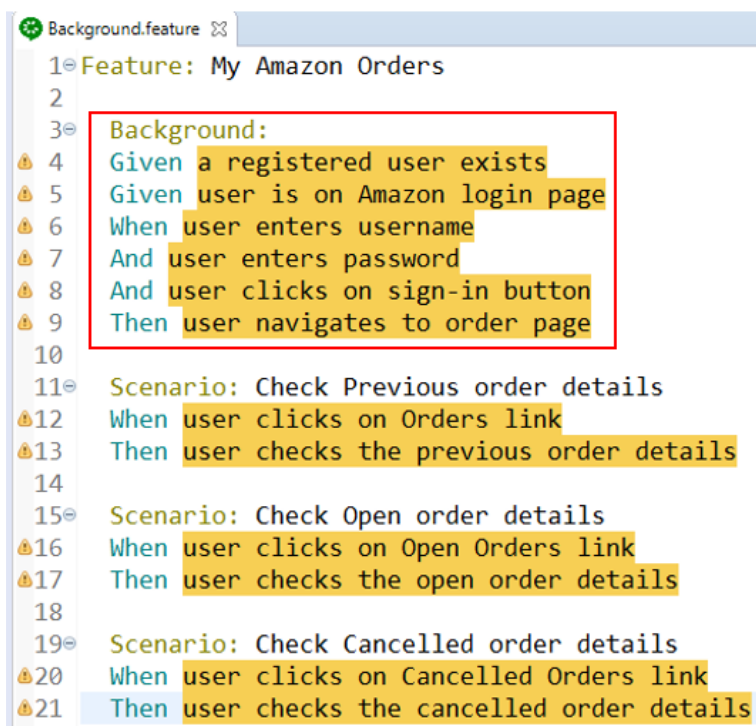
If you carefully notice the 3 scenarios, the below steps are repeated in each scenario

```
Given a registered user exists
Given user is on Amazon login page
When user enters username
And user enters password
And user clicks on sign-in button
Then user navigates to order page
```

It is not advisable to repeat the steps inside the scenarios.

Background keyword

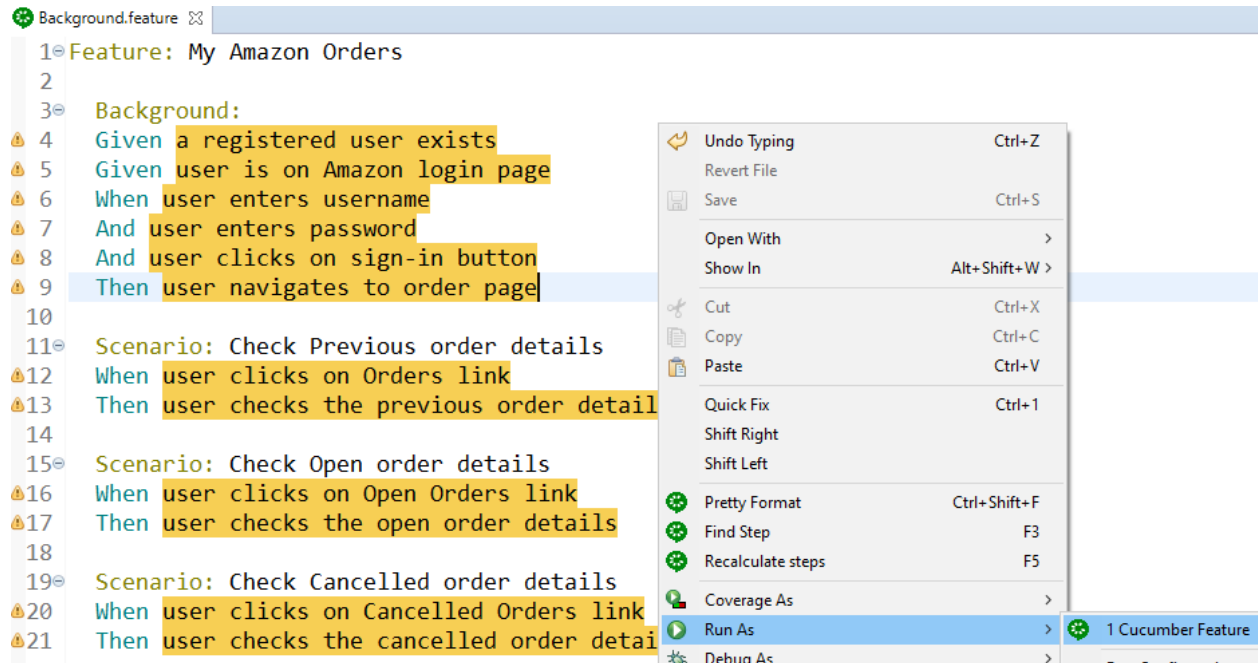
We will now write all the common steps as part of 'Background' keyword, see below. Delete all the common steps from the 3 scenarios. So we will be left with below. If you notice, the feature is more readable now. Each scenario is now focussing on what exactly we want to test as part of that scenario. All the common steps need not be written for every scenario



