Project Description:

This project is developed using the follow tech stack

- Code written in Java programming language.
- The Selenium web driver is used for the browser automation.
- Cucumber Framework is used to achieve the Behaviour driven development (BDD) approach.
- Apache Maven is used for project building process and dependency management.
- The Page Object Model strategy is used to manage the web page elements in the separate class.
- JUnit framework is used for test validation checks.
- Data driven approach is used to avoid the hardcoding.

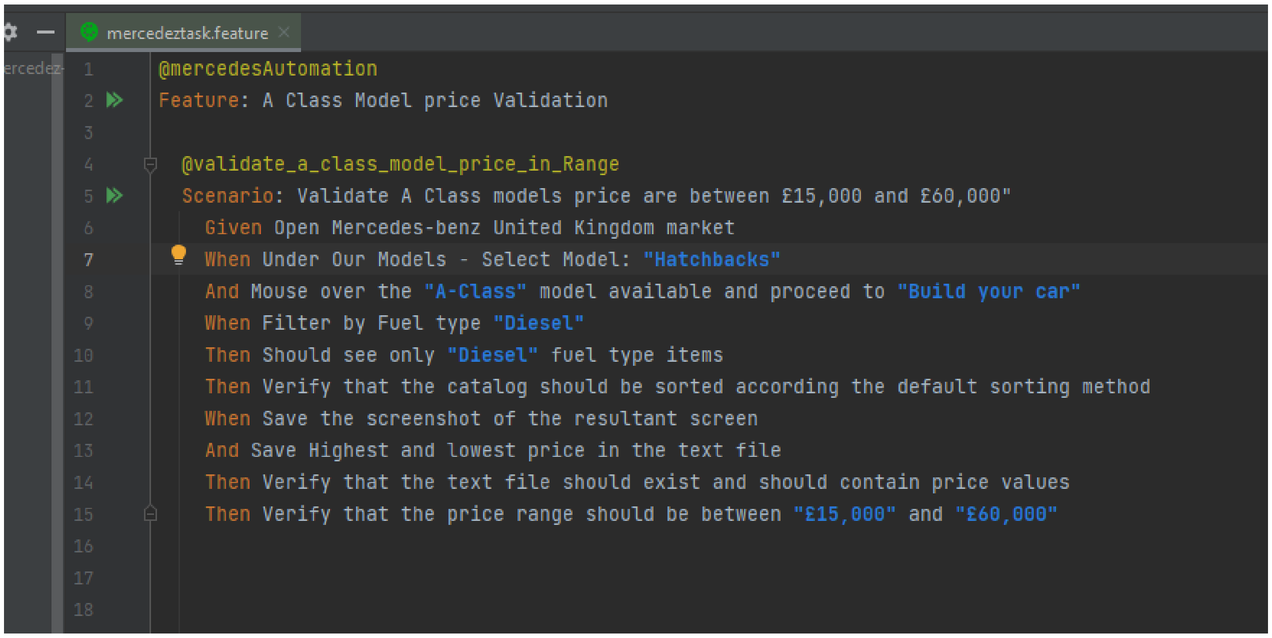
Execution capabilities:

a. This project has been configured to run on Chrome and MS Edge web browsers.
b. Right now, it only able run-on Windows OS. However, if little modification is made with web driver, then it will be able to run on Linux and Mac OS as well.

Framework Structure:

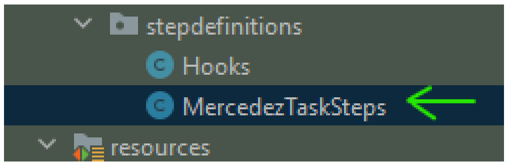
The Feature File

The First file in the cucumber framework is the “**feature file”** which contains the all the Test scenarios which are written in Gherkin language statement (English) to enable the Behaviour driven development (BDD) approach.



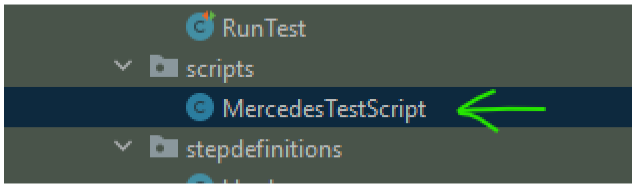
```
@mercedesAutomation
Feature: A Class Model price Validation

  @validate_a_class_model_price_in_Range
  Scenario: Validate A Class models price are between £15,000 and £60,000"
    Given Open Mercedes-benz United Kingdom market
    When Under Our Models - Select Model: "Hatchbacks"
    And Mouse over the "A-Class" model available and proceed to "Build your car"
    When Filter by Fuel type "Diesel"
    Then Should see only "Diesel" fuel type items
    Then Verify that the catalog should be sorted according the default sorting method
    When Save the screenshot of the resultant screen
    And Save Highest and lowest price in the text file
    Then Verify that the text file should exist and should contain price values
    Then Verify that the price range should be between "£15,000" and "£60,000"
```

Step definition java class:
In Order to connect the feature file with code script, the step definition file used for that as it contains the blank method which mapped with each cucumber statement. These step definition methods are used to call the methods defined in the coding script.
The validation checks are also managed in that file.



