**TwoFA Automation:**

A suite of automated tests for the MOD Login web application enrolment using TwoFA authentication OTP token

The TwoFA Project uses Specflow is a C# automation framework that is used for BDD(Behavioral Driven Development) and also makes use of the Gherkin language for the creation of features and Scenarios

The TwoFA for API Automation uses Rest Sharp

Execution process:

Clone the repository:

<https://modais.visualstudio.com/Defence%20Gateway%20Applications/_git/DG%20-%20STS%20-%202FAAutomation>

Getting Started with Specflow:

SpecFlow acceptance tests follow the BDD paradigm: define specifications using examples understandable to business users as well as developers and testers.

Install the SpecFlow for Visual Studio :

1. open Extensions and Manage Extensions

Extensions>Manage Extensions>Search Specflow for Visual Studio 2019/2017>Install>Restart VS>Rebuild

1. Install Browser Drivers

Running tests:

1)Running Spec flow Scenarios based on Tags via Command line:

* Pre-requisite: Nunit.Console - This package includes the nunit3-console runner and test engine for version 3 of the NUnit unit-testing framework.
* Right click on framework solution and click on “open folder in file explorer”

**Syntax**:

PS

[Path of Project] \packages\NUnit.ConsoleRunner.3.11.1\tools> .\nunit3-console.exe “[Path of Project] \bin\Debug\BDD\_TwoFA.dll" --where "cat=*TwoFAAPIValidateSerials* "

2)Running Spec Flow Scenarios from the Visual Studio:

Execute SpecFlow Tests: Once we build the project, our test/scenario will show up in the tests explorer.

Test Outcome: The generated output from the test execution is nice. You can observe each executed step and its execution time.

Debug Scenarios: If you need to debug your tests, you have a few options. You can set a break point inside your binding methods.

**UI Test Scripts Execution Steps:**

To execute the script for the SMS option to get the OTP for the given Mobile number in the FeaturesFile/TwoFA.feature file:

1. Already enrolled user test execution : Go to Test Explorer in the Visual Studio and “@TwoFAUIValidateSMSOptionPages” click on the Scenario “ Verify TwoFA authentication pages with SMS option “ “already enrolled “condition is yes and right click and “Run” Output is: SMS is generated to the given mobile number and enter manually in the Enter Token Page and click on submit, the page redirected to the Defence Home Page
2. User login and click enrolment SMS option : Go to Test Explorer in the Visual Studio and “@TwoFAUIValidateSMSOptionPages” click on the Scenario “ Verify TwoFA authentication pages with SMS option “ “already enrolled “condition is No and right click and “Run” Output is: SMS is generated to the given mobile number and enter manually in the Enter Token Page and click on submit, the page redirected to the Defence Home Page

End-to-End Execution : put the break point under the Pages folder /TokenPage.cs file EnterTokenNum() method in that line String OTP=””, go to Test Explorer in the Visual Studio and “ @TwoFAUIValidateLogin\_viaSMS\_OTP

” click on the Scenario “ Verify TwoFA authentication pages with SMS option “ “already enrolled “condition is No and right click and “Debug” steps are: once SMS is generated to the given mobile number , enter the token number manually to verify and click continue then token number is automatically entered in the UI entertoken webpage then it is redirected to the <https://sso-dev.dgw-dev.mod.uk/adfs/ls/?wa=wsignin1.0&wtrealm=https%3a%2f%2fwww-dev.dgw-dev.mod.uk%2f&wctx=rm%3d0%26id%3dpassive%26ru%3d%252f&wct=2020-06-03T09%3a14%3a30Z> link and automatically click on the “Defence Gateway link then it is redirected to the Defence home page , verified the logout button in the homepage and quit the browser.