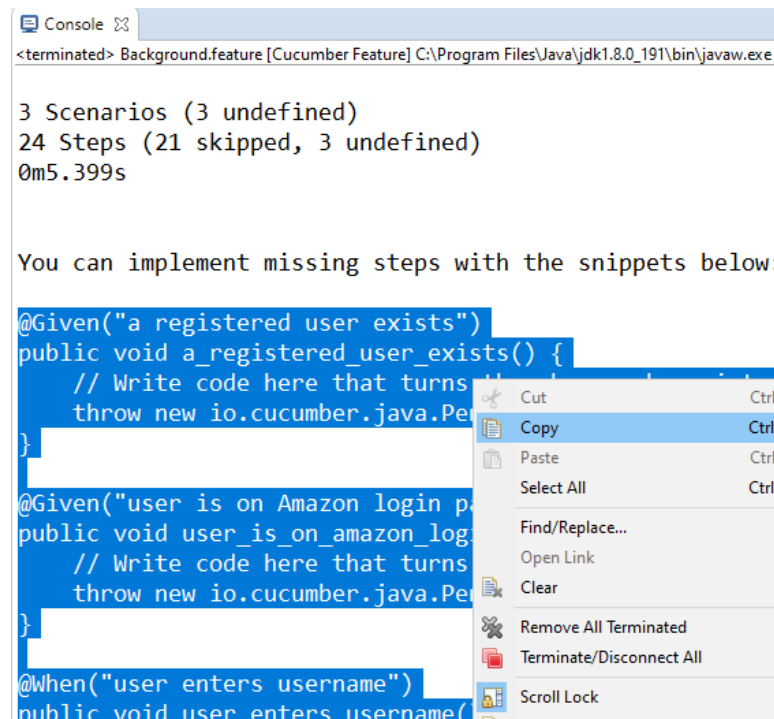
```
Background.feature
1 Feature: My Amazon Orders
2
3 Background:
4   Given a registered user exists
5   Given user is on Amazon login page
6   When user enters username
7   And user enters password
8   And user clicks on sign-in button
9   Then user navigates to order page
10
11 Scenario: Check Previous order details
12   When user clicks on Orders link
13   Then user checks the previous order details
14
15 Scenario: Check Open order details
16   When user clicks on Open Orders link
17   Then user checks the open order details
18
19 Scenario: Check Cancelled order details
20   When user clicks on Cancelled Orders link
21   Then user checks the cancelled order details
```

So the property of 'Background' keyword is that, when you run it, it will be executed before each and every scenario. So in this case, it would be executed 3 times since we have 3 scenarios.