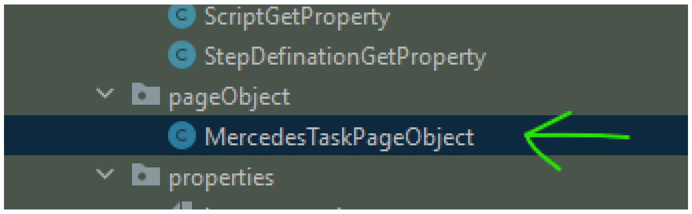
The script class:

This file contains the main code which is required for the developing the business logic for the test automation.



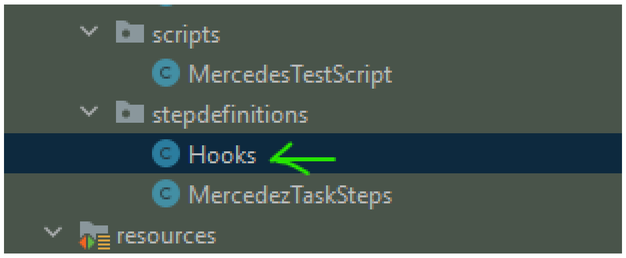
The PageObject java Class:

This file contains the all the UI web elements which is required by selenium web driver for performing the browser automation. This file is linked using the instanciation to the script class.



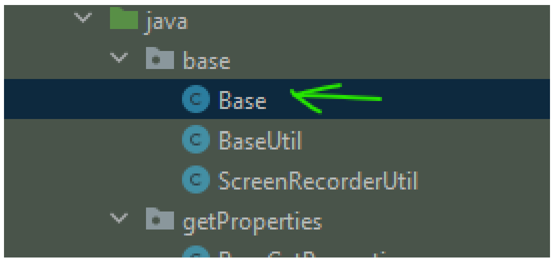
The Hooks java class:
This file contains the before and after method that are executed before and After test execution.
The before method is used to invoke the browser and start the video recording.
The After method is used to quit the browser stop the recording and add the attachments to the allure report:

- text file of prices
- image file of last web screen
- the video file of entire scenario



The base java class:

The base file contains the methods with their definition that may be repeatedly used in the code script. It is the super class of the script class.



The BaseUtil class:

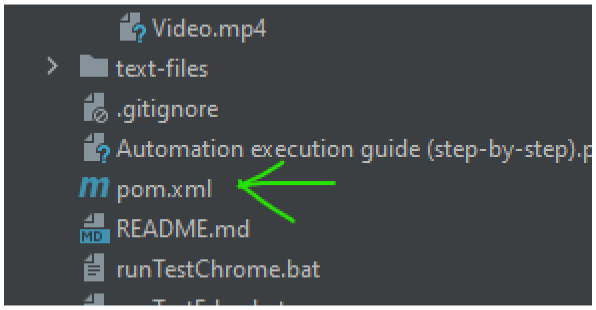
The file contains the web driver to use in the project

The screen recorder class.

This file contains the methods required to make the video recording for test scenarios.

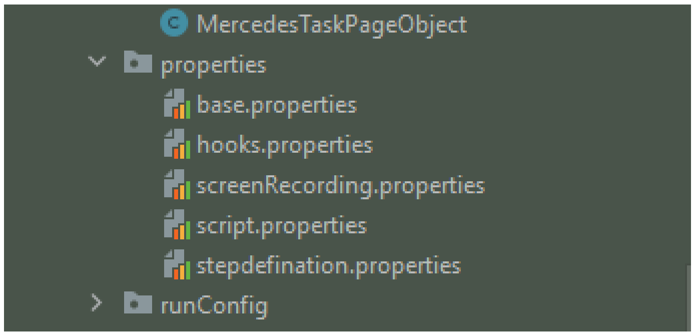
Pom.xml file

This file is managed by Maven build system to manage project dependencies and project build process, etc.



Properties files:

These files contain the data to be driven and used by java classes. These are used to avoid the hardcoding in the code.



The RunConfig java class:

This file is used to check the execution of the automation

<u>Testing Scenario flow:</u>

The test scenario is developed in the following steps:

