Fiddler setup for LIMA overrides

Tuesday, June 20, 2017 8:33 AM

Fiddler Setup for setting Lima Overrides

Fiddler Configuration steps:

1) Install Fiddler
2) Launch Fiddler
3) Replace the CustomRules.js file in C:\Users\<username>\Documents\Fiddler2\Scripts or Fiddler4 \Scripts directory with the CustomRules.js in H:\Department\QualityAssurance\QAAutomation\Fiddler
4) Enable the option: Decrypt HTTPS Traffic (Tools->Fiddler Options->HTTPS tab), It will prompt you to install the Fiddler Root certificate, Click Yes. It should install fine on the laptops, if so, jump to step 9). If not, follow steps from 5)
5) On the HTTPS tab, Click on Export Root Certificate to Desktop
6) Double click the certificate exported on the desktop and click Install Certificate
7) Browse to place the certs in the store, enable Show all Physical Stores
8) Select Local Computer under Trusted Root Certification Authorities
9) Navigate through Next and Finish
10) Relaunch Fiddler

Note: On the QAAuto VMs, the steps above have been taken care and to avoid having to run Fiddler physically, we have the Fiddler setup as a service and running throughout.

What should the scripter do to make use of the Fiddler setup?

1) Ensure Fiddler or Fiddler Service is running

2) Script side:

   Ruby:

   In the Url.rb file, Set the global constant $bOverride = true

   Protractor:

   In the conf.js file, remove code to load the extension and copy the following lines
   in onPrepare: function() -

```
 CommonFunctions = require( __dirname + '/PageObjects/common-files/common-functions.po' );
 CommonFunctions.setCookieAndNavigateTo('factset.com');
```

How it works:

Application expects a header named “X-FDS-Override-Name” and its value is the expected Lima override which helps the application hit the appropriate server.

This can be done simply by navigating to the function “static function OnBeforeRequest (oSession: Session)”in Fiddler’s CustomeRules.js file. In this function, add the header values you want to add in the format:

Eg. oSession.oRequest["X-FDS-Override-Name "] = "spar-devel";

But this injects the override to all the requests running on the system. This is not ideal when run on the Selenium grid as there are multiple runs on the same machine. Hence we have modified the CustomRules.js such that the Override is set by translating it from a Cookie.

In order to do this, the script needs to set the cookie with name as "LimaOverrideCookie" and the cookie value as the Override value which will get translated as the header “X-FDS-Override-Name” using Fiddler

eg. If “spar-devel” is the pre-build override

Ruby:

```
browser.goto "http://spar.staging-cauth.factset.com/index.html#/doc/CLIENT:DEVEL-AUTOMATION/report/1"
browser.manage().add_cookie({:name => "LimaOverrideCookie", :value => "spar-devel"})
```

Protractor:

```
browser.get(‘https://spar.staging-cauth.factset.com/index.html#/doc/CLIENT:DEVEL-AUTOMATION/report/1’ );
browser.manage().addCookie( 'LimaOverrideCookie', 'spar-devel', ' ', 'factset.com' );
```

- Use the FactSet .Net credentails to login and it re-directs to the appropriate application
- Navigate to the url again as Login page gets redirected to the default view (document) of the application initially

In Ruby, this is taken care in the UtilsCommon.NavigateTo() function
In Protractor, this is taken care in the CommonFunctions.setCookieAndNavigateTo() function

Reference

Here is the document to setup fiddler http://tellusdev/qadocs/Fiddler%20Setup%20for%20setting%20Lima%20Overrides.htm.