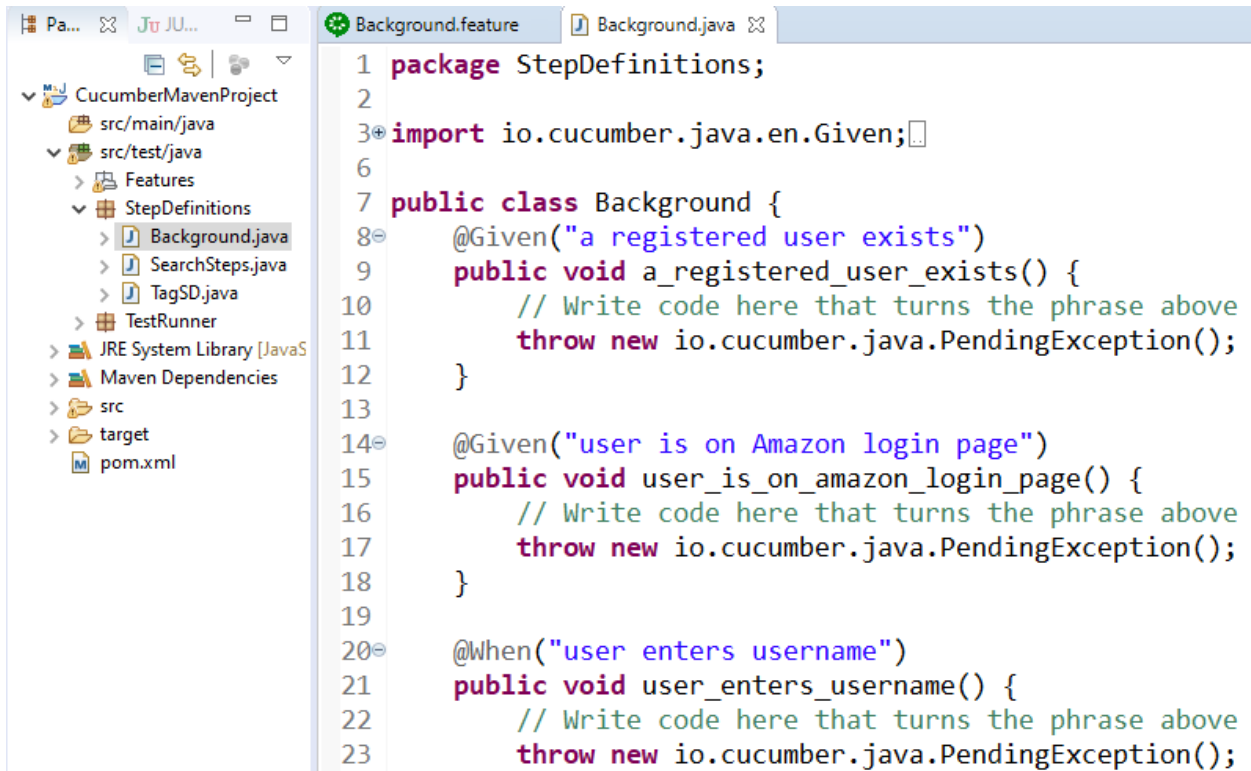
So save the feature file and run it



Copy the code snippets from the console window, see below

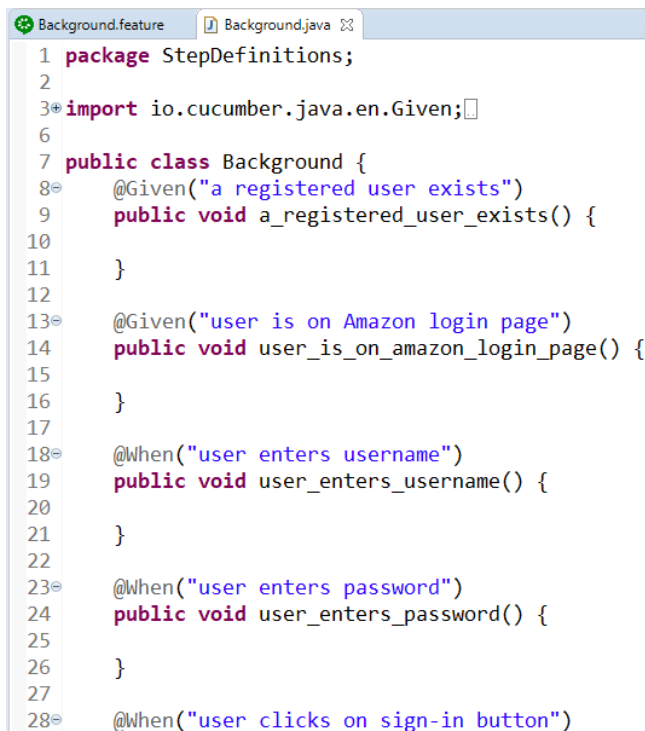


Create a step definition file and paste it. Import the missing classes



```
1 package StepDefinitions;
2
3 import io.cucumber.java.en.Given;
4
5
6
7 public class Background {
8     @Given("a registered user exists")
9     public void a_registered_user_exists() {
10         // Write code here that turns the phrase above
11         throw new io.cucumber.java.PendingException();
12     }
13
14     @Given("user is on Amazon login page")
15     public void user_is_on_amazon_login_page() {
16         // Write code here that turns the phrase above
17         throw new io.cucumber.java.PendingException();
18     }
19
20     @When("user enters username")
