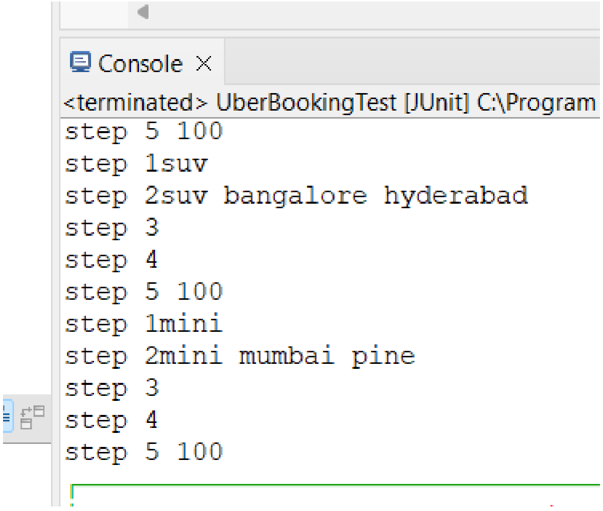
Without pretty plugin see the output-

```
package testRunners;

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

@RunWith(Cucumber.class)
@CucumberOptions( 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 {

}
```

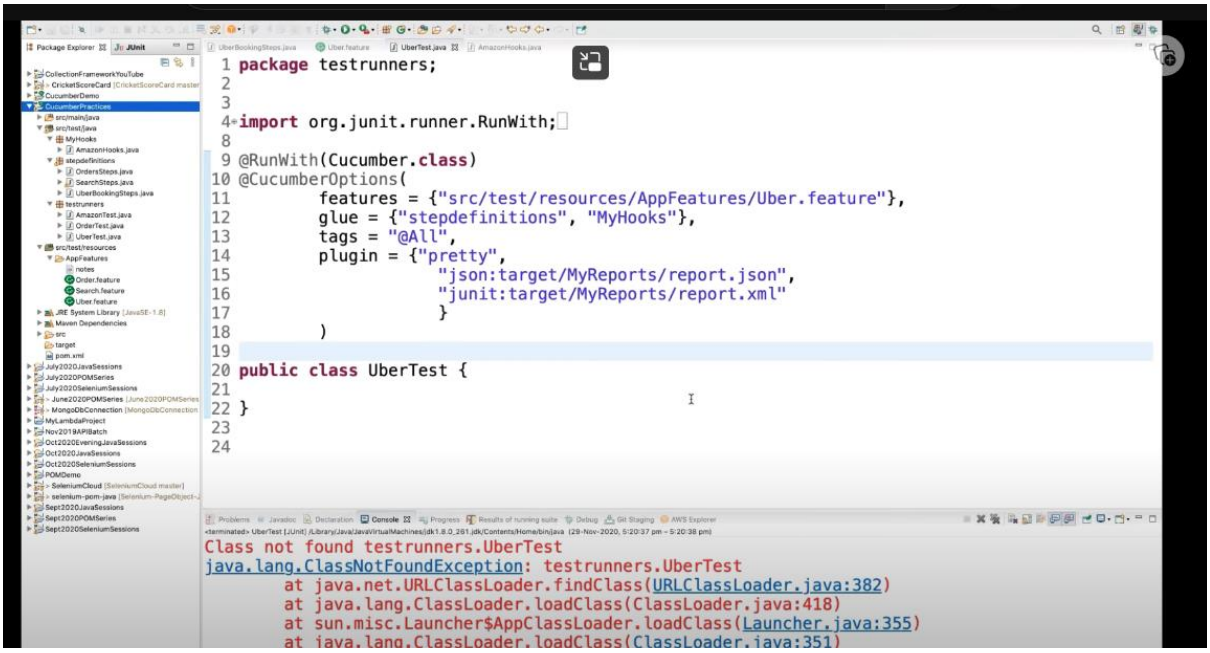
Console:



```
<terminated> UberBookingTest [JUnit] C:\Program
step 5 100
step 1suv
step 2suv bangalore hyderabad
step 3
step 4
step 5 100
step 1mini
step 2mini mumbai pine
step 3
step 4
step 5 100
```

Pretty will prettily print on output.

Naveen deleted target folder and tried to run program and got the below error-



```java
package testrunners;


import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(
        features = {"src/test/resources/AppFeatures/Uber.feature"},
        glue = {"stepdefinitions", "MyHooks"},
        tags = "@All",
        plugin = {"pretty",
                        "json:target/MyReports/report.json",
                        "junit:target/MyReports/report.xml"
                        }
        )

public class UberTest {

}
```

```
<terminated> UberTest [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/Contents/Home/bin/java (29-Nov-2020, 5:20:37 pm – 5:20:38 pm)
Class not found testrunners.UberTest
java.lang.ClassNotFoundException: testrunners.UberTest
        at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:355)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
```

To solve this, right click on project -> maven -> update project.

Run again and it should be solved. Some of the file may have been deleted from target folder which is needed for project, so rebuilding the project works in such case.

Add json and junit report-

```
package testRunners;

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

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report in the mentioned location
@CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",
                        "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.feature" },
tags = "@All",
glue = { "StepDefinitions", "MyHooks"})

//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 {

}
```

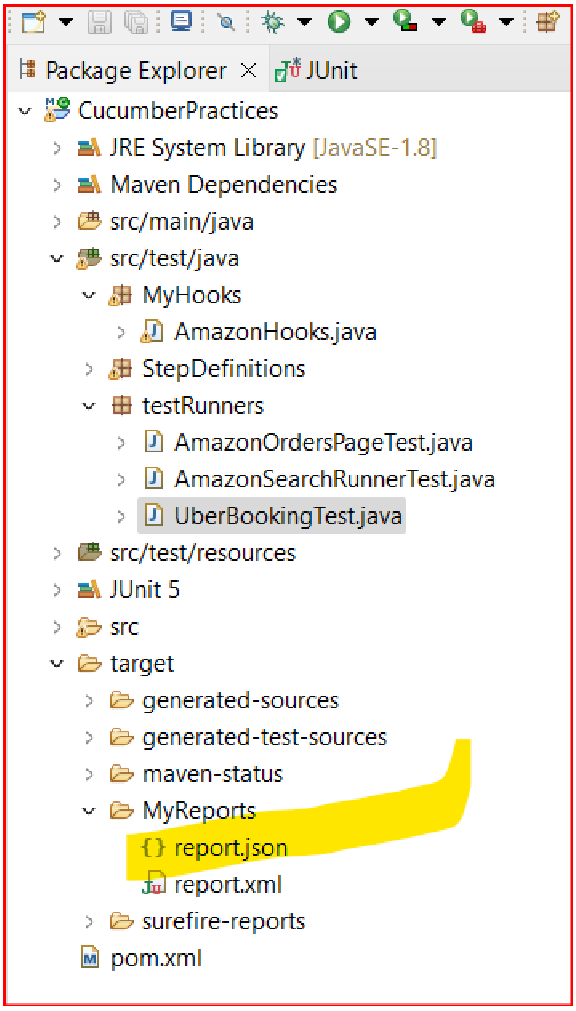
Refresh project by right clicking on it.

Target folder.

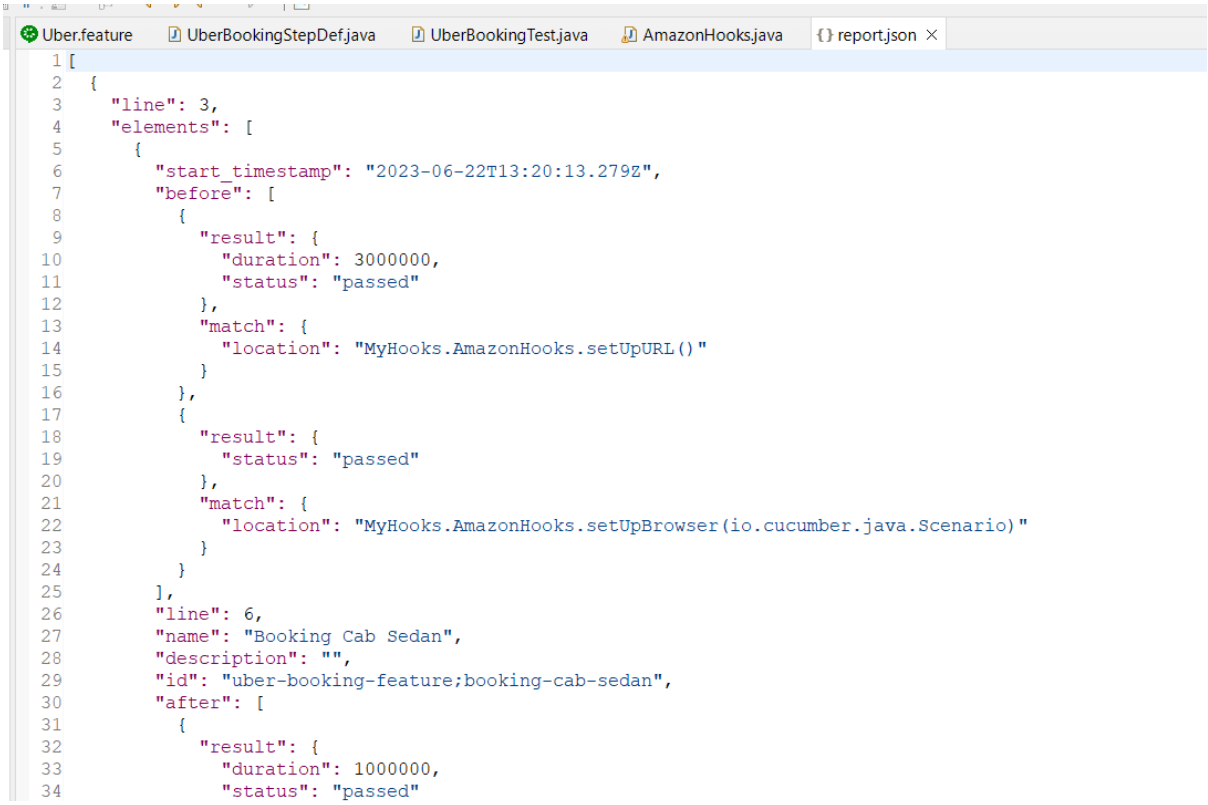
My reports.

Open the reports.

Note- even if we delete the target folder it will be regenerated during the run.