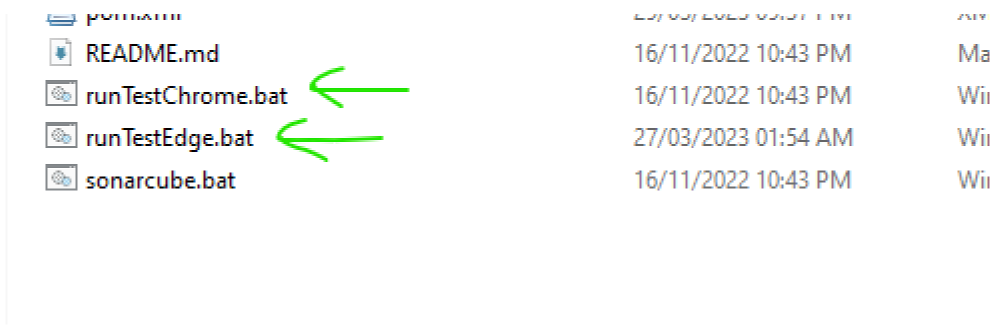
1. The Browser is navigated to the Mercedes-Benz United Kingdom market web page.
2. Scroll down to the Our model section on the page Select Hatchbacks model
    a. Validation 1: To verify that hatchbacks model is available in the list otherwise this test step will be failed.
3. Look for the class A car and hover the mouse over it and select build your car.
    a. Validation 2: To verify that class A car is present or not. If not, then this test step would be failed.
    b. Validation 3: To verify that “build your car” is present or not. If not, then this test step would be failed.
4. On the car configuration page, scroll down to the fuel type and select diesel.
    a. Validation 4: To verify that at least one item is available after selection of diesel. If not, then this test step will fail
5. Validation 5: To verify that all the available items must have the fuel type diesel.
6. Validation 6: To verify that all that the catalogue must be sorted in the default sorting method (Ascending price order)
7. Save the screenshot of the resultant screen
8. Validation 7: To verify that screenshot must be taken and saved.
9. Save the highest and lowest price in the text file.
10. Validation 8: To validate that text file must contain the values. If not, then step will be failed
11. Validation 9: To validate all the available prices must be between £15,000 and £60,000

How to execution the Automation Test:

There are total 3 different ways to execute the project you can choose any one of them to run the test but first 2 are recommended:

Method1: Execution of the “runTest.bat” file

In Windows, double click on the “runTestChrome.bat” or “runTestEdge.bat”



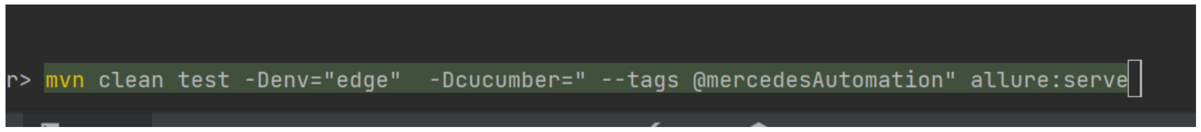
```
README.md            16/11/2022 10:43 PM   Ma
runTestChrome.bat    16/11/2022 10:43 PM   Wi
runTestEdge.bat      27/03/2023 01:54 AM   Wi
sonarcube.bat        16/11/2022 10:43 PM   Wi
```

Method 2: Execution by running the maven command directly in the command prompt

- Open Windows Command prompt or terminal:
- Go to directory where project is cloned or downloaded.
- Type the command: mvn clean test -Dcucumber=" --tags @mercedesAutomation" allure:serve (for chrome browser execution)

```
mvn clean test -Dcucumber=" --tags @mercedesAutomation" allure:serve
```

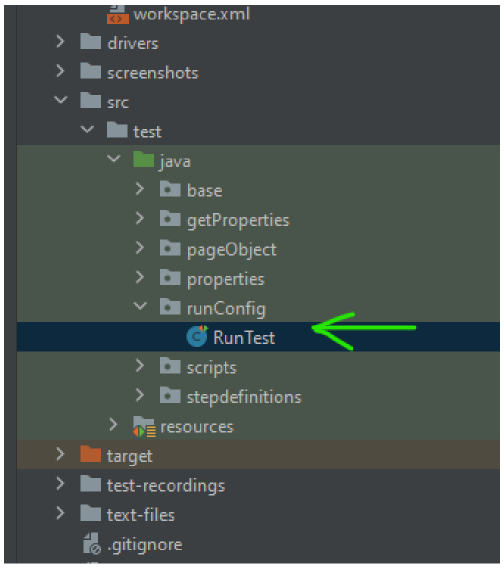
- mvn clean test -Denv="edge" -Dcucumber=" --tags @mercedesAutomation" allure:serve (For edge browser execution)



```
r> mvn clean test -Denv="edge"  -Dcucumber=" --tags @mercedesAutomation" allure:serve
```