21     public void user_enters_username() {
22         // Write code here that turns the phrase above
23         throw new io.cucumber.java.PendingException();
24     }
25 }
```

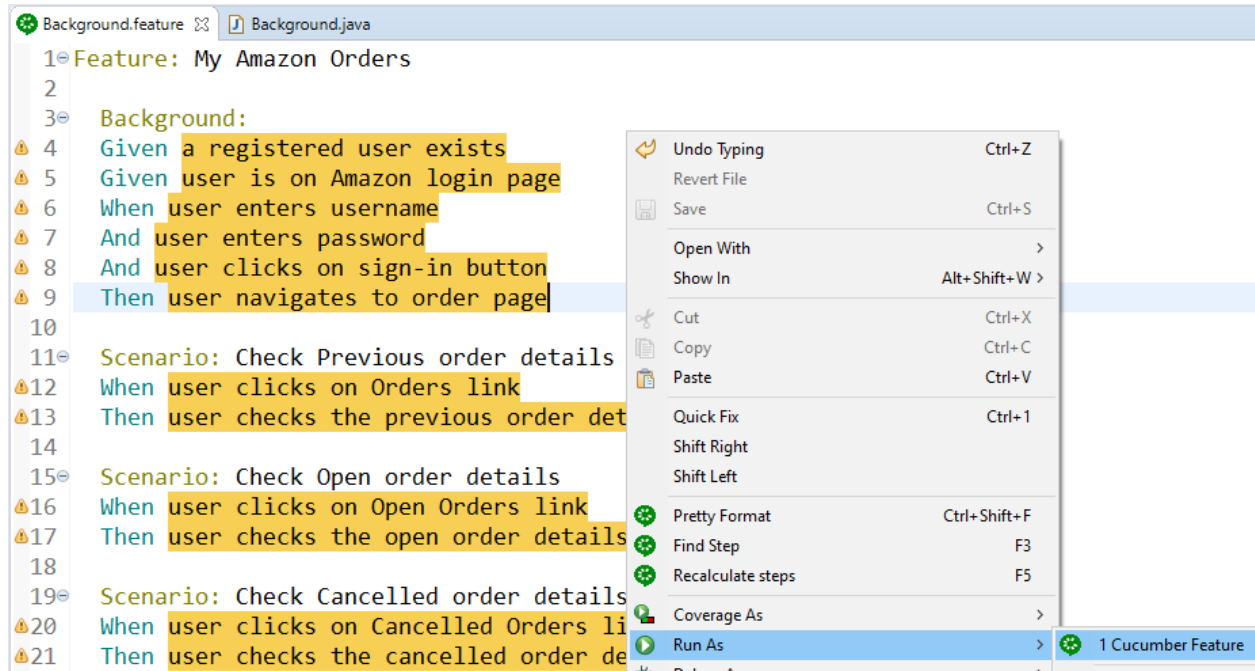
Delete all the throw exceptions



```
1 package StepDefinitions;
2
3 import io.cucumber.java.en.Given;
4
5
6
7 public class Background {
8     @Given("a registered user exists")
9     public void a_registered_user_exists() {
10     }
11
12
13     @Given("user is on Amazon login page")
14     public void user_is_on_amazon_login_page() {
15     }
16
17
18     @When("user enters username")
19     public void user_enters_username() {
20     }
21
22
23     @When("user enters password")
24     public void user_enters_password() {
25     }
26
27
28     @When("user clicks on sign-in button")
29 }
```

Save the step def file.