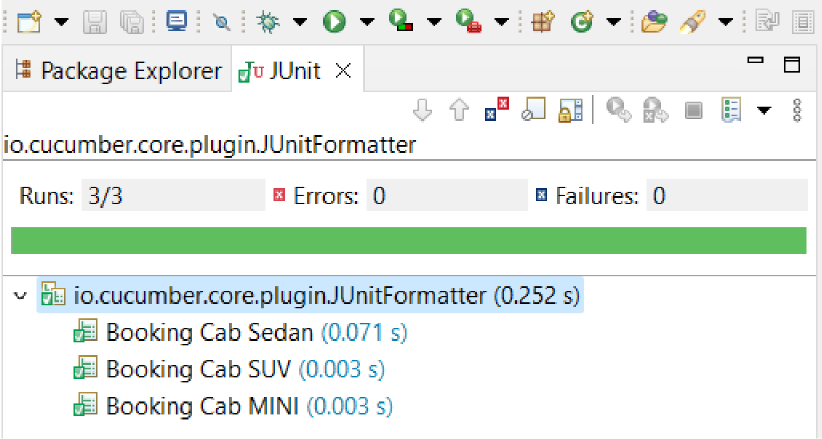


```
CucumberPractices
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src/main/java
  src/test/java
    MyHooks
      AmazonHooks.java
    StepDefinitions
    testRunners
      AmazonOrdersPageTest.java
      AmazonSearchRunnerTest.java
      UberBookingTest.java
  src/test/resources
  JUnit 5
  src
  target
    generated-sources
    generated-test-sources
    maven-status
    MyReports
      report.json
      report.xml
    surefire-reports
  pom.xml
```



```
[
  {
    "line": 3,
    "elements": [
      {
        "start_timestamp": "2023-06-22T13:20:13.279Z",
        "before": [
          {
            "result": {
              "duration": 3000000,
              "status": "passed"
            },
            "match": {
              "location": "MyHooks.AmazonHooks.setUpURL()"
            }
          },
          {
            "result": {
              "status": "passed"
            },
            "match": {
              "location": "MyHooks.AmazonHooks.setUpBrowser(io.cucumber.java.Scenario)"
            }
          }
        ],
        "line": 6,
        "name": "Booking Cab Sedan",
        "description": "",
        "id": "uber-booking-feature;booking-cab-sedan",
        "after": [
          {
            "result": {
              "duration": 1000000,
              "status": "passed"
```

Double click on junit report and it is the same we use daily-



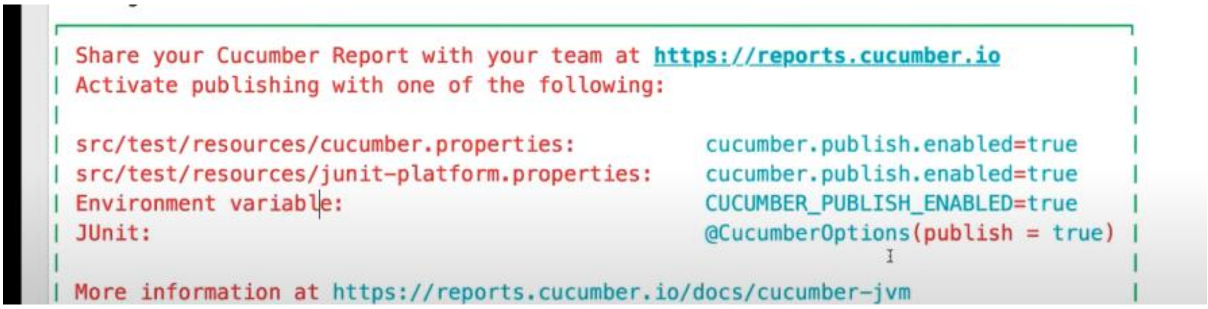
io.cucumber.core.plugin.JUnitFormatter

Runs: 3/3 ☒ Errors: 0 ☒ Failures: 0

- io.cucumber.core.plugin.JUnitFormatter (0.252 s)
  - Booking Cab Sedan (0.071 s)
  - Booking Cab SUV (0.003 s)
  - Booking Cab MINI (0.003 s)

Now cucumber has given option to publish report-

This has come from cucumber 6.9.



```
| 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             |
```

The above message comes in console when we run the test.

We can use any of the four options cucumber is suggesting.

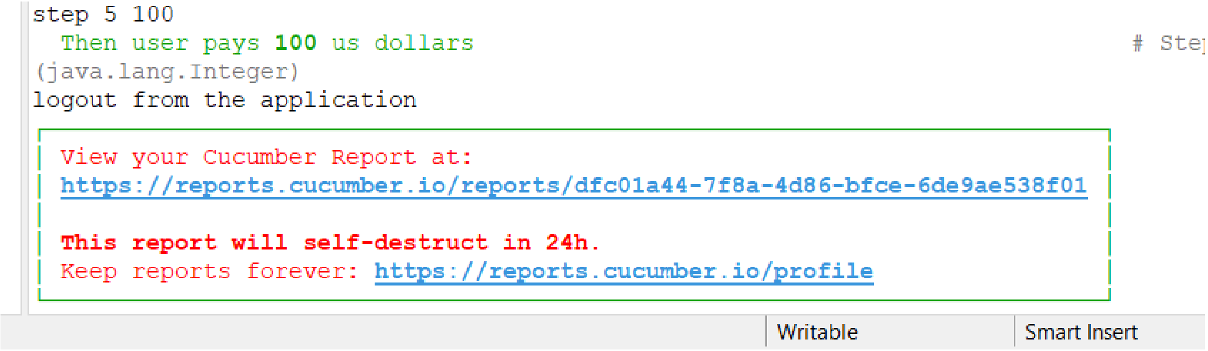
Publish cucumber reports-

Runner:

```
package testRunners;

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

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report in the mentioned location
@CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",
                          "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.feature" },
tags = "@All",
glue = { "StepDefinitions", "MyHooks"},
publish=true)

//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 {

}
```

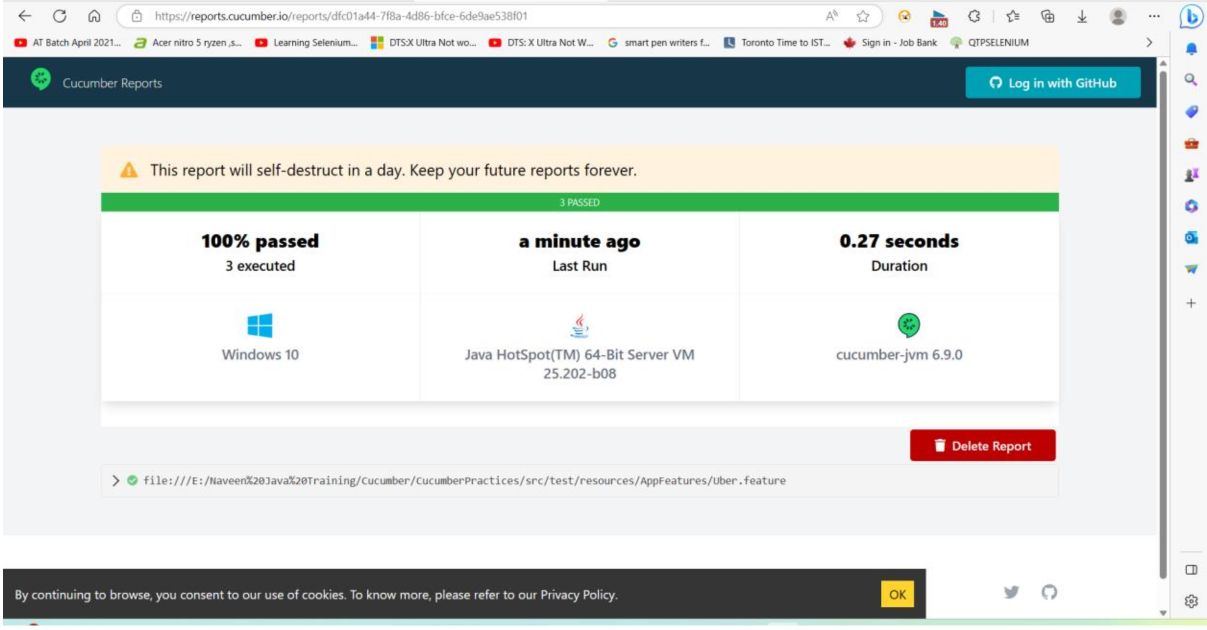
Console:

Now it says this message in console.



```
step 5 100
  Then user pays 100 us dollars                                      # Ste
(java.lang.Integer)
logout from the application
┌──────────────────────────────────────────────────────────────────────────┐
│ View your Cucumber Report at:                                            │
│ https://reports.cucumber.io/reports/dfc01a44-7f8a-4d86-bfce-6de9ae538f01 │
│                                                                          │
│ This report will self-destruct in 24h.                                   │
│ Keep reports forever: https://reports.cucumber.io/profile                │
└──────────────────────────────────────────────────────────────────────────┘
```

Open url and see the beautiful report:



This report will self-destruct in a day. Keep your future reports forever.

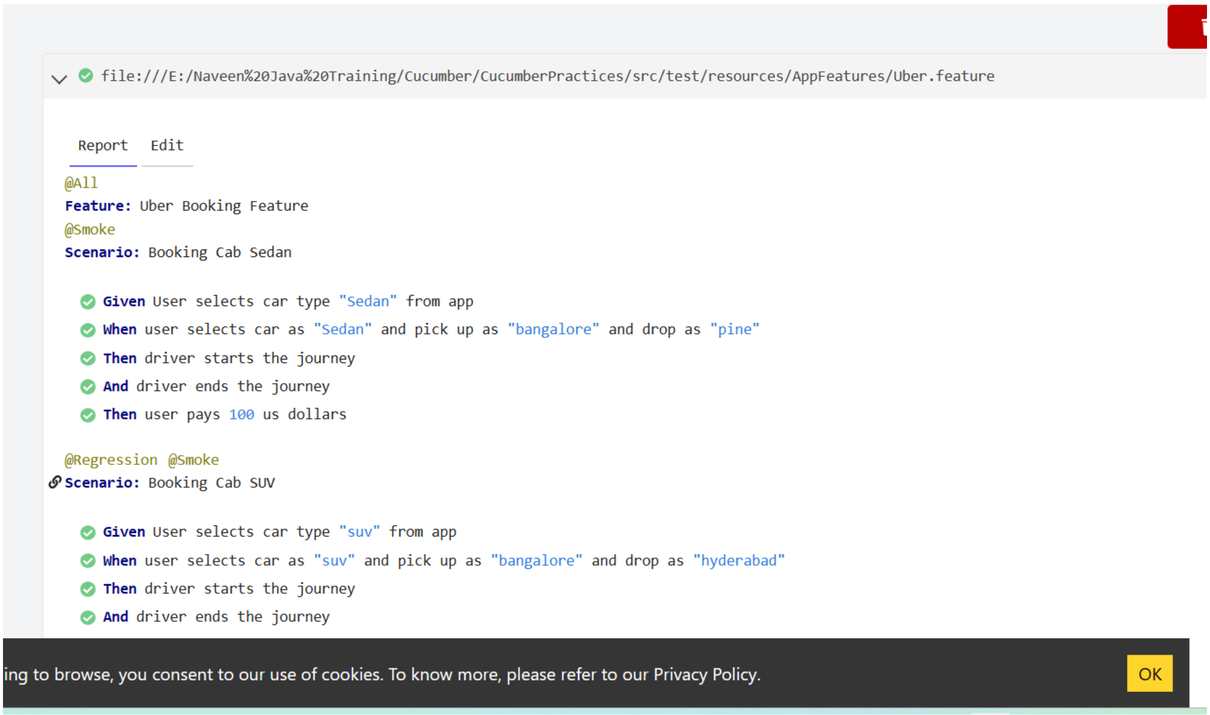
3 PASSED

100% passed	a minute ago	0.27 seconds
3 executed	Last Run	Duration
Windows 10	Java HotSpot(TM) 64-Bit Server VM 25.202-b08	cucumber-jvm 6.9.0

file:///E:/Naveen%20Java%20Training/Cucumber/CucumberPractices/src/test/resources/AppFeatures/Uber.feature

By continuing to browse, you consent to our use of cookies. To know more, please refer to our Privacy Policy.

