# This is cucumber expression what we have written:



# We cannot mix regular expression and cucumber expression in a single method -

A picture containing text, font, line, screenshot

Description automatically generated

Else we get error. And step definition will also be flagged.

A picture containing text, screenshot, font

Description automatically generated

See line 20 where we have used both forms of expression.

But inside a file, for example, if there are ten methods, two of them can have regular expression only and 8 of them can have cucumber expression only.

# Regular expression-

Used to fetch certain values.

# Quantifiers in regular expression-

How many times a character needs to occur.

# Capture group –

The “()” brackets are known as capture group.

A screenshot of a computer

Description automatically generated with medium confidence

Note-

{n} says total number of occurrences of the letter or digit.



A screenshot of a computer

Description automatically generated with low confidence

# Shorthand characters guide-

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated

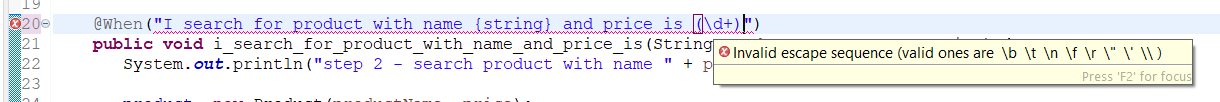
[Predefined shorthand character classes - Java 9 Regular Expressions [Book] (oreilly.com)](https://www.oreilly.com/library/view/java-9-regular/9781787288706/251e3389-2f47-4c59-8ac6-f58a2f24556f.xhtml)



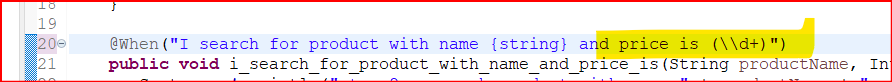
A picture containing text, font, screenshot

Description automatically generated

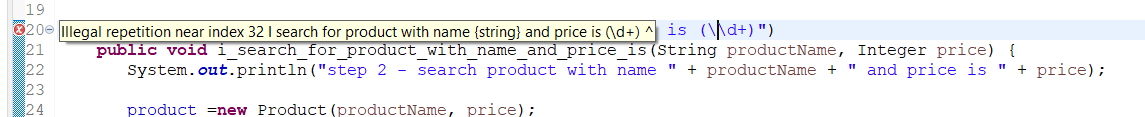
# Invalid escape sequence error displayed when we use regular expression-



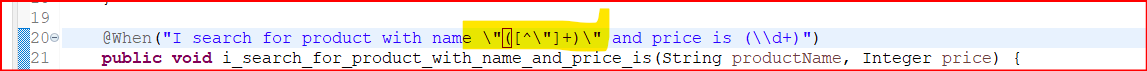
This is because we need to one more slash before “\”. See in the above error, the list of valid escape sequences.



Still, we will get error for mixing regular expression with cucumber expression:



This is regular expression and it will capture all the strings except double quote:



The highlighted step is not flagged, which means our regular expression is correct.

A close-up of a logo

Description automatically generated with low confidence

Run the search feature file and we should get error:

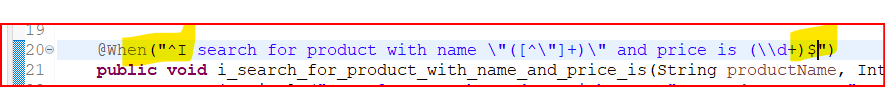
|  |
| --- |
| io.cucumber.core.exception.CucumberException: Step [I search for product with name "([^"]+)" and price is (\d+)] is defined with 2 parameters at 'StepDefinitions.SearchFeatureStepDef.i\_search\_for\_product\_with\_name\_and\_price\_is(java.lang.String,java.lang.Integer)'.  However, the gherkin step has 0 arguments.  Step text: I search for product with name "apple" and price is 1000  at io.cucumber.core.runner.PickleStepDefinitionMatch.arityMismatch(PickleStepDefinitionMatch.java:83)  at io.cucumber.core.runner.PickleStepDefinitionMatch.runStep(PickleStepDefinitionMatch.java:42)  at io.cucumber.core.runner.ExecutionMode$1.execute(ExecutionMode.java:10)  at io.cucumber.core.runner.TestStep.executeStep(TestStep.java:92)  at io.cucumber.core.runner.TestStep.run(TestStep.java:64)  at io.cucumber.core.runner.PickleStepTestStep.run(PickleStepTestStep.java:51)  at io.cucumber.core.runner.TestCase.run(TestCase.java:104)  at io.cucumber.core.runner.Runner.runPickle(Runner.java:71)  at io.cucumber.core.runtime.Runtime.lambda$execute$5(Runtime.java:110)  at io.cucumber.core.runtime.CucumberExecutionContext.runTestCase(CucumberExecutionContext.java:117)  at io.cucumber.core.runtime.Runtime.lambda$execute$6(Runtime.java:110)  at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)  at java.util.concurrent.FutureTask.run(FutureTask.java:266)  at io.cucumber.core.runtime.Runtime$SameThreadExecutorService.execute(Runtime.java:233)  at java.util.concurrent.AbstractExecutorService.submit(AbstractExecutorService.java:112)  at io.cucumber.core.runtime.Runtime.lambda$run$2(Runtime.java:86)  at java.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.java:193)  at java.util.stream.SliceOps$1$1.accept(SliceOps.java:204)  at java.util.ArrayList$ArrayListSpliterator.tryAdvance(ArrayList.java:1359)  at java.util.stream.ReferencePipeline.forEachWithCancel(ReferencePipeline.java:126)  at java.util.stream.AbstractPipeline.copyIntoWithCancel(AbstractPipeline.java:498)  at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:485)  at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline.java:471)  at java.util.stream.ReduceOps$ReduceOp.evaluateSequential(ReduceOps.java:708)  at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)  at java.util.stream.ReferencePipeline.collect(ReferencePipeline.java:499)  at io.cucumber.core.runtime.Runtime.run(Runtime.java:87)  at io.cucumber.core.cli.Main.run(Main.java:92)  at cucumber.api.cli.Main.run(Main.java:30)  at cucumber.api.cli.Main.main(Main.java:15) |