Method 3: Execution by Running the RunTest.java file

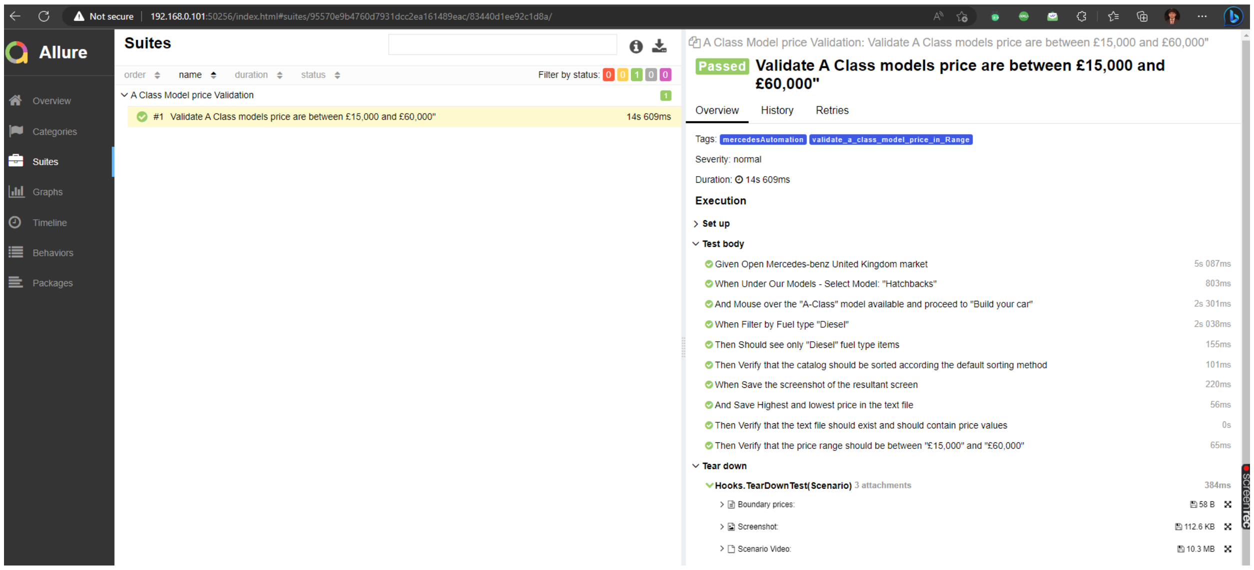
- Install IntelliJ or any other java IDE Open project in IDE
- Go to the file in the directory <project repository>\src\test\java\runConfig
- Right click on the “RunTest.java” file select run
- In that case no report will be generated.



```
    workspace.xml
  > drivers
  > screenshots
  v src
    v test
      v java
        > base
        > getProperties
        > pageObject
        > properties
        v runConfig
            RunTest
        > scripts
        > stepdefinitions
      > resources
  > target
  > test-recordings
  > text-files
    .gitignore
```

Output files and Report:

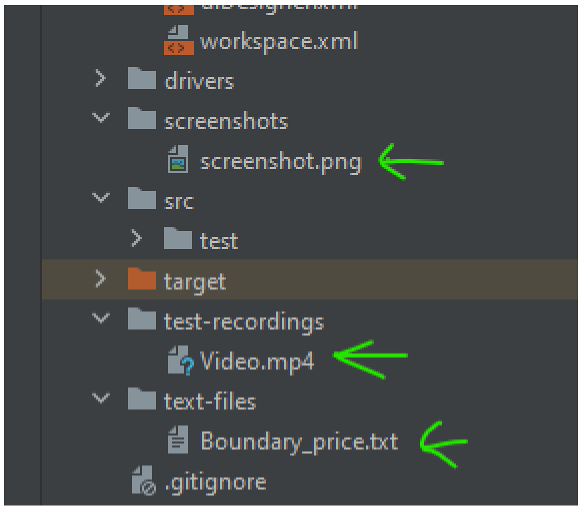
After the execution, the Allure report will be generated if "allure:serve" command is used. That shows the project execution details.



```
Not secure | 192.168.0.101:50256/index.html#suites/95570e9b4760d7931dcc2ea161489eac/83440d1ee92c1d8a/

Allure | Suites
order  name  duration  status          Filter by status: 0 0 1 0 0
v A Class Model price Validation                                   1
   #1 Validate A Class models price are between £15,000 and £60,000"    14s 609ms

A Class Model price Validation: Validate A Class models price are between £15,000 and £60,000"
Passed  Validate A Class models price are between £15,000 and £60,000"
Overview   History   Retries
Tags: mercedesAutomation  validate_a_class_model_price_in_Range
Severity: normal
Duration: 14s 609ms
Execution
> Set up
v Test body
  Given Open Mercedes-benz United Kingdom market                                        5s 087ms
  When Under Our Models - Select Model: "Hatchbacks"                                    803ms
  And Mouse over the "A-Class" model available and proceed to "Build your car"          2s 301ms
  When Filter by Fuel type "Diesel"                                                     2s 038ms
  Then Should see only "Diesel" fuel type items                                         155ms
  Then Verify that the catalog should be sorted according the default sorting method    101ms
  When Save the screenshot of the resultant screen                                      220ms
  And Save Highest and lowest price in the text file                                    56ms
  Then Verify that the text file should exist and should contain price values           0s
  Then Verify that the price range should be between "£15,000" and "£60,000"            65ms
v Tear down
  v Hooks.TearDownTest(Scenario) 3 attachments                                          384ms
    > Boundary prices:                                                                  58 B
    > Screenshot:                                                                       112.6 KB
    > Scenario Video:                                                                   10.3 MB
```

After the execution, there are following 3 files will be generated:

- text file of prices
- image file of last web screen
- the video file of entire scenario



These 3 files are also attached to Allure report.