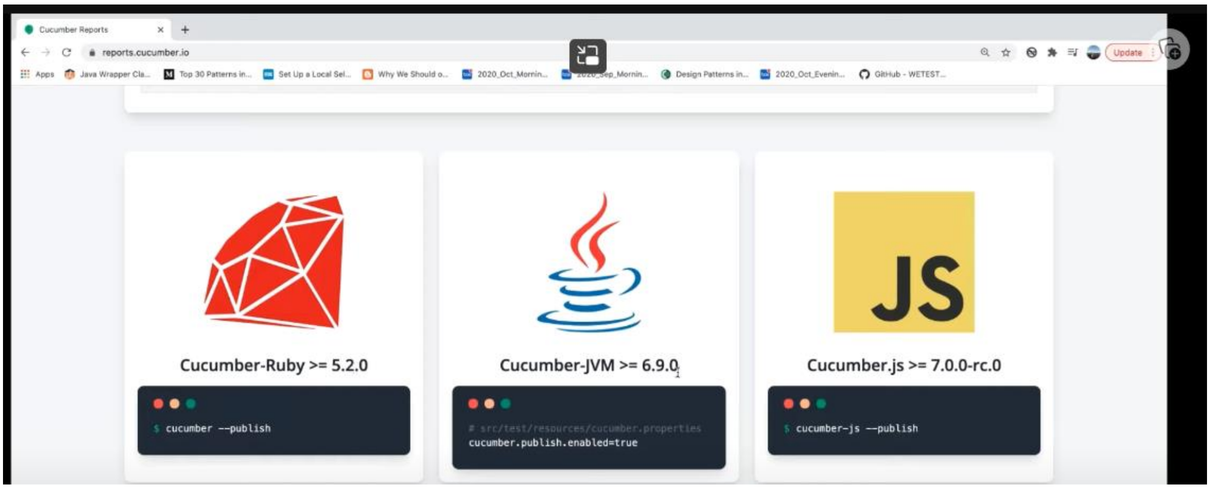
To see the feature file, expand the small arrow:


file:///E:/Naveen%20Java%20Training/Cucumber/CucumberPractices/src/test/resources/AppFeatures/Uber.feature

```
@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
```

ing to browse, you consent to our use of cookies. To know more, please refer to our Privacy Policy.

See the cucumber versions needed when you are working with different languages to publish the report-



Cucumber-Ruby >= 5.2.0

```
$ cucumber --publish
```

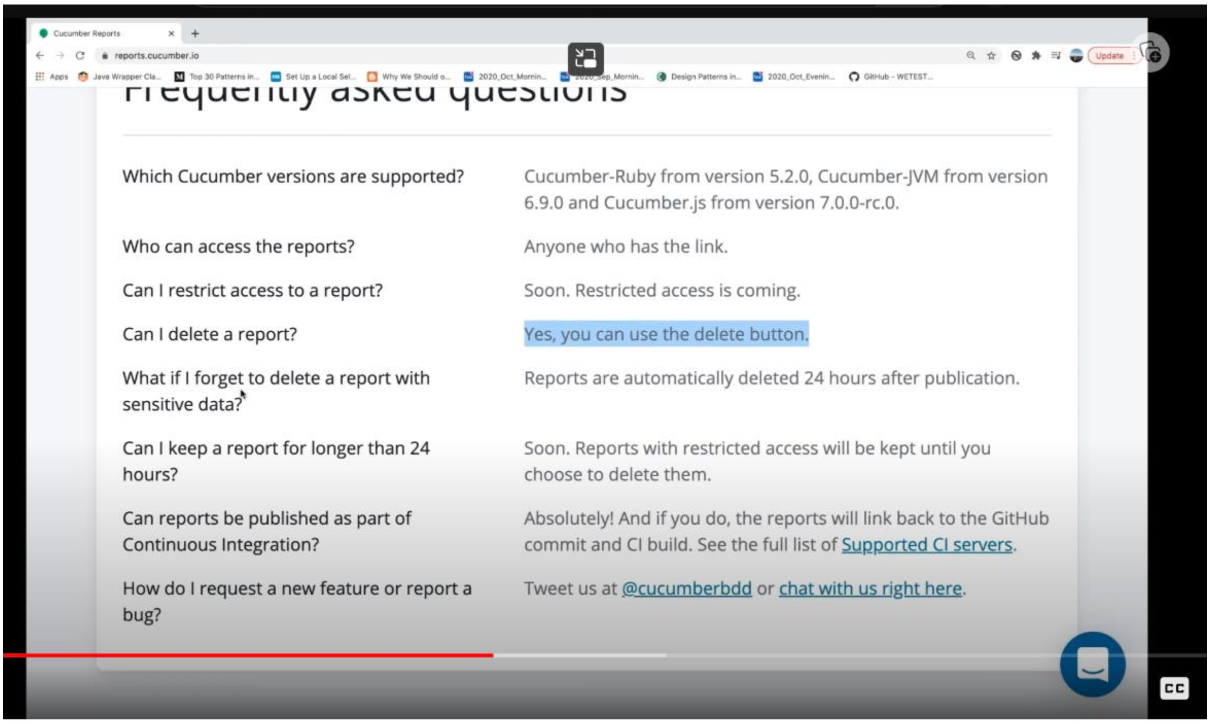
Cucumber-JVM >= 6.9.0

```
# src/test/resources/cucumber.properties
cucumber.publish.enabled=true
```

Cucumber.js >= 7.0.0-rc.0

```
$ cucumber-js --publish
```

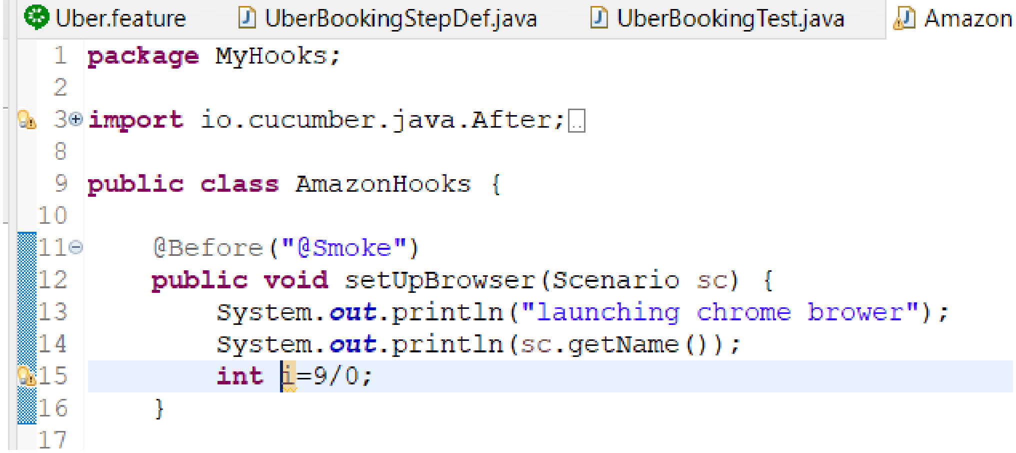
FAQs-



Frequently asked questions

Which Cucumber versions are supported?	Cucumber-Ruby from version 5.2.0, Cucumber-JVM from version 6.9.0 and Cucumber.js from version 7.0.0-rc.0.
Who can access the reports?	Anyone who has the link.
Can I restrict access to a report?	Soon. Restricted access is coming.
Can I delete a report?	Yes, you can use the delete button.
What if I forget to delete a report with sensitive data?	Reports are automatically deleted 24 hours after publication.
Can I keep a report for longer than 24 hours?	Soon. Reports with restricted access will be kept until you choose to delete them.
Can reports be published as part of Continuous Integration?	Absolutely! And if you do, the reports will link back to the GitHub commit and CI build. See the full list of Supported CI servers.
How do I request a new feature or report a bug?	Tweet us at @cucumberbdd or chat with us right here.

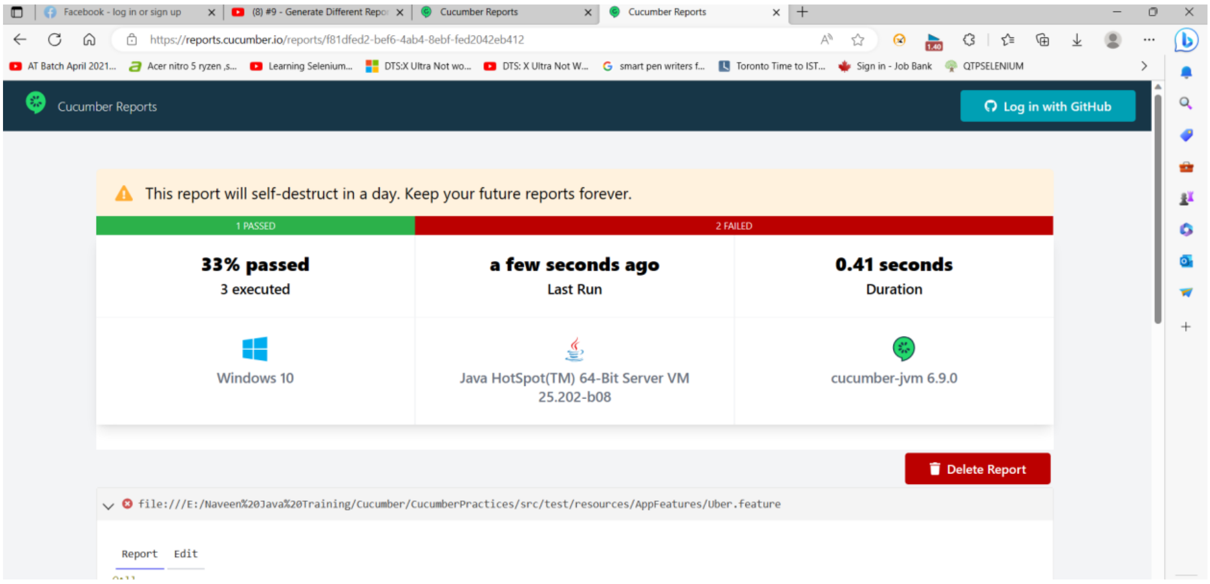
Purposely failed a before hook method-



```java
package MyHooks;

import io.cucumber.java.After;

public class AmazonHooks {

    @Before("@Smoke")
    public void setUpBrowser(Scenario sc) {
        System.out.println("launching chrome brower");
        System.out.println(sc.getName());
        int i=9/0;
    }

```

Run the runner:

Open report.



This report will self-destruct in a day. Keep your future reports forever.

