# When you get this error- Steps are defined and glued but still error is displayed that step definition is not present-

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Write only the package name and remove the full path which we had given like “/src/test/resources/StepDefinitions”.

A screenshot of a computer

Description automatically generated with low confidence

# Run with tag = smoke-

Feature file:

|  |
| --- |
| Feature: Uber Booking Feature  @Smoke  Scenario: Booking Cab  Given User selects car type "Sedan" from app  When user selects car as "Sedan" and pick up as "bangalore" and drop as "pine"  Then driver starts the journey  And driver ends the journey  Then user pays 100 us dollars  @Regression  Scenario: Booking Cab  Given User selects car type "suv" from app  When user selects car as "suv" and pick up as "bangalore" and drop as "hyderabad"  Then driver starts the journey  And driver ends the journey  Then user pays 100 us dollars  @Prod  Scenario: Booking Cab  Given User selects car type "mini" from app  When user selects car as "mini" and pick up as "mumbai" and drop as "pine"  Then driver starts the journey  And driver ends the journey  Then user pays 100 us dollars |

Step def of uber:

Even though we have written multiple scenarios, since there is no change in feature file, except passing different sets of data, so the same code will work.

|  |
| --- |
| package StepDefinitions;  import io.cucumber.java.en.Given;  import io.cucumber.java.en.Then;  import io.cucumber.java.en.When;  public class UberBookingStepDef {  @Given("User selects car type {string} from app")  public void user\_selects\_car\_type\_from\_app(String carType) {  System.out.println("step 1" + carType);  }  @When("user selects car as {string} and pick up as {string} and drop as {string}")  public void user\_selects\_car\_as\_and\_pick\_up\_as\_and\_drop\_as(String carType, String pickupLocation,  String dropLocation) {  System.out.println("step 2" + carType + " " + pickupLocation + " " + dropLocation);  }  @Then("driver starts the journey")  public void driver\_starts\_the\_journey() {  System.out.println("step 3");  }  @Then("driver ends the journey")  public void driver\_ends\_the\_journey() {  System.out.println("step 4");  }  @Then("user pays {int} us dollars")  public void user\_pays\_us\_dollars(Integer price) {  System.out.println("step 5" + " " + price);  }  } |

Runner of uber:

|  |
| --- |
| 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/Uber.feature" }, tags = "@Smoke", glue = { "StepDefinitions" })  //tags cannot be applied to given, when, then etc etc.  //tags can be applied at feature level. by default it will be applied on all scenarios.  //tags can be applied at scenario level.  //tags can be applied at scenario outline level.  public class UberBookingTest {  } |

Output:

A screenshot of a computer

Description automatically generated with medium confidence

# Added multiple values to the tag with OR operation-

Runner file:

|  |
| --- |
| 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/Uber.feature" }, tags = "@Smoke or @Regression", glue = { "StepDefinitions" })  //tags cannot be applied to given, when, then etc etc.  //tags can be applied at feature level. by default it will be applied on all scenarios.  //tags can be applied at scenario level.  //tags can be applied at scenario outline level.  //we can run with multiple tags.  public class UberBookingTest {  } |

Output:

A screenshot of a computer

Description automatically generated

Console:

|  |
| --- |
|  |

# Multiple tag values with “and” operation-

Runner file with and:

|  |
| --- |
| 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/Uber.feature" }, tags = "@Smoke and @Regression", glue = { "StepDefinitions" })  //tags cannot be applied to given, when, then etc etc.  //tags can be applied at feature level. by default it will be applied on all scenarios.  //tags can be applied at scenario level.  //tags can be applied at scenario outline level.  //we can run with multiple tags.  public class UberBookingTest {  } |

Output:

Nothing returned as there is no scenario with both tags (Smoke and Regression) present.