<u>Pre-requisites for execution: (May be required)</u>

There are following pre-requests may be required to run the automation project

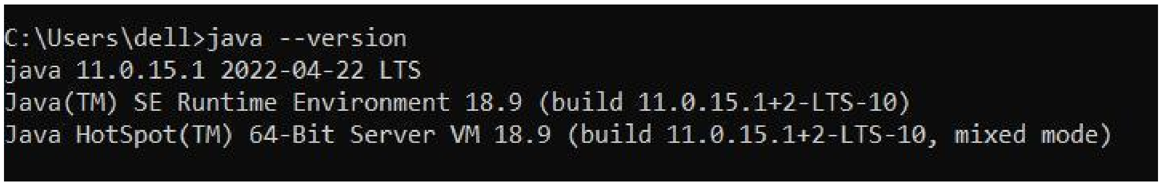
2. Make sure that of Google Chrome is installed. (latest version recommended)

3. The JDK or JRE must be installed in your PC. (JDK version 8 or higher)

    a. Open command prompt or terminal.

    b. Type “java -- version” and press enter.

    c. Should be able to get java version if it is installed and configure correctly.



```
C:\Users\dell>java --version
java 11.0.15.1 2022-04-22 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.15.1+2-LTS-10)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.15.1+2-LTS-10, mixed mode)
```

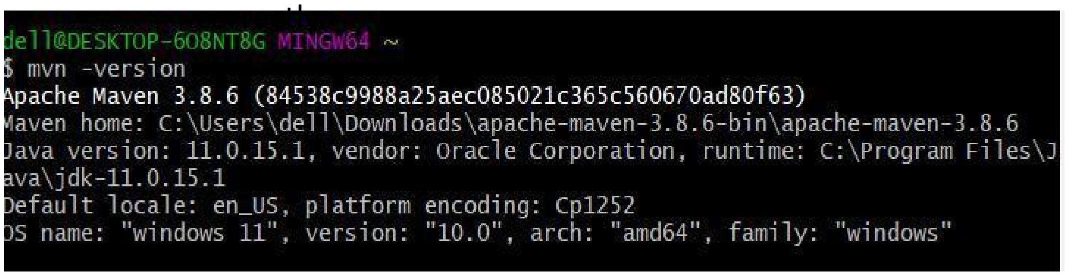
If you are getting the error “java command not found” then following the steps as stated in the following video to install the JDK in your PC.

How to install JDK:
https://www.youtube.com/watch?v=IJ-PJbvJBGs&ab_channel=ProgrammingKnowledge

3. The Maven must be installed in your pc (3.6 or higher version is required)

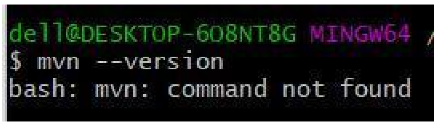
    a. In the command prompt or terminal type: mvn -version

    b. Should be able to get maven version if it is installed and configured



```
dell@DESKTOP-6O8NT8G MINGW64 ~
$ mvn -version
Apache Maven 3.8.6 (84538c9988a25aec085021c365c560670ad80f63)
Maven home: C:\Users\dell\Downloads\apache-maven-3.8.6-bin\apache-maven-3.8.6
Java version: 11.0.15.1, vendor: Oracle Corporation, runtime: C:\Program Files\J
ava\jdk-11.0.15.1
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```

If you are getting the error “mvn: command not found”