1 PASSED | 2 FAILED

33% passed
3 executed

a few seconds ago
Last Run

0.41 seconds
Duration

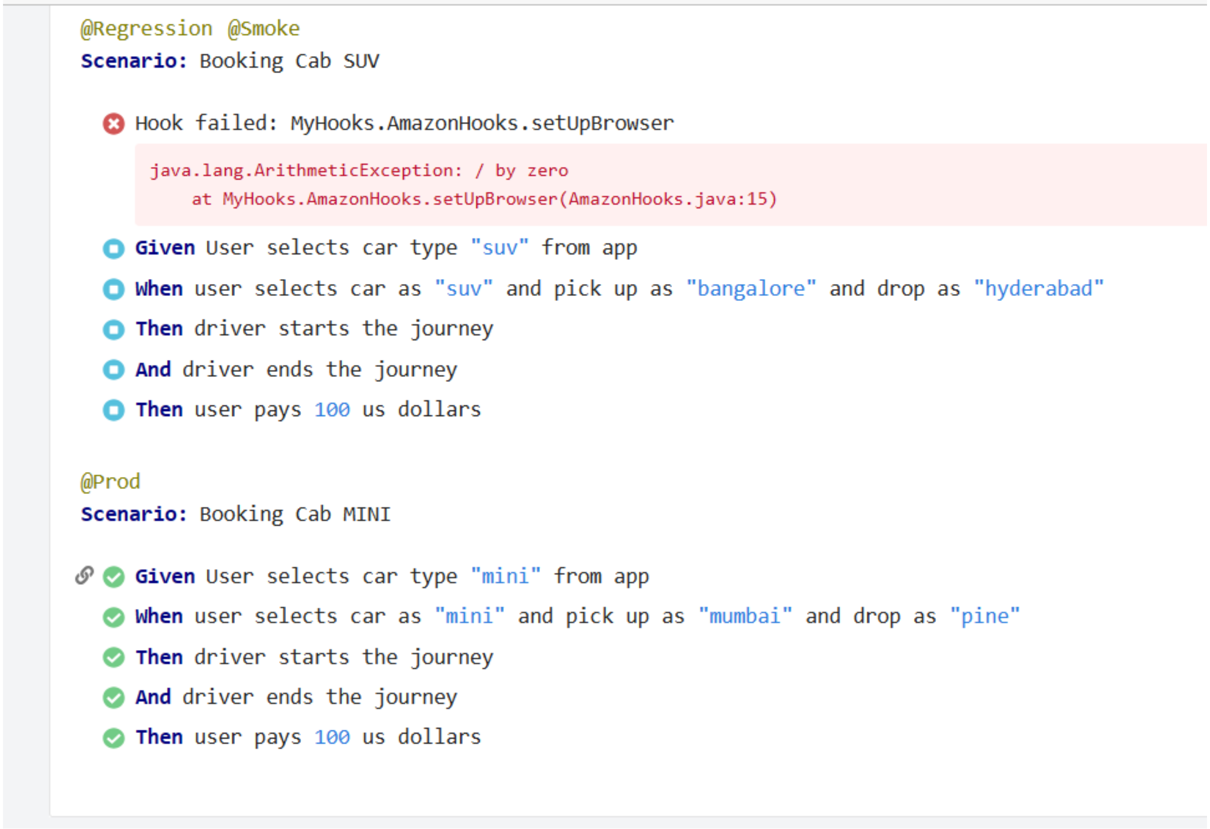
Windows 10

Java HotSpot(TM) 64-Bit Server VM
25.202-b08

cucumber-jvm 6.9.0

file:///E:/Naveen%20Java%20Training/Cucumber/CucumberPractices/src/test/resources/AppFeatures/Uber.feature

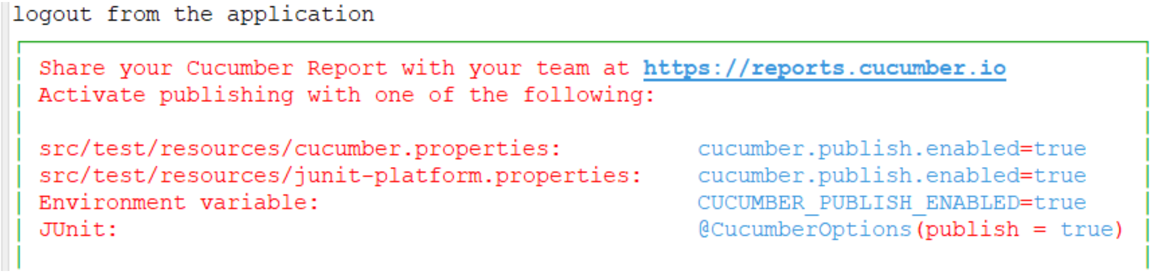
Wherever the “before hook” was used those scenarios failed, rest of the scenarios where “before” was not used they are passed. These reports get auto deleted after a day.



```
@Regression @Smoke
Scenario: Booking Cab SUV

  ✗ Hook failed: MyHooks.AmazonHooks.setUpBrowser
    java.lang.ArithmeticException: / by zero
        at MyHooks.AmazonHooks.setUpBrowser(AmazonHooks.java:15)
  ○ 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
```

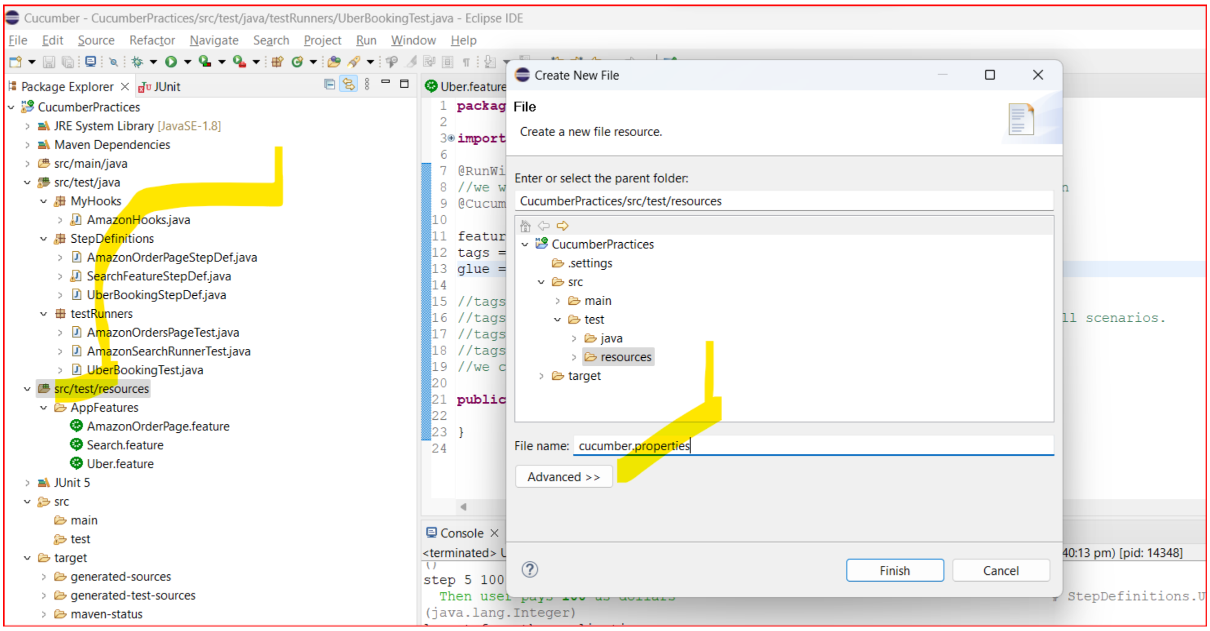
Another way to publish the reports-

We used the fourth option earlier. Now lets go with the first option.



```
logout from the application
┌──────────────────────────────────────────────────────────────────────────────────────┐
│ 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)    │
│                                                                                      │
```

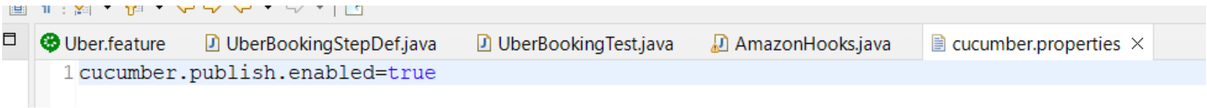
Create properties file.



Enter or select the parent folder: CucumberPractices/src/test/resources

File name: cucumber.properties

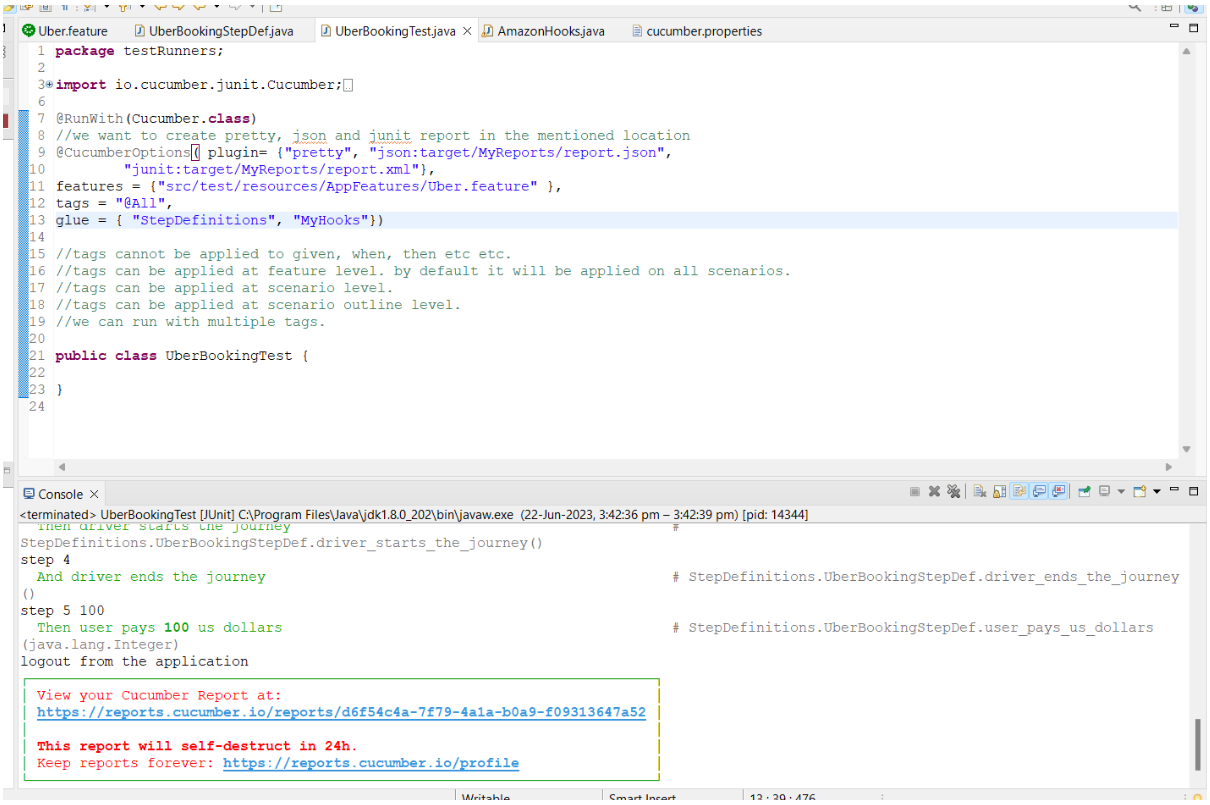
Mention this property and save it.



```
cucumber.publish.enabled=true
```

Run feature file.