1. To execute the script for the Email option to get the OTP for the given Email in the feature file:
2. Go to Test Explorer in the Visual Studio and “TwoFAUIValidateEmailOptionPages”click on the Scenario “ Verify TwoFA authentication pages with Email option “ “already enrolled “condition is yes and right click and “Run” Output is: OTP is generated to the given Email address in the feature file and enter manually in the Enter Token Page and click on submit, the page redirected to the Defence Home Page
3. User login and click enrolment SMS option : Go to Test Explorer in the Visual Studio and “@TwoFAUIValidateEmailOptionPages” click on the Scenario “ Verify TwoFA authentication pages with Email option “ “already enrolled “condition is No and right click and “Run” Output is: SMS is generated to the given Email and enter manually in the Enter Token Page and click on submit, the page redirected to the Defence Home Page
4. End-to-End Execution : put the break point in the TokenPage.cs file EnterTokenNum() method in that line String OTP=””, go to Test Explorer in the Visual Studio and “TwoFAUIValidateLogin\_viaEmail\_OTP” click on the Scenario “ Verify TwoFA authentication pages with Email option “ “already enrolled “condition is No and right click and “Debug” steps are: once OTP is generated to the given Email Address , enter the token number manually to verify and click continue then token number is automatically entered in the UI entertoken webpage then it is redirected to the <https://sso-dev.dgw-dev.mod.uk/adfs/ls/?wa=wsignin1.0&wtrealm=https%3a%2f%2fwww-dev.dgw-dev.mod.uk%2f&wctx=rm%3d0%26id%3dpassive%26ru%3d%252f&wct=2020-06-03T09%3a14%3a30Z> link and automatically click on the “Defence Gateway link then it is redirected to the Defence home page , verified the logout button in the homepage and quit the browser.

**API Test Scripts Execution Process:**

Go to Test Explorer in the Visual Studio select @TwoFAAPIValidateSerials

1. click on the Scenario “ Verify TWoFA API OTP Authentications “ and right click and “Run” Output is: SMS is generated to the given mobile number : to verify the OTP go to postman tool and enter the API endpoint url is : <https://privacyidea.dgw-dev.mod.uk/validate/check> Example Request Body is : {

"pass" : "600887",

"user" : "swapna.tootikoora",

"realm" : "mod.local"

}

click on the send button Out Put is Status Code is “200 ok” and Response is : message": "matching 1 tokens

2)End-To-End Execution(In Debug Mode): Put the debug point in ,go to folder TwoFADefinitions/TwoFADef.cs :

public void WhenVerifyTheOTPReceivedIsCorrectOrNot()

{

this.scenarioContext.Add("APIOTP", "805080");//enter the token number here to verify

string ValidateToken = apiMethods.ValidateOTP(this.scenarioContext.Get<string>("APIOTP"), this.scenarioContext.Get<string>("ApiToken"));

Assert.IsTrue(ValidateToken.Contains("matching 1 tokens"), "OTPis incoreect..");

}

Go to Test Explorer in the Visual Studio select TwoFAAPIValidateOTP click on the Scenario “Verify the OTP received is correct or not “ and right click and click on “Debug” , OTP is generated vis SMS to the given mobile number , then enter the token in the above mentioned method, click “Continue” button . verify the output

**Framework Structure: BDD-TWOFA**

>**APIDataFiles**: Create the API Requests (Payload) in the form of Json Files

>**APIRoots**: Read all the JSON files and map it to the class objects, and properties that matches in the JSON files

>**ApplicationCommon**: Serialize and Deseriliaze the json file into C# classes

>**FeaturesFile**: Consists of 2FA Scenarios for enrolment through different options and validate OTP

>**Pages**: contains page modules (one page per .cs file) which encapsulate all actions that can be performed on the specified page. For example \*EnrollPage.cs\* contains actions such as \*SMSDetails\* and \*EmailDetails\*

>**TwoFADefinitions**: Step definition maps the Test Case Steps in the feature files(introduced by Given/When/Then) to code. For example: write the code to execute the steps on 2FA Scenarios in the feature file and checks the outcomes against expected results.25

>**Utilities**: class stores common methods which will use all over the framework

>**Packages.config**: To maintain the list of **packages** referenced by the project. This allows NuGet to easily restore the project's dependencies when the project to be transported to a different machine, such as a build server, without all those **packages**.

>**App.config**: To test the Scenarios in Chrome or IE browser, change the browser name according that: <add key="Browser" value="Chrome" /> or<add key="Browser" value="Internet" />

**Add Feature File:**

* Create the folder name of Features and click on the Features folder>Add>New Item
* It will create Feature file , in that create your Scenarios which we want automate the functionality

Example:

Feature: TwoFA Feature Testing

In order to access my MOD account

As a user of the MOD website

I want to log into the website using TwoFA OTP authentications

@TwoFAAPI

Scenario Outline: Verify TWoFA API OTP Authentications

Given User has valid token for valid credentials

When User makes request with valid token and for "<Type>" and "<TypeValue>"

Then User should get valid Code using valid serial

Examples:

| Type | TypeValue |

| sms | 43434334343|

| email | rtrrere |

### To be able to execute the scenario you need to define the so-called bindings for each step. We need to create step definitions that bind the statements in the test scenario to the application code. Generate Bindings, Once you click insider the specification file and open the context menu, you will find the 'Generate Step Definitions' item.