```
dell@DESKTOP-6O8NT8G MINGW64
$ mvn --version
bash: mvn: command not found
```

Follow the steps that are mentioned in the video to install and configure the maven:

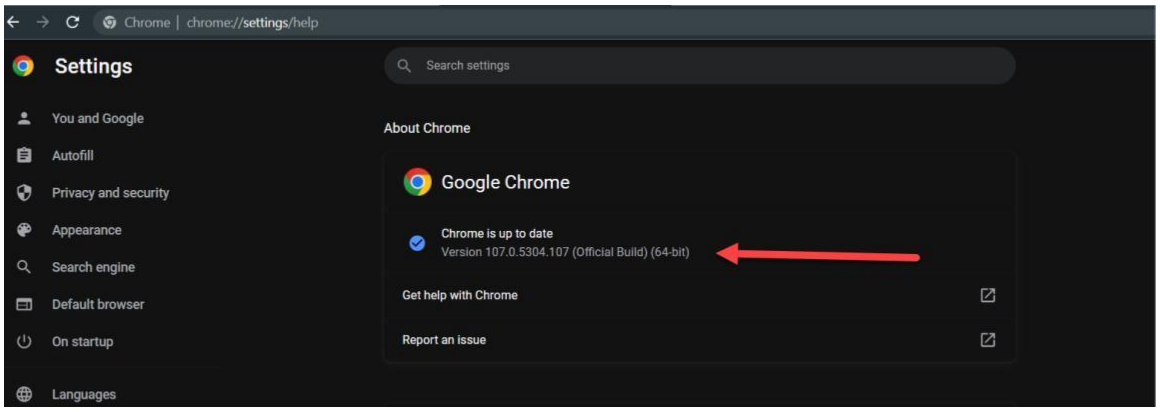
https://www.youtube.com/watch?v=RfCWg5ay5B0&ab_channel=CodingMagic

Errors that may encounter during execution (Leave this section if everything works is ok)

Web driver issue:

During execution you may encounter error related to the chrome driver. In that case do the following steps to resolve it:

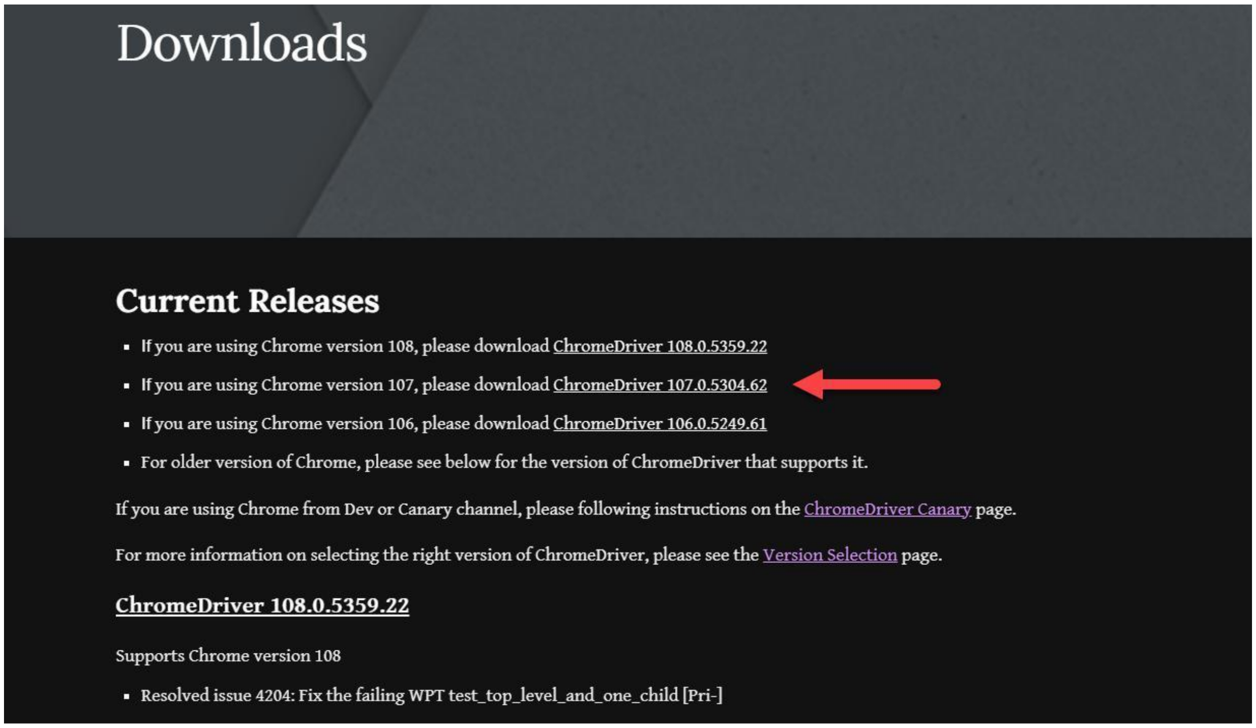
1. Check the google chrome version



Chrome is up to date
Version 107.0.5304.107 (Official Build) (64-bit)

2. Go to the site and download the chrome driver having the same version as google chrome has like 107 version for both.

    a. Link: https://chromedriver.chromium.org/downloads


Downloads

Current Releases

- If you are using Chrome version 108, please download ChromeDriver 108.0.5359.22