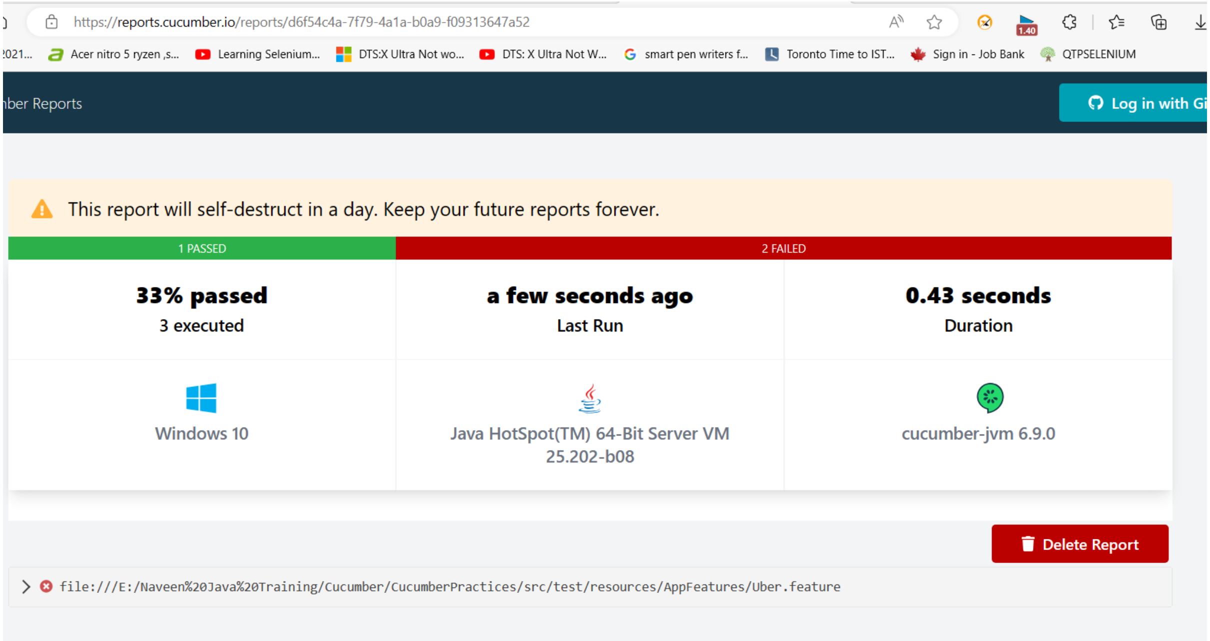
It ran -> go to the file path.



```
package testRunners;

import io.cucumber.junit.Cucumber;

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report in the mentioned location
@CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",
        "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.feature" },
tags = "@All",
glue = { "StepDefinitions", "MyHooks"})

//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 {

}
```

```
<terminated> UberBookingTest [JUnit] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe  (22-Jun-2023, 3:42:36 pm – 3:42:39 pm) [pid: 14344]
  Then driver starts the journey                              #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
step 4
  And driver ends the journey                                 # StepDefinitions.UberBookingStepDef.driver_ends_the_journey
()
step 5 100
  Then user pays 100 us dollars                               # StepDefinitions.UberBookingStepDef.user_pays_us_dollars
(java.lang.Integer)
logout from the application

┌───────────────────────────────────────────────────────────────────────────┐
│ View your Cucumber Report at:                                             │
│ https://reports.cucumber.io/reports/d6f54c4a-7f79-4a1a-b0a9-f09313647a52  │
│                                                                           │
│ This report will self-destruct in 24h.                                    │
│ Keep reports forever: https://reports.cucumber.io/profile                 │
└───────────────────────────────────────────────────────────────────────────┘
```

Report generated.



This report will self-destruct in a day. Keep your future reports forever.

1 PASSED

2 FAILED

33% passed
3 executed

a few seconds ago
Last Run

0.43 seconds
Duration

Windows 10

Java HotSpot(TM) 64-Bit Server VM
25.202-b08

cucumber-jvm 6.9.0

```
file:///E:/Naveen%20Java%20Training/Cucumber/CucumberPractices/src/test/resources/AppFeatures/Uber.feature
```

Good coding practice is to maintain the property in separate file like above.

Monochrome-

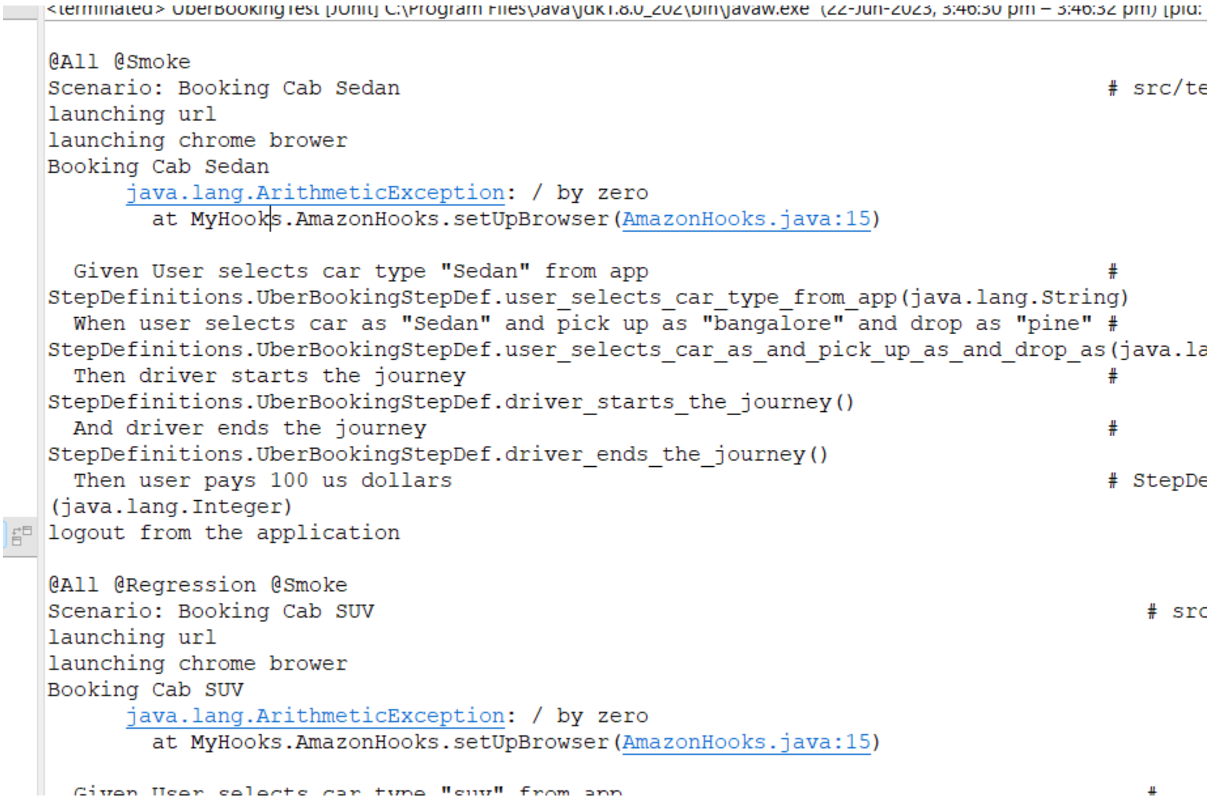
Alignment not proper in console or console messed up.

This is for console formatting.

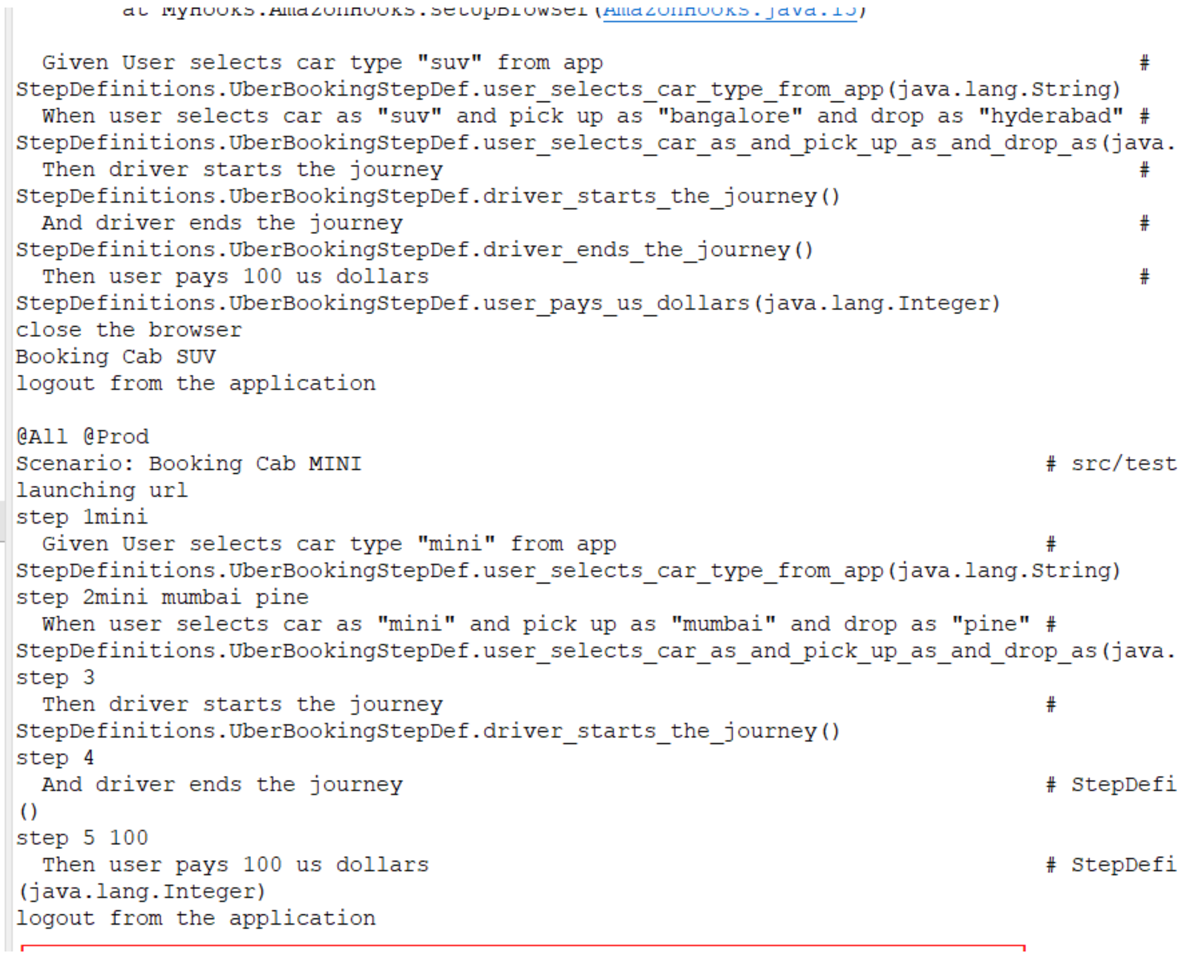
Runner file:

```
package testRunners;

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

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report in the mentioned location
@CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",
                   "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.feature" },
tags = "@All",
glue = { "StepDefinitions", "MyHooks"},
monochrome=true)

//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 {

}
```