A screenshot of a computer

Description automatically generated

# Add multiple tags to scenario-

Feature file:

A picture containing text, font, screenshot, line

Description automatically generated

Run the runner as junit:

A screenshot of a computer

Description automatically generated with medium confidence

# Using not operator in runner-

Runner:

A screenshot of a computer

Description automatically generated with medium confidence

Junit output:

A screenshot of a computer

Description automatically generated with medium confidence

Console output:

A screenshot of a computer program

Description automatically generated with medium confidence

# “All” tag at feature level-

All scenarios for the feature will inherit the feature level tags.

Feature file:

A picture containing text, font, line, screenshot

Description automatically generated

Runner file:

A screen shot of a computer

Description automatically generated with medium confidence

Junit output:

A screenshot of a computer error

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

Console output:

See, every scenario has the “@All” tag associated.

|  |
| --- |
|  |

# Run from maven-

Open cmd.

Go to project path.

Type “mvn test”

You should get all success.

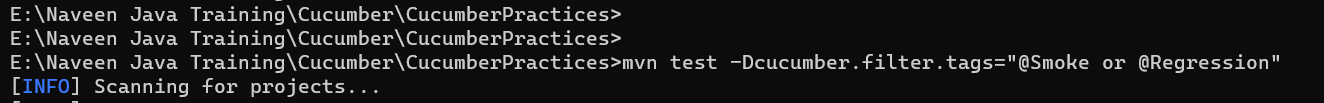
A screenshot of a computer

Description automatically generated

# How to override the tags written inside runner using maven command line-

Upto cucumber 6 we needed to use the word “-Dcucumber.options”. From cucumber 6 we started using “filters”.

The “prod” tag did not run as we have overridden “tags=All” inside runner file, with maven filters.



A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

# We can use such naming conventions to link to jira and epics and for reporting purpose-

A picture containing text, font, screenshot

Description automatically generated

# Codes for reference purpose –

Feature file:

|  |
| --- |
| @All  Feature: Uber Booking Feature  @Smoke  Scenario: Booking Cab Sedan  Given User selects car type "Sedan" from app  When user selects car as "Sedan" and pick up as "bangalore" and drop as "pine"  Then driver starts the journey  And driver ends the journey  Then user pays 100 us dollars  @Regression @Smoke  Scenario: Booking Cab SUV  Given User selects car type "suv" from app  When user selects car as "suv" and pick up as "bangalore" and drop as "hyderabad"  Then driver starts the journey  And driver ends the journey  Then user pays 100 us dollars  @Prod  Scenario: Booking Cab MINI  Given User selects car type "mini" from app  When user selects car as "mini" and pick up as "mumbai" and drop as "pine"  Then driver starts the journey  And driver ends the journey  Then user pays 100 us dollars |

Runner file:

|  |
| --- |
| 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/Uber.feature" }, tags = "@All", glue = { "StepDefinitions" })  //tags cannot be applied to given, when, then etc etc.  //tags can be applied at feature level. by default it will be applied on all scenarios.  //tags can be applied at scenario level.  //tags can be applied at scenario outline level.  //we can run with multiple tags.  public class UberBookingTest {  } |

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 UberBookingStepDef {  @Given("User selects car type {string} from app")  public void user\_selects\_car\_type\_from\_app(String carType) {  System.out.println("step 1" + carType);  }  @When("user selects car as {string} and pick up as {string} and drop as {string}")  public void user\_selects\_car\_as\_and\_pick\_up\_as\_and\_drop\_as(String carType, String pickupLocation,  String dropLocation) {  System.out.println("step 2" + carType + " " + pickupLocation + " " + dropLocation);  }  @Then("driver starts the journey")  public void driver\_starts\_the\_journey() {  System.out.println("step 3");  }  @Then("driver ends the journey")  public void driver\_ends\_the\_journey() {  System.out.println("step 4");  }  @Then("user pays {int} us dollars")  public void user\_pays\_us\_dollars(Integer price) {  System.out.println("step 5" + " " + price);  }  } |

Output is shown above in this document when we ran the code.