[Skip to end of metadata](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#page-metadata-end)

* Created by [Oushana, Romin Bob](https://devcloud.swcoe.ge.com/devspace/display/~212556710), last modified by [Xu, Li](https://devcloud.swcoe.ge.com/devspace/display/~212437180) on [Feb 26, 2018](https://devcloud.swcoe.ge.com/devspace/pages/diffpagesbyversion.action?pageId=1094052848&selectedPageVersions=184&selectedPageVersions=185)

[Go to start of metadata](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#page-metadata-start)

**This repository is PUBLIC, simply clone it.**

**ProUI can be used to reduce front-end test automation effort if the tests are suitable for functional testing. As it built on top of Selenium Web driver and Protractor, it provides the capability to construct the tests using common APIs. It provides common methods, logger/reporting capabilities, and support for both Jasmine and Cucumber.**

* [This repository is PUBLIC, simply clone it.](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-ThisrepositoryisPUBLIC,simplycloneit.)
* [Distribution List](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-DistributionList)
* [System Requirements](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-SystemRequirements)
* [Installation & Execution](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Installation&Execution)
  + [Current Util Version](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-CurrentUtilVersion)
  + [Introductory Video](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-IntroductoryVideo)
  + [Instructions](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Instructions)
  + [Execution commands](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Executioncommands)
* [File Structure](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-FileStructure)
* [Types of Locators](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-TypesofLocators)
* [Logging/Reports](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Logging/Reports)
  + [Browser logging:](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Browserlogging:)
    - [Implementation:](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Implementation:)
  + [Tags from grunt:](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Tagsfromgrunt:)
* [Common Utils](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-CommonUtils)
  + [TestHelper](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-TestHelper)
  + [Object Manager](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-ObjectManager)
  + [Highchart](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Highchart)
    - [Simple highchart validation example:](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Simplehighchartvalidationexample:)
    - [Highchart data validation example:](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Highchartdatavalidationexample:)
  + [RestHelper](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-RestHelper)
    - [RestHelper for getting Highchart series data:](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-RestHelperforgettingHighchartseriesdata:)
  + [Protractor Image Comparison (protractor-image-comparison)](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-ProtractorImageComparison(protractor-image-comparison))
* [Constructing Tests](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-ConstructingTests)
  + [Conf File](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-ConfFile)
  + [Feature File Example (ApmLogin)](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-FeatureFileExample(ApmLogin))
  + [Asset-Step-Def](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-Asset-Step-Def)
  + [Page Object Example (Asset-Tenant-PO)](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-PageObjectExample(Asset-Tenant-PO))
  + [iFrame Example](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-iFrameExample)
* [Drag and Drop Example](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-DragandDropExample)
* [Rally Integration](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-RallyIntegration)
  + [Steps to generate API Key](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-StepstogenerateAPIKey)
  + [External Data Parameterization](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-ExternalDataParameterization)
  + [Viewing Results on Rally After Test Execution](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-ViewingResultsonRallyAfterTestExecution)
  + [Jasmine Rally Configuration](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-JasmineRallyConfiguration)
  + [Rally Debugging](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-RallyDebugging)
* [Windows Configuration](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-WindowsConfiguration)
* [Internet Explorer Configuration](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-InternetExplorerConfiguration)
* [Pushing to own Repository/branch](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-PushingtoownRepository/branch)
* [Jenkins Configuration](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