Console :



```
<terminated> UberBookingTest [JUnit] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe (22-Jun-2023, 3:46:30 pm – 3:46:32 pm) [pid:

@All @Smoke
Scenario: Booking Cab Sedan                                                                   # src/te
launching url
launching chrome brower
Booking Cab Sedan
      java.lang.ArithmeticException: / by zero
        at MyHooks.AmazonHooks.setUpBrowser(AmazonHooks.java:15)

  Given User selects car type "Sedan" from app                                               #
StepDefinitions.UberBookingStepDef.user_selects_car_type_from_app(java.lang.String)
  When user selects car as "Sedan" and pick up as "bangalore" and drop as "pine" #
StepDefinitions.UberBookingStepDef.user_selects_car_as_and_pick_up_as_and_drop_as(java.la
  Then driver starts the journey                                                             #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
  And driver ends the journey                                                                #
StepDefinitions.UberBookingStepDef.driver_ends_the_journey()
  Then user pays 100 us dollars                                                              # StepDe
(java.lang.Integer)
logout from the application

@All @Regression @Smoke
Scenario: Booking Cab SUV                                                                       # src
launching url
launching chrome brower
Booking Cab SUV
      java.lang.ArithmeticException: / by zero
        at MyHooks.AmazonHooks.setUpBrowser(AmazonHooks.java:15)

  Given User selects car type "suv" from app                                                   #
```



```
        at MyHooks.AmazonHooks.setUpBrowser(AmazonHooks.java:15)

  Given User selects car type "suv" from app                                                   #
StepDefinitions.UberBookingStepDef.user_selects_car_type_from_app(java.lang.String)
  When user selects car as "suv" and pick up as "bangalore" and drop as "hyderabad" #
StepDefinitions.UberBookingStepDef.user_selects_car_as_and_pick_up_as_and_drop_as(java.
  Then driver starts the journey                                                               #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
  And driver ends the journey                                                                  #
StepDefinitions.UberBookingStepDef.driver_ends_the_journey()
  Then user pays 100 us dollars                                                                #
StepDefinitions.UberBookingStepDef.user_pays_us_dollars(java.lang.Integer)
close the browser
Booking Cab SUV
logout from the application

@All @Prod
Scenario: Booking Cab MINI                                                             # src/test
launching url
step 1mini
  Given User selects car type "mini" from app                                          #
StepDefinitions.UberBookingStepDef.user_selects_car_type_from_app(java.lang.String)
step 2mini mumbai pine
  When user selects car as "mini" and pick up as "mumbai" and drop as "pine" #
StepDefinitions.UberBookingStepDef.user_selects_car_as_and_pick_up_as_and_drop_as(java.
step 3
  Then driver starts the journey                                                       #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
step 4
  And driver ends the journey                                                          # StepDefi
()
step 5 100
  Then user pays 100 us dollars                                                        # StepDefi
(java.lang.Integer)
logout from the application
```

Now monochrome is not used much.

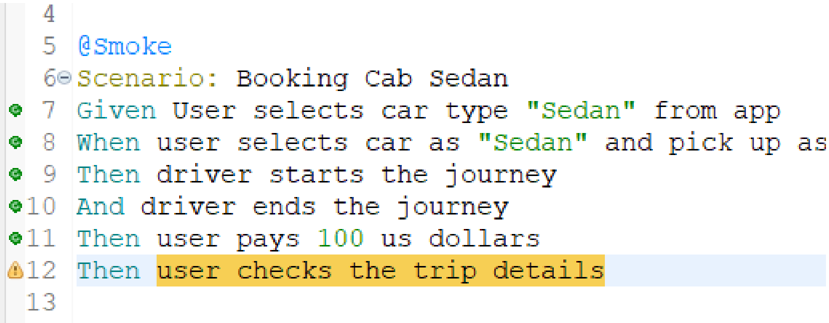
Cucumber 6 and above is already giving color and better option.

Dry run-

Runner file:

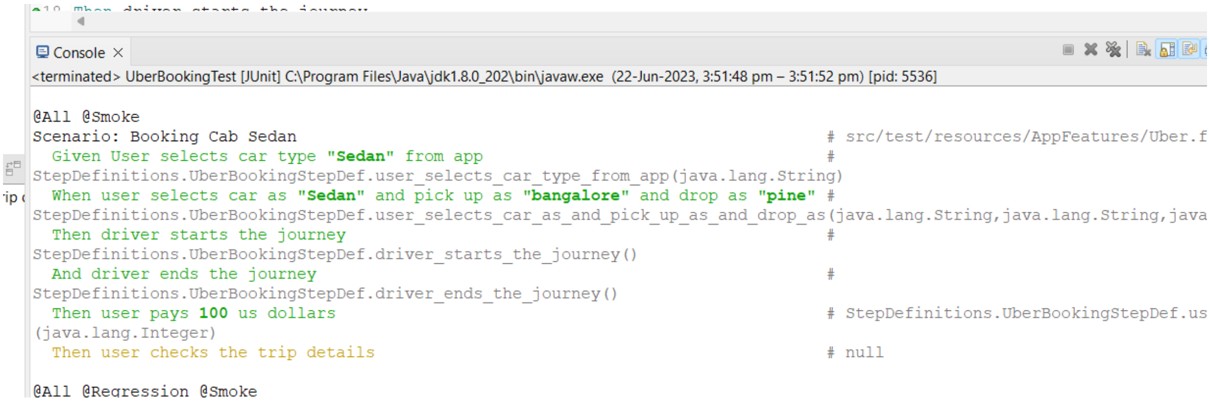
```
package testRunners;

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

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report in the mentioned location
@CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",
                          "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.feature" },
tags = "@All",
glue = { "StepDefinitions", "MyHooks"},
monochrome=false,
dryRun =true
)

//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 {

}
```

We added this one step in feature file:



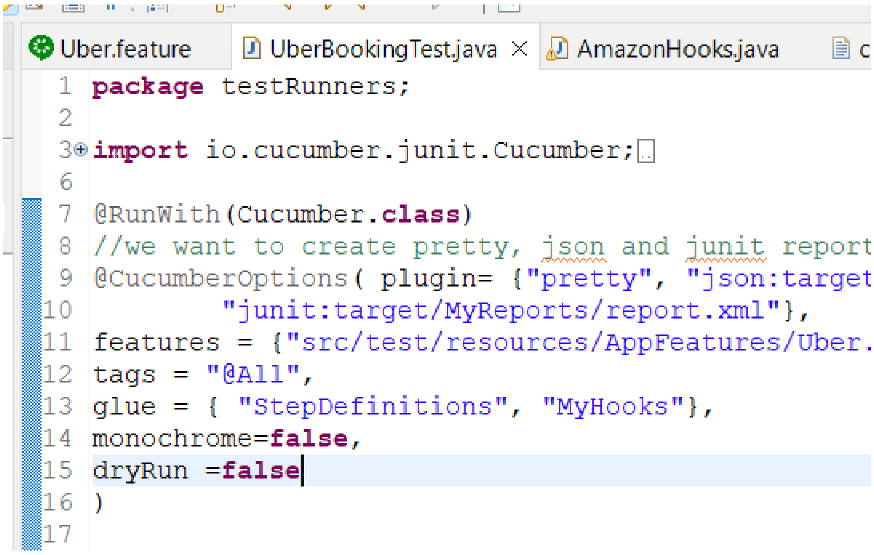
```

@Smoke
Scenario: Booking Cab Sedan
Given User selects car type "Sedan" from app
When user selects car as "Sedan" and pick up as
Then driver starts the journey
And driver ends the journey
Then user pays 100 us dollars
Then user checks the trip details

```

Now when we run the runner it will not give error, but dry run will point this step to null indicating there are no step definitions to the line.



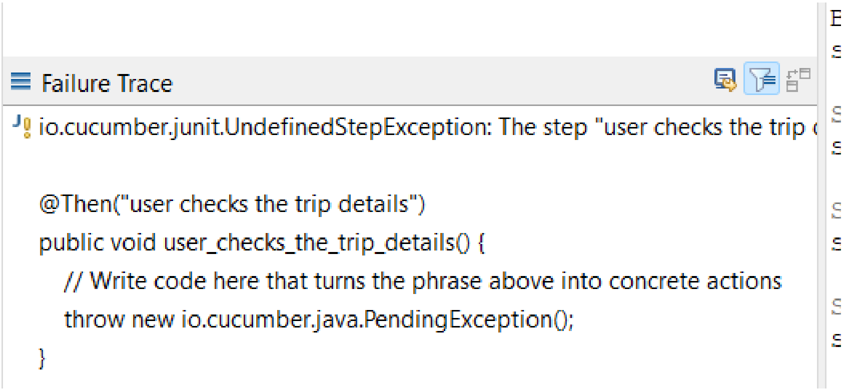
```
<terminated> UberBookingTest [JUnit] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe  (22-Jun-2023, 3:51:48 pm – 3:51:52 pm) [pid: 5536]

@All @Smoke
Scenario: Booking Cab Sedan                                                     # src/test/resources/AppFeatures/Uber.f
  Given User selects car type "Sedan" from app                                  #
StepDefinitions.UberBookingStepDef.user_selects_car_type_from_app(java.lang.String)
  When user selects car as "Sedan" and pick up as "bangalore" and drop as "pine" #
StepDefinitions.UberBookingStepDef.user_selects_car_as_and_pick_up_as_and_drop_as(java.lang.String,java.lang.String,java
  Then driver starts the journey                                                #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
  And driver ends the journey                                                   #
StepDefinitions.UberBookingStepDef.driver_ends_the_journey()
  Then user pays 100 us dollars                                                 # StepDefinitions.UberBookingStepDef.us
(java.lang.Integer)
  Then user checks the trip details                                             # null

@All @Regression @Smoke
```

When you make dry run as false, nothing changes except in left hand side we get the step to implement-

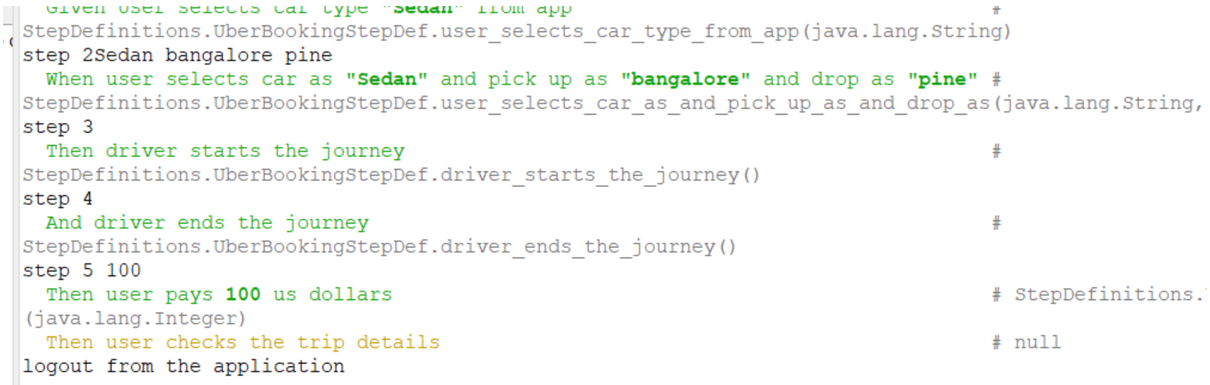


```
package testRunners;

import io.cucumber.junit.Cucumber;

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report
@CucumberOptions( plugin= {"pretty", "json:target
          "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.
tags = "@All",
glue = { "StepDefinitions", "MyHooks"},
monochrome=false,
dryRun =false
)
```

Console:



