AUTOMATION SOLUTION – S3 GROUP

S3 GROUP

[Requirement 2](#_Toc485303679)

[Solution 2](#_Toc485303680)

[Approach & Technology 2](#_Toc485303681)

[Test Coverage 3](#_Toc485303682)

[Code Snippet 4](#_Toc485303683)

[Steps to build and execute the tests 4](#_Toc485303684)

[Report 4](#_Toc485303685)

# Requirement

* Automate the testing of S3 Health Group home page
* <http://www.s3connectedhealth.com/>
* Languages, use any one of the following: Java, C#, Ruby
* Use Selenium Framework
* Place the solution in GitHub

# Solution: -

Solution in GitHub: - <https://github.com/sashichotu4321/S3Group>

# Approach & Technology: -

* Understood the requirements and assessed the feasibility of creating the automation solution.
* Used Selenium Webdriver 3.0.0 drivers with SpecFlow framework and C# as programming language.
* Use HttpClient Request/Response for API’s validation.
* Title of the home page (UI) and responses from the API (<http://www.s3connectedhealth.com/>) has been validated.
* Behaviour driven development approach and used Microsoft excel as test data input file, using SpecFlow+Excel plugin to integrate excel with the solution.
* Handle errors using exception handling techniques (try catch blocks).
* Using SepcRun as UnitTestProvider (App.config)
* Advanced reporting using SpecFlow+Runner. HTML file format with rich UI experience (Project Summary, Execution Summary, Timelines, Scenarios Summary and Step Details).

|  |  |
| --- | --- |
| Tools & Technologies | Details |
| Operating system | Windows 10 |
| Programming Language | C# |
| Automation tools | Selenium Webdriver v3.0.0 |
| Automation framework | SpecFlow v2.1.0 |
| Test Data Input | SpecFlow+Excel |
| Reporting methodology | SpecRunner inbuilt HTML reports |
| IDE | Visual Studio 2015 |
| NuGet Packages | Selenium.WebDriver v3.0.0  Selenium.Support v3.0.0  Selenium.WebDriver.ChromeDriver v2.30.0.1  Selenium.InternetExplorer.WebDriver v3.4  NUnit v3.5.0  Newtonsoft.Json v10.0.2 (for API testing)  SpecFlow v2.1.0  SpecFlow.Plus.Excel v1.4.2  SpecRun.Runner v1.5.2  SpecRun.SpecFlow v1.5.2 |
| Browser | IE 11.0 and Chrome v58.0 |
| Version Control Repository | GitHub |

# Test Coverage: -

The below listed validations had been covered in the solution provided.

* + Validated title of S3 Group home page
  + Validated ResponsePhrase, StatusCode and Version of the API

Scenarios Covered:-

UI Part -

Scenario#1:- (Execution Status - Passed)

1. Launch S3 Group website
2. Validate the title contains S3.
3. Exit the browser

API Part -

Scenario#2:- (Execution Status - Passed)

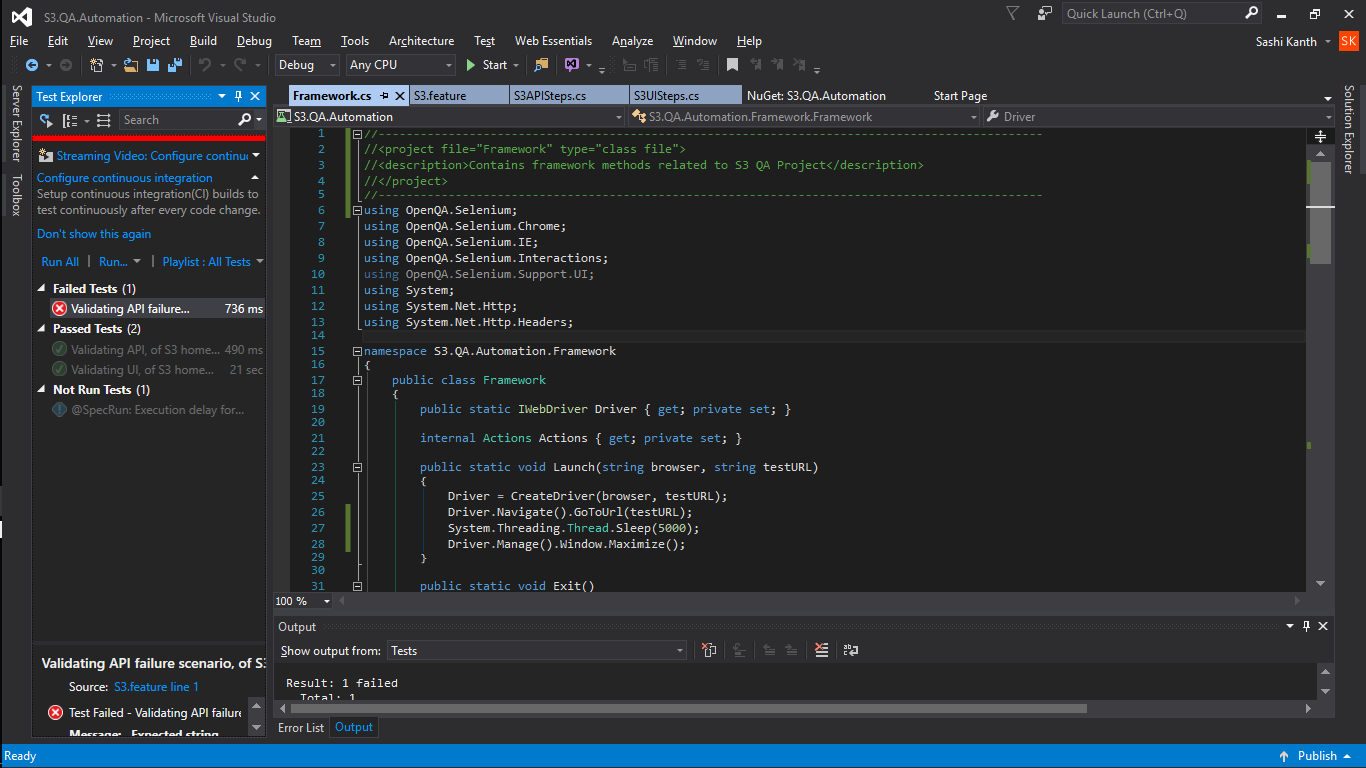
1. Access the “http://www.s3connectedhealth.com/” API
2. Validate that the StatusCode and Version from the responses is “OK” and “1.1” respectively

Scenario#3:- (Execution Status – Purposely Failed, to show how fail scenarios look in reports)

1. Access the “http://www.s3connectedhealth.com/” API
2. Validate that the Reason Phrase from the responses is “NOT OK”
3. Response from the API is “OK” and hence the test failed
4. I failed this test purposely to show the failed scenarios too.

# Code Snippet: -

Please find the code snippet preview in the screenshot below.



# Steps to build and execute the tests: -

1. Checkout the solution from GitHub - <https://github.com/sashichotu4321/S3Group>.
2. Open the solution in Visual Studio IDE.
3. Go to Test-Windows-Test Explorer
4. Build the solution to get all the Nuget packages and dll’s.
5. Right click the tests in the Test Explorer pane and click on Run tests.
6. After execution, click on the Report file link in Output Pane.
7. Reports can also be accessed from “TestResults” folder in solution.

Report***: -***

The report shows the following details

* Project Summary
* Results
* Test Timeline Summary
* Test Results View
* Feature Summary
* Error Summary
* Scenario Summary
* Execution Details
* Steps

Reports Snapshot: - (Path-<Solution>/TestResults)