How to resolve the above error:

Open the step definition file of say, search feature.

Add “^” at start and “$” at end.

Then run the search feature file.



Output:

|  |
| --- |
| Jun 12, 2023 4:51:27 PM cucumber.api.cli.Main run  WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main  Scenario: Search for a product # src/test/resources/AppFeatures/Search.feature:24  step 1 - i am on search page  Given I have a search field on Amazon page # StepDefinitions.SearchFeatureStepDef.i\_have\_a\_search\_field\_on\_amazon\_page()  step 2 - search product with name apple and price is 1000  When I search for product with name "apple" and price is 1000 # StepDefinitions.SearchFeatureStepDef.i\_search\_for\_product\_with\_name\_and\_price\_is(java.lang.String,java.lang.Integer)  step 3 - product with apple : is displayed  returned product is apple  Then Product with name "apple" should be displayed # StepDefinitions.SearchFeatureStepDef.product\_with\_name\_should\_be\_displayed(java.lang.String)  1 Scenarios (1 passed)  3 Steps (3 passed)  0m0.371s |

# Codes-

Search feature file:

|  |
| --- |
| #Author: your.email@your.domain.com  #Keywords Summary :  #Feature: List of scenarios.  #Scenario: Business rule through list of steps with arguments.  #Given: Some precondition step  #When: Some key actions  #Then: To observe outcomes or validation  #And,But: To enumerate more Given,When,Then steps  #Scenario Outline: List of steps for data-driven as an Examples and <placeholder>  #Examples: Container for s table  #Background: List of steps run before each of the scenarios  #""" (Doc Strings)  #| (Data Tables)  #@ (Tags/Labels):To group Scenarios  #<> (placeholder)  #""  ## (Comments)  #Sample Feature Definition Template  Feature: Amazon Search  #this is the feature for which we want to write code and requirements  #one feature file can have multiple scenarios  Scenario: Search for a product  #here we give the scenario name  #below scenario (given, when, then, and) etc are known as steps  Given I have a search field on Amazon page  #given can be considered as pre-condition  When I search for product with name "apple" and price is 1000  #string is in double quotes  Then Product with name "apple" should be displayed |

Step definition file for search feature:

|  |
| --- |
| package StepDefinitions;  import io.cucumber.java.en.Given;  import io.cucumber.java.en.Then;  import io.cucumber.java.en.When;  import junit.framework.Assert;  import amazonImplementation.Product;  import amazonImplementation.Search;  public class SearchFeatureStepDef {  Product product;  Search search;  @Given("I have a search field on Amazon page")  public void i\_have\_a\_search\_field\_on\_amazon\_page() {  System.out.println("step 1 - i am on search page");  }  @When("^I search for product with name \"([^\"]+)\" and price is (\\d+)$")  public void i\_search\_for\_product\_with\_name\_and\_price\_is(String productName, Integer price) {  System.out.println("step 2 - search product with name " + productName + " and price is " + price);  product = new Product(productName, price);  }  @Then("Product with name {string} should be displayed")  public void product\_with\_name\_should\_be\_displayed(String productName) {  System.out.println("step 3 - product with " + productName + " : is displayed");  search = new Search();  String productNameReturned = search.displayProductName(product);  System.out.println("returned product is " + productNameReturned);  Assert.assertEquals(product.getProductName(), productNameReturned);  }  } |

Output:

|  |
| --- |
| Jun 12, 2023 4:51:27 PM cucumber.api.cli.Main run  WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main  Scenario: Search for a product # src/test/resources/AppFeatures/Search.feature:24  step 1 - i am on search page  Given I have a search field on Amazon page # StepDefinitions.SearchFeatureStepDef.i\_have\_a\_search\_field\_on\_amazon\_page()  step 2 - search product with name apple and price is 1000  When I search for product with name "apple" and price is 1000 # StepDefinitions.SearchFeatureStepDef.i\_search\_for\_product\_with\_name\_and\_price\_is(java.lang.String,java.lang.Integer)  step 3 - product with apple : is displayed  returned product is apple  Then Product with name "apple" should be displayed # StepDefinitions.SearchFeatureStepDef.product\_with\_name\_should\_be\_displayed(java.lang.String)  1 Scenarios (1 passed)  3 Steps (3 passed)  0m0.371s |