Let us run the feature file once again



Notice the console o/p


```
Console
<terminated> Background.feature [Cucumber Feature] C:\Program Files\Java\jdk1.8.0_191\bin\
Scenario: Check Previous order details # sr
  Given a registered user exists # St
  Given user is on Amazon login page # St
  When user enters username # St
  And user enters password # St
  And user clicks on sign-in button # St
  Then user navigates to order page # St
  When user clicks on Orders link # St
  Then user checks the previous order details # St

Scenario: Check Open order details # src/te
  Given a registered user exists # StepDe
  Given user is on Amazon login page # StepDe
  When user enters username # StepDe
  And user enters password # StepDe
  And user clicks on sign-in button # StepDe
  Then user navigates to order page # StepDe
  When user clicks on Open Orders link # StepDe
  Then user checks the open order details # StepDe

Scenario: Check Cancelled order details # s
  Given a registered user exists # S
  Given user is on Amazon login page # S
  When user enters username # S
  And user enters password # S
  And user clicks on sign-in button # S
  Then user navigates to order page # S
  When user clicks on Cancelled Orders link # S
  Then user checks the cancelled order details # S
```

The o/p is exactly same as we were expecting. All the common steps that we wrote as part of 'Background' section get automatically attached to each and every scenario.

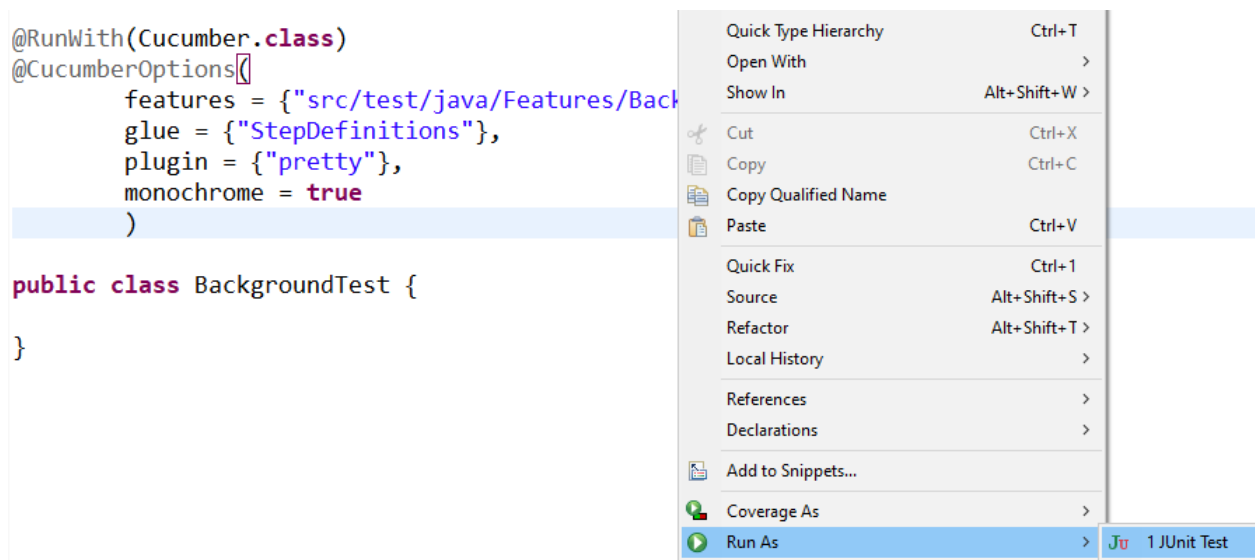
Let us create a runner file



```
1 package TestRunner;
2
3 import org.junit.runner.RunWith;
4
5
6
7
8 @RunWith(Cucumber.class)
9 @CucumberOptions(
10     features = {"src/test/java/Features/Background.feature"},
11     glue = {"StepDefinitions"},
12     plugin = {"pretty"},
13     monochrome = true
14 )
15
16 public class BackgroundTest {
17
18 }
```