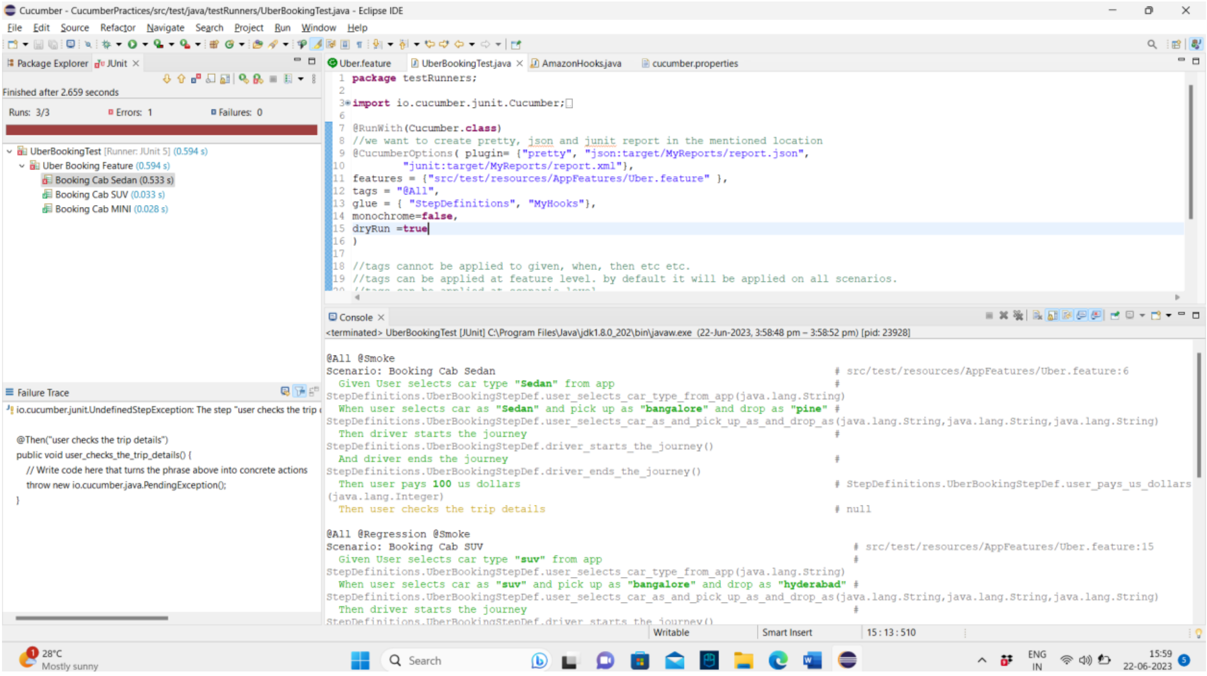
```
io.cucumber.junit.UndefinedStepException: The step "user checks the trip 

@Then("user checks the trip details")
public void user_checks_the_trip_details() {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}
```



```
  Given User selects car type "Sedan" from app                                  #
StepDefinitions.UberBookingStepDef.user_selects_car_type_from_app(java.lang.String)
step 2Sedan bangalore pine
  When user selects car as "Sedan" and pick up as "bangalore" and drop as "pine" #
StepDefinitions.UberBookingStepDef.user_selects_car_as_and_pick_up_as_and_drop_as(java.lang.String,
step 3
  Then driver starts the journey                                                #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
step 4
  And driver ends the journey                                                   #
StepDefinitions.UberBookingStepDef.driver_ends_the_journey()
step 5 100
  Then user pays 100 us dollars                                                 # StepDefinitions.
(java.lang.Integer)
  Then user checks the trip details                                             # null
logout from the application
```

With true also the same thing is applicable-

Dry run is depreciated seems like.



Cucumber - CucumberPractices/src/test/java/testRunners/UberBookingTest.java - Eclipse IDE

Finished after 2.659 seconds

Runs: 3/3 Errors: 1 Failures: 0

```
UberBookingTest [Runner: JUnit 5] (0.594 s)
  Uber Booking Feature (0.594 s)
    Booking Cab Sedan (0.533 s)
    Booking Cab SUV (0.033 s)
    Booking Cab MINI (0.028 s)
```

Failure Trace

```
io.cucumber.junit.UndefinedStepException: The step "user checks the trip

@Then("user checks the trip details")
public void user_checks_the_trip_details() {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
}
```

```java
package testRunners;

import io.cucumber.junit.Cucumber;

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report in the mentioned location
@CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",
        "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.feature" },
tags = "@All",
glue = { "StepDefinitions", "MyHooks"},
monochrome=false,
dryRun =true
)

//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.
```

Console

<terminated> UberBookingTest [JUnit] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe (22-Jun-2023, 3:58:48 pm – 3:58:52 pm) [pid: 23928]

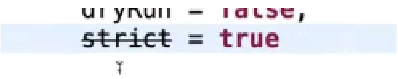
```
@All @Smoke
Scenario: Booking Cab Sedan                                                    # src/test/resources/AppFeatures/Uber.feature:6
  Given User selects car type "Sedan" from app                                 #
StepDefinitions.UberBookingStepDef.user_selects_car_type_from_app(java.lang.String)
  When user selects car as "Sedan" and pick up as "bangalore" and drop as "pine" #
StepDefinitions.UberBookingStepDef.user_selects_car_as_and_pick_up_as_and_drop_as(java.lang.String,java.lang.String,java.lang.String)
  Then driver starts the journey                                               #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
  And driver ends the journey                                                  #
StepDefinitions.UberBookingStepDef.driver_ends_the_journey()
  Then user pays 100 us dollars                                                # StepDefinitions.UberBookingStepDef.user_pays_us_dollars
(java.lang.Integer)
  Then user checks the trip details                                            # null

@All @Regression @Smoke
Scenario: Booking Cab SUV                                                         # src/test/resources/AppFeatures/Uber.feature:15
  Given User selects car type "suv" from app                                      #
StepDefinitions.UberBookingStepDef.user_selects_car_type_from_app(java.lang.String)
  When user selects car as "suv" and pick up as "bangalore" and drop as "hyderabad" #
StepDefinitions.UberBookingStepDef.user_selects_car_as_and_pick_up_as_and_drop_as(java.lang.String,java.lang.String,java.lang.String)
  Then driver starts the journey                                                  #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
```

Useful for sharing this runner file amongst teams as this mapping helps to track feature files with step definition files.

Strict-

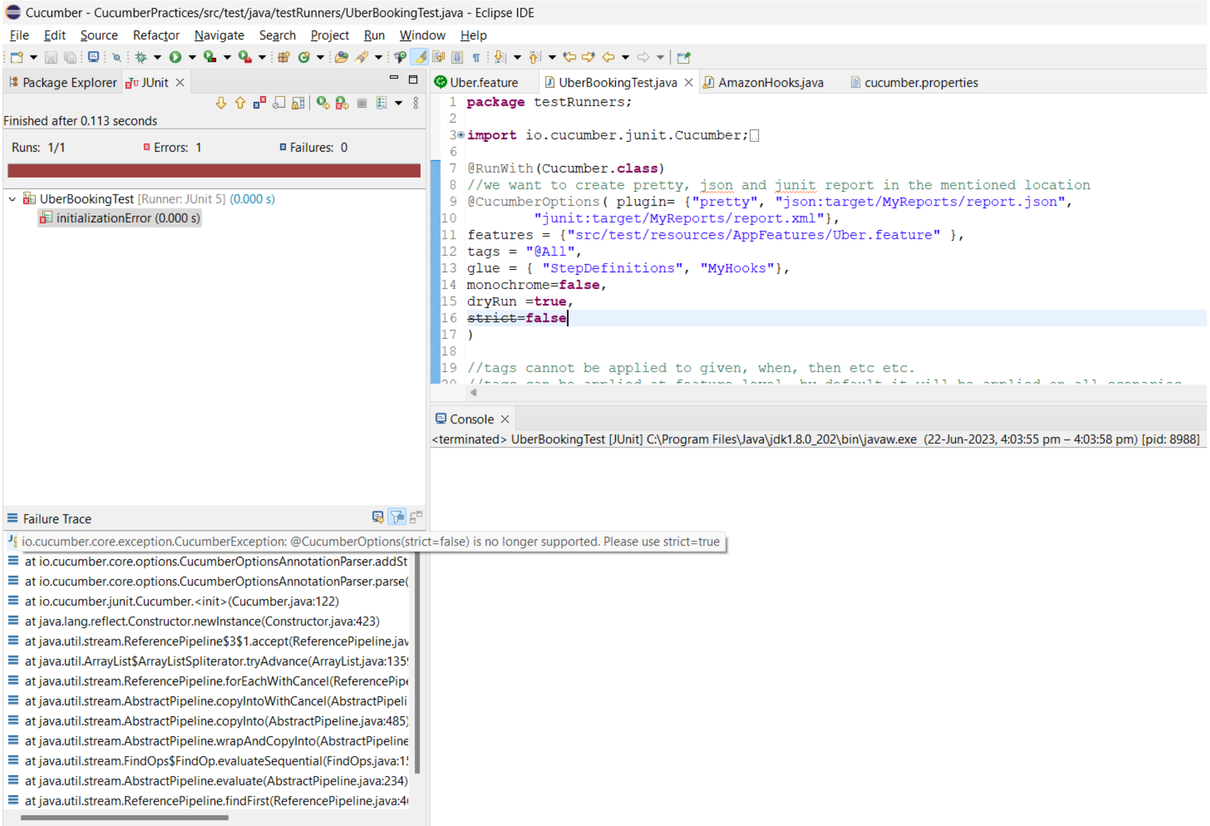
If you want to ignore the error, like for example, you wrote feature but no step definition then use strict=false.

But strict is removed now. We need to add in runner in cucumber options.



```
dryRun = false,
~~strict~~ = true
```

Strict=true means strictly follow the rules. Strict =false means no need to follow the rules.


Cucumber - CucumberPractices/src/test/java/testRunners/UberBookingTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
Uber.feature
UberBookingTest.java
AmazonHooks.java
cucumber.properties
Console
Failure Trace

Finished after 0.113 seconds

Runs: 1/1 Errors: 1 Failures: 0

UberBookingTest [Runner: JUnit 5] (0.000 s)
initializationError (0.000 s)

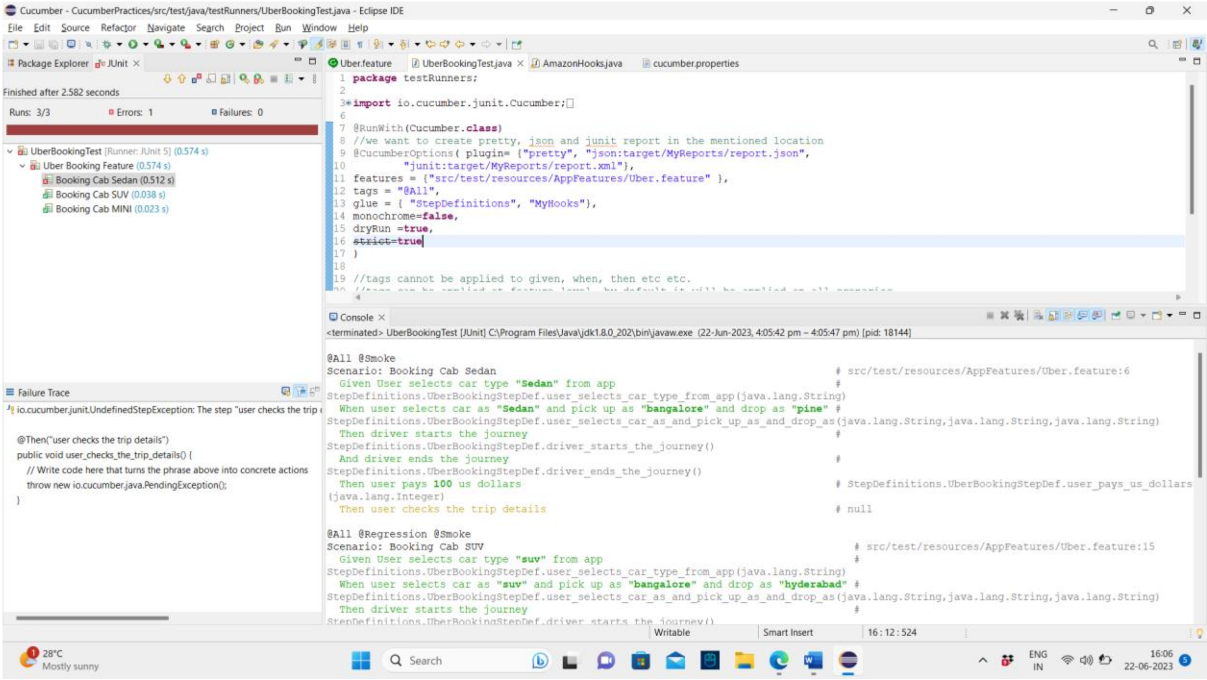
```
package testRunners;

import io.cucumber.junit.Cucumber;

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report in the mentioned location
@CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",
        "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.feature" },
tags = "@All",
glue = { "StepDefinitions", "MyHooks"},
monochrome=false,
dryRun =true,
~~strict~~=false
)

//tags cannot be applied to given, when, then etc etc.
```

```
<terminated> UberBookingTest [JUnit] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe (22-Jun-2023, 4:03:55 pm – 4:03:58 pm) [pid: 8988]
```

```
io.cucumber.core.exception.CucumberException: @CucumberOptions(strict=false) is no longer supported. Please use strict=true
at io.cucumber.core.options.CucumberOptionsAnnotationParser.addSt
at io.cucumber.core.options.CucumberOptionsAnnotationParser.parse(
at io.cucumber.junit.Cucumber.<init>(Cucumber.java:122)
at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
at java.util.stream.ReferencePipeline$3$1.accept(ReferencePipeline.jav
at java.util.ArrayList$ArrayListSpliterator.tryAdvance(ArrayList.java:135
at java.util.stream.ReferencePipeline.forEachWithCancel(ReferencePip
at java.util.stream.AbstractPipeline.copyIntoWithCancel(AbstractPipeli
at java.util.stream.AbstractPipeline.copyInto(AbstractPipeline.java:485)
at java.util.stream.AbstractPipeline.wrapAndCopyInto(AbstractPipeline
at java.util.stream.FindOps$FindOp.evaluateSequential(FindOps.java:1
at java.util.stream.AbstractPipeline.evaluate(AbstractPipeline.java:234)
at java.util.stream.ReferencePipeline.findFirst(ReferencePipeline.java:4
```

It gives failure trace that strict is out.

Use strict=true and it works regularly-



Finished after 2.582 seconds

Runs: 3/3 Errors: 1 Failures: 0

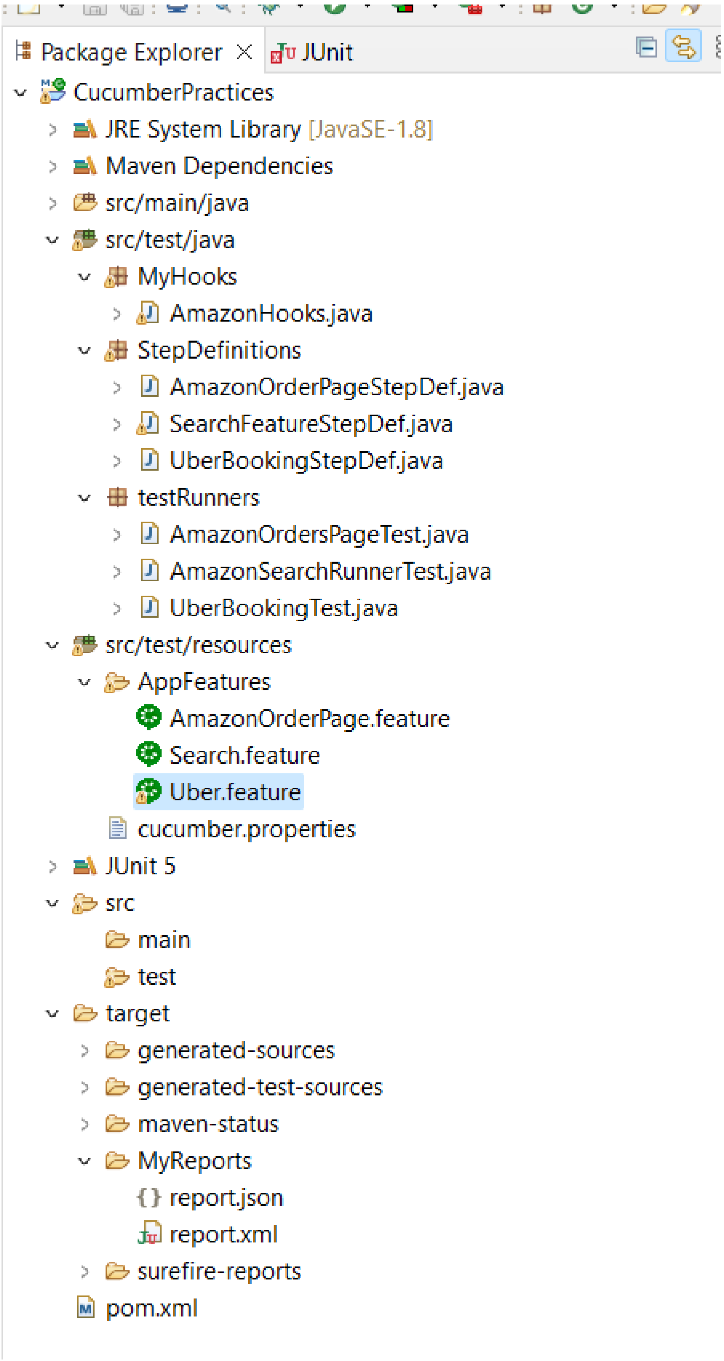
UberBookingTest [Runner: JUnit 5] (0.574 s)
Uber Booking Feature (0.574 s)
Booking Cab Sedan (0.512 s)
Booking Cab SUV (0.038 s)
Booking Cab MINI (0.023 s)

```
package testRunners;

import io.cucumber.junit.Cucumber;

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report in the mentioned location
@CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",
        "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.feature" },
tags = "@All",
glue = { "StepDefinitions", "MyHooks"},
monochrome=false,
dryRun =true,
~~strict~~=true
)

//tags cannot be applied to given, when, then etc etc.
```

```
io.cucumber.junit.UndefinedStepException: The step "user checks the trip

  @Then("user checks the trip details")
  public void user_checks_the_trip_details() {
    // Write code here that turns the phrase above into concrete actions
    throw new io.cucumber.java.PendingException();
  }
```

```
<terminated> UberBookingTest [JUnit] C:\Program Files\Java\jdk1.8.0_202\bin\javaw.exe (22-Jun-2023, 4:05:42 pm – 4:05:47 pm) [pid: 18144]

@All @Smoke
Scenario: Booking Cab Sedan                                              # src/test/resources/AppFeatures/Uber.feature:6
  Given User selects car type "Sedan" from app                           #
StepDefinitions.UberBookingStepDef.user_selects_car_type_from_app(java.lang.String)
  When user selects car as "Sedan" and pick up as "bangalore" and drop as "pine" #
StepDefinitions.UberBookingStepDef.user_selects_car_as_and_pick_up_as_and_drop_as(java.lang.String,java.lang.String,java.lang.String)
  Then driver starts the journey                                         #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
  And driver ends the journey                                            #
StepDefinitions.UberBookingStepDef.driver_ends_the_journey()
  Then user pays 100 us dollars                                          # StepDefinitions.UberBookingStepDef.user_pays_us_dollars
(java.lang.Integer)
  Then user checks the trip details                                      # null

@All @Regression @Smoke
Scenario: Booking Cab SUV                                                  # src/test/resources/AppFeatures/Uber.feature:15
  Given User selects car type "suv" from app                               #
StepDefinitions.UberBookingStepDef.user_selects_car_type_from_app(java.lang.String)
  When user selects car as "suv" and pick up as "bangalore" and drop as "hyderabad" #
StepDefinitions.UberBookingStepDef.user_selects_car_as_and_pick_up_as_and_drop_as(java.lang.String,java.lang.String,java.lang.String)
  Then driver starts the journey                                           #
StepDefinitions.UberBookingStepDef.driver_starts_the_journey()
```

Project structure-



```
CucumberPractices
  JRE System Library [JavaSE-1.8]
  Maven Dependencies
  src/main/java
  src/test/java
    MyHooks
      AmazonHooks.java
    StepDefinitions
      AmazonOrderPageStepDef.java
      SearchFeatureStepDef.java
      UberBookingStepDef.java
    testRunners
      AmazonOrdersPageTest.java
      AmazonSearchRunnerTest.java
      UberBookingTest.java
  src/test/resources
    AppFeatures
      AmazonOrderPage.feature
      Search.feature
      Uber.feature
    cucumber.properties
  JUnit 5
  src
    main
    test
  target
    generated-sources
    generated-test-sources
    maven-status
    MyReports
      report.json
      report.xml
    surefire-reports
  pom.xml
```

Codes used-

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
#Then user checks the trip details

@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.

```
package testRunners;

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

@RunWith(Cucumber.class)
//we want to create pretty, json and junit report in the mentioned location
@CucumberOptions( plugin= {"pretty", "json:target/MyReports/report.json",
                    "junit:target/MyReports/report.xml"},
features = {"src/test/resources/AppFeatures/Uber.feature" },
tags = "@All",
glue = { "StepDefinitions", "MyHooks"},
monochrome=false,
dryRun =true,
strict=true
)

//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 {

}
```

Hooks.

```
package MyHooks;

import io.cucumber.java.After;
import io.cucumber.java.AfterStep;
import io.cucumber.java.Before;
import io.cucumber.java.BeforeStep;
import io.cucumber.java.Scenario;

public class AmazonHooks {

        @Before("@Smoke")
        public void setUpBrowser(Scenario sc) {
                System.out.println("launching chrome brower");
                System.out.println(sc.getName());
                //int i=9/0;
        }

        @Before(order = 2)
        public void setUpURL() {
                System.out.println("launching url");
        }

        @After("@Regression")
        public void tearDownClose(Scenario sc) {
                System.out.println("close the browser");
                System.out.println(sc.getName());
        }

        @After(order = 2)
        public void tearDownLogout() {
                System.out.println("logout from the application");
        }

//      @BeforeStep
//      public void takeScreenshot() {
//              System.out.println("takes screenshot after every step");
//      }
//
//      @AfterStep
//      public void refreshBrowser() {
//              System.out.println("refresh the browser after every step");
//      }
}
```

Cucumber.properties.

```
cucumber.publish.enabled=true
```