

Cucumber Features

localhost:63342/VehicleCheck/target/features/cucumber-pretty/index.html?\_ijt=ob9bm30g13o11vqr4ci28ankft

▼ **Feature:** Perform a free car check

▼ **@vehicleTest Scenario Outline:** Verify Vehicle details

**Given** I parse list of Car Registration Numbers from the "<input\_file>"

**And** I parse expected car details from the "<output\_file>"

**When** I check car details in the car tax check website

**Then** Actual vehicle details are verified with expected details

▼ **Examples:**

input_file	output_file
src/test/resources/testdata/car_input.txt	src/test/resources/testdata/car_output.txt

▼ **@vehicleTest Scenario Outline:** Verify Vehicle details

**Given** I parse list of Car Registration Numbers from the "src/test/resources/testdata/car\_input.txt"

**And** I parse expected car details from the "src/test/resources/testdata/car\_output.txt"

**When** I check car details in the car tax check website

**Then** Actual vehicle details are verified with expected details

VehicleCheck -- vehicle-check.feature [VehicleCheck.test]

VehicleCheck | src | test | resources | features | vehicle-check.feature

Project | java | drivers | CustomDriverSource | CustomDriverSourceImpl | DriverType | pages | stepdefinitions | steps | utils | TestRunner | resources | features | vehicle-check.feature

1 Feature: Perform a free car check

2

3 @vehicleTest

4 Scenario Outline: Verify Vehicle details

5 Given I parse list of Car Registration Numbers from the "<input\_file>"

6 And I parse expected car details from the "<output\_file>"

7 When I check car details in the car tax check website

8 Then Actual vehicle details are verified with expected details

9 Examples:

10 | input\_file | output\_file

11 | src/test/resources/testdata/car\_input.txt | src/test/resources/testdata/car\_out

12

Run: Feature: vehicle-check

Done: Scenarios 1 of 1 (14 s 300 ms)

Test Results

Feature: Perform a free car check 13 s 118 ms

Scenario Outline: Verify Vehicle details 13 s 118 ms

Examples: 13 s 118 ms

Scenario: Line: 11 13 s 118 ms

Given I parse list of Car Registration Numbers from the "src, 859 ms

And I parse expected car details from the "src/test/resources, 92 ms

When I check car details in the car tax check website 3 s 443 ms

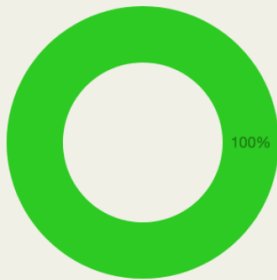
Then Actual vehicle details are verified with expected det, 8 s 722 ms

Hook: after 2 ms

/Library/Java/JavaVirtualMachines/BK01\_jdk1.8.0\_181.jdk/Contents/Home/bin/java -Dorg.jetbra:  
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".  
SLF4J: Defaulting to no-operation (NOP) logger implementation  
SLF4J: See <http://www.slf4j.org/codes.html#StaticLoggerBinder> for further details.  
[DN09HRM, BW57BOW, KT17DLX, S618HTN]  
Starting ChromeDriver 83.0.4103.39 (ccbf011cb2d2b19b506d84400483861342c20cd-refs/branch-he:  
Only local connections are allowed.  
Please see <https://chromedriver.chromium.org/security-considerations> for suggestions on keep  
ChromeDriver was started successfully.  
Jun 21, 2020 5:39:18 AM org.openqa.selenium.remote.ProtocolHandshake createSession  
INFO: Detected dialect: W3C  
No vehicle output data found for Vehicle Reg : BW57BOW

1 Scenarios (1 passed)  
4 Steps (4 passed)  
0m13.301s

## Test Results: All Tests

1 test scenarios (including 1 rows of test data) | [Download](#)[Summary](#)[Test Results](#)

## Scenario Results (including rows of test data)

	Automated
✓ Passing	1 100%
⌚ Pending	0
⌚ Ignored	0
⌚ Skipped	0
✗ Unsuccessful	
✗ Failed	0
⚠ Broken	0
⚠ Compromised	0
Total	1

Execution Time	Clock Time	Fastest Test	Slowest Test	Average Execution Time
17s	17ms	17s	17s	17s

## Functional Coverage Overview

## Features

Feature	Scenarios	% Pass	Result	Coverage
<a href="#">Perform a free car check</a>	1	100%	✓	<div></div>

## Tags

[Vehicletest 1](#)

Home &gt; Perform A Free Car Check &gt; Verify Vehicle details

☒ Overall Test Results ☐ Requirements ☐ Features

Report generated 21-06-2020 05:38



Perform A Free Car Check

Vehicletest (tag)



Verify Vehicle details

## Scenario Outline

Given I parse list of Car Registration Numbers from the "<input\_file>"  
And I parse expected car details from the "<output\_file>"  
When I check car details in the car tax check website  
Then Actual vehicle details are verified with expected details

## Examples:

#	Input File	Output File
1	src/test/resources/testdata/car_input.txt	src/test/resources/testdata/car_output.txt

Steps	Screenshots	Outcome	
Example: {input_file=src/test/resources/testdata/car_input.txt, output_file=src/test/resources/testdata/car_output.txt}		SUCCESS	17.37s
Given I parse list of Car Registration Numbers from the "src/test/resources/testdata/car_input.txt"		SUCCESS	0.33s
Parsing list of Car Registration Numbers from the car_input.txt		SUCCESS	0.01s
And I parse expected car details from the "src/test/resources/testdata/car_output.txt"		SUCCESS	0.14s
Parsing list of expected car details from the car_output.txt		SUCCESS	0.13s
When I check car details in the car tax check website		SUCCESS	6.32s
Navigate to car check website		SUCCESS	6.32s
Then Actual vehicle details are verified with expected details		SUCCESS	10.57s
Verifying Actual vehicle details against expected details		SUCCESS	10.56s
		SUCCESS	17.49s