Save the file

Let us run it



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

public class BackgroundTest {

}
```

We get the same output

```

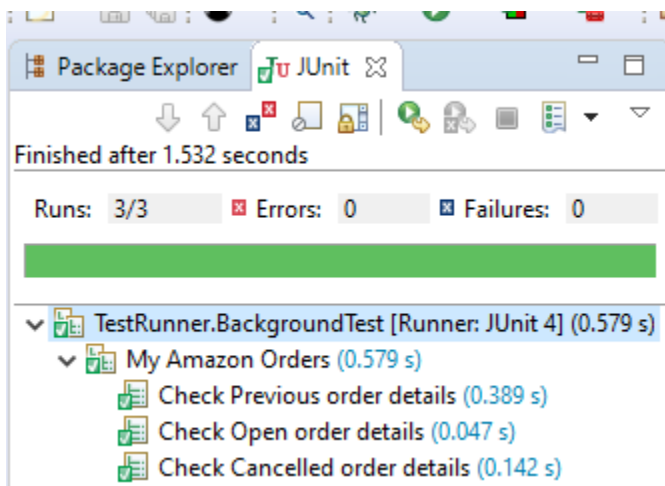
Console
<terminated> BackgroundTest [JUnit] C:\Program Files\Java\jdk1.8.0_191\bin\javaw.exe (24-Feb
Scenario: Check Previous order details          # sr
  Given a registered user exists                 # St
  Given user is on Amazon login page             # St
  When user enters username                     # St
  And user enters password                      # St
  And user clicks on sign-in button              # St
  Then user navigates to order page              # St
  When user clicks on Orders link                # St
  Then user checks the previous order details    # St

Scenario: Check Open order details              # src/te
  Given a registered user exists                 # StepDe
  Given user is on Amazon login page             # StepDe
  When user enters username                     # StepDe
  And user enters password                      # StepDe
  And user clicks on sign-in button              # StepDe
  Then user navigates to order page              # StepDe
  When user clicks on Open Orders link          # StepDe
  Then user checks the open order details        # StepDe

Scenario: Check Cancelled order details          # s
  Given a registered user exists                 # S
  Given user is on Amazon login page             # S
  When user enters username                     # S
  And user enters password                      # S
  And user clicks on sign-in button              # S
  Then user navigates to order page              # S
  When user clicks on Cancelled Orders link      # S
  Then user checks the cancelled order details   # S

```

Even the junit runner shows 3 scenarios getting passed



So this is how we use the 'Background' keyword.

Reference

You can refer <https://relishapp.com/cucumber/cucumber/docs/gherkin/background> for further documentation

relishapp.com/cucumber/cucumber/docs/gherkin/background

Api

Cli

Defining steps

Docs

Extending cucumber

Formatters

Gherkin

◦ Background

› Choosing the language from the feature file header

› Doc strings

› Language help

› Scenario outlines

› Scenario outlines --expand option

› Unicode in tables

› Using descriptions to give features context

› Using star notation instead of Given/When/Then

Wire protocol

Writing support code

Background

Often you find that several scenarios in the same feature start with a common context.

Cucumber provides a mechanism for this, by providing a **Background** keyword where you can specify steps that should be run before each scenario in the feature. Typically these will be **Given** steps, but you can use any steps that you need to.

Hint: If you find that some of the scenarios don't fit the background, consider splitting them into a separate feature.

Background

Given a file named "features/passing_background.feature" with:

```
Feature: Passing background sample

Background:
  Given '10' cukes

Scenario: passing background
  Then I should have '10' cukes

Scenario: another passing background
  Then I should have '10' cukes
```

And a file named "features/scenario_outline_passing_background.feature" with:

Thank you for reading!