- If you are using Chrome version 107, please download ChromeDriver 107.0.5304.62
- If you are using Chrome version 106, please download ChromeDriver 106.0.5249.61
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

If you are using Chrome from Dev or Canary channel, please following instructions on the ChromeDriver Canary page.

For more information on selecting the right version of ChromeDriver, please see the Version Selection page.

ChromeDriver 108.0.5359.22

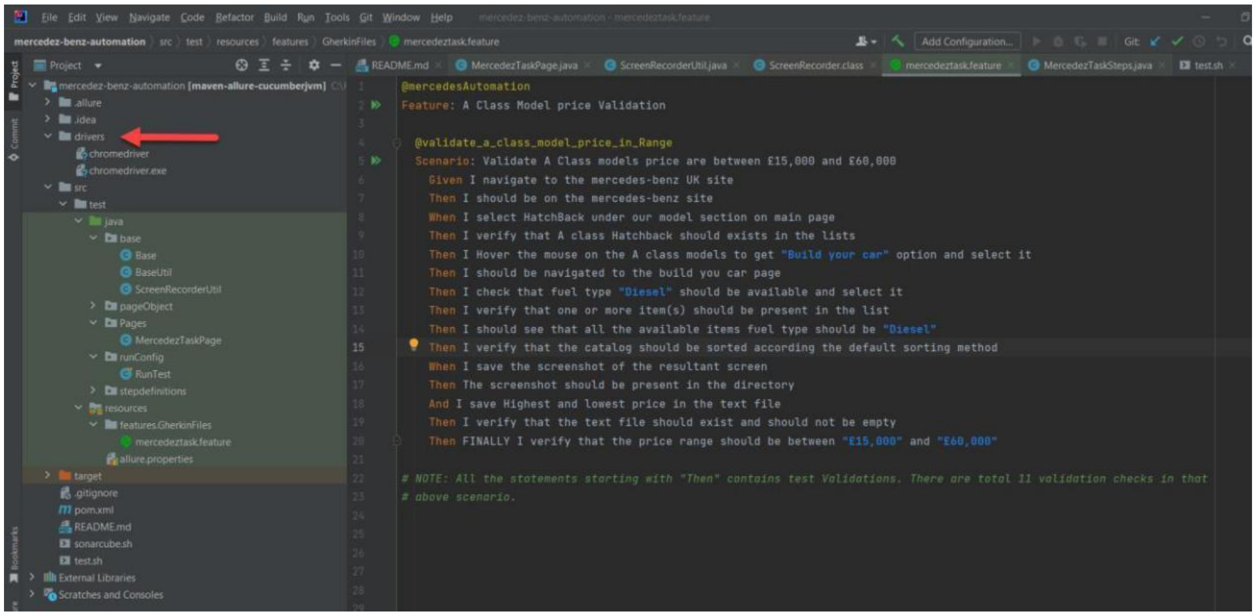
Supports Chrome version 108

- Resolved issue 4204: Fix the failing WPT test_top_level_and_one_child [Pri-]

3. Extract the downloaded zip file.

4. Copy the chromedriver file and paste it to <Project repository>/Drivers

5. Before the copy step, remove the existing chrome drivers in the <Project repository>/Drivers



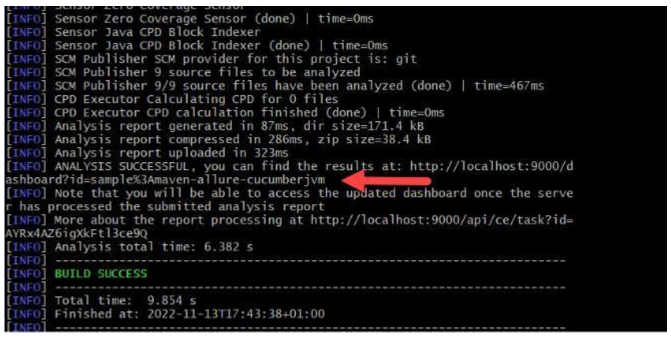
```
@mercedesAutomation
Feature: A Class Model price Validation

  @validate_a_class_model_price_in_Range
  Scenario: Validate A Class models price are between £15,000 and £60,000
    Given I navigate to the mercedes-benz UK site
    Then I should be on the mercedes-benz site
    When I select HatchBack under our model section on main page
    Then I verify that A class Hatchback should exists in the lists
    Then I Hover the mouse on the A class models to get "Build your car" option and select it
    Then I should be navigated to the build you car page
    Then I check that fuel type "Diesel" should be available and select it
    Then I verify that one or more item(s) should be present in the list
    Then I should see that all the available items fuel type should be "Diesel"
    Then I verify that the catalog should be sorted according the default sorting method
    When I save the screenshot of the resultant screen
    Then The screenshot should be present in the directory
    And I save Highest and lowest price in the text file
    Then I verify that the text file should exist and should not be empty
    Then FINALLY I verify that the price range should be between "£15,000" and "£60,000"

# NOTE: All the statements starting with "Then" contains test Validations. There are total 11 validation checks in that
# above scenario.
```

Same steps are followed for the Edge driver as well.

SonarQube Analysis

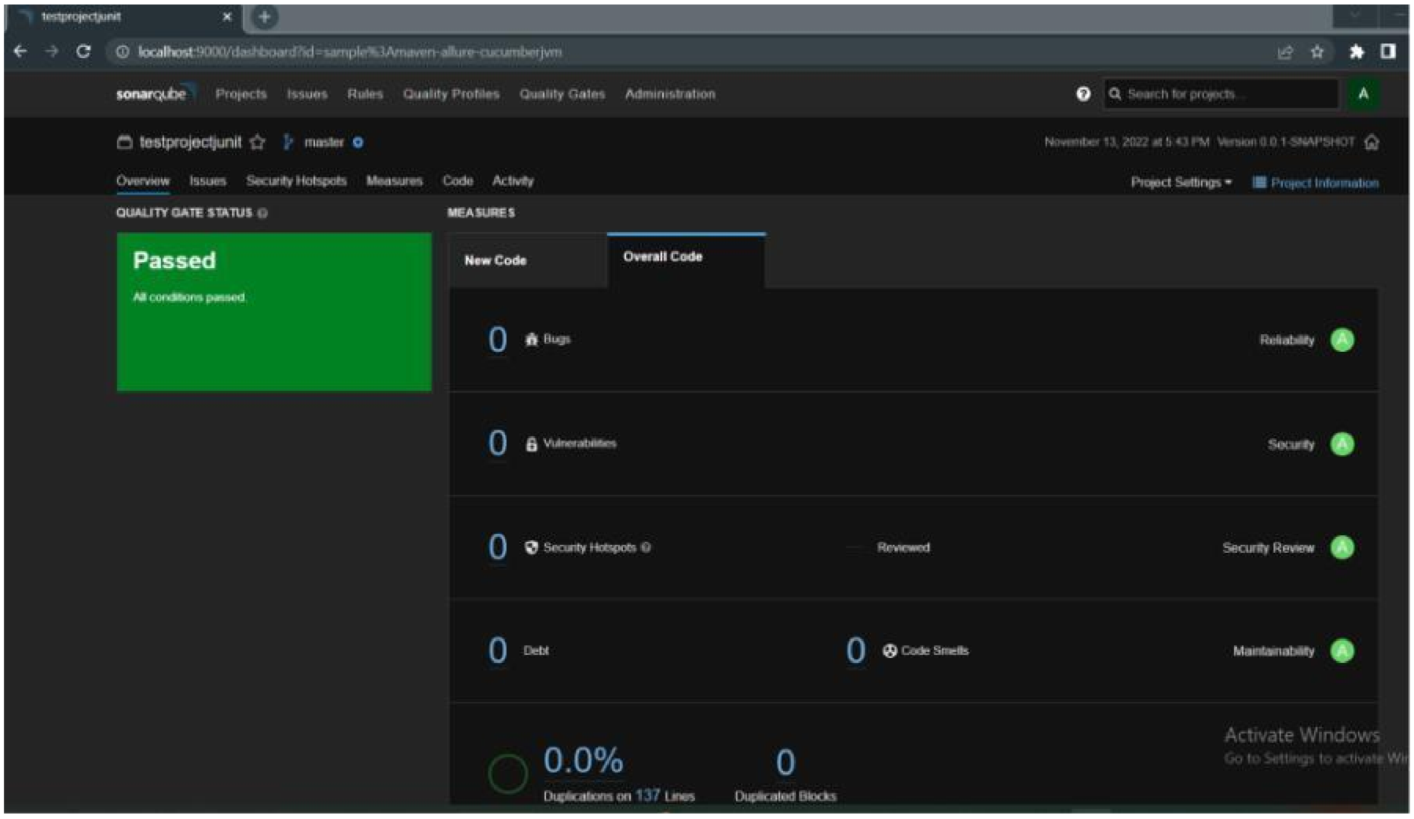
In Order get sonarQube analysis do the following steps

1. Configure and run SonarQube server.
2. Go to the test project repository
3. Run sonar cube analysis
    a. Execute the runSonarQube.bat file in windows
    b. OR RUN THE command in windows command prompt: `mvn sonar:sonar`
4. Go to the link mentioned in the SonarQube logs



```
[INFO] Sensor Zero Coverage Sensor
[INFO] Sensor Zero Coverage Sensor (done) | time=0ms
[INFO] Sensor Java CPD Block Indexer
[INFO] Sensor Java CPD Block Indexer (done) | time=0ms
[INFO] SCM Publisher SCM provider for this project is: git
[INFO] SCM Publisher 9 source files to be analyzed
[INFO] SCM Publisher 9/9 source files have been analyzed (done) | time=467ms
[INFO] CPD Executor Calculating CPD for 0 files
[INFO] CPD Executor CPD calculation finished (done) | time=0ms
[INFO] Analysis report generated in 87ms, dir size=171.4 kB
[INFO] Analysis report compressed in 286ms, zip size=38.4 kB
[INFO] Analysis report uploaded in 323ms
[INFO] ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/d
ashboard?id=sample%3Amaven-allure-cucumberjvm
[INFO] Note that you will be able to access the updated dashboard once the serve
r has processed the submitted analysis report
[INFO] More about the report processing at http://localhost:9000/api/ce/task?id=
AYRx4AZ6igXkFtl3ce9Q
[INFO] Analysis total time: 6.382 s
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  9.854 s
[INFO] Finished at: 2022-11-13T17:43:38+01:00
[INFO] ------------------------------------------------------------------------
```

5. The SonarQube Report will be opened


testprojectjunit
localhost:9000/dashboard?id=sample%3Amaven-allure-cucumberjvm
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
testprojectjunit
master
November 13, 2022 at 5:43 PM Version 0.0.1-SNAPSHOT
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
QUALITY GATE STATUS
Passed
All conditions passed.
MEASURES
New Code
Overall Code
0 Bugs
Reliability A
0 Vulnerabilities
Security A
0 Security Hotspots
Reviewed
Security Review A
0 Debt
0 Code Smells
Maintainability A
0.0%
Duplications on 137 Lines
0
Duplicated Blocks
Activate Windows
Go to Settings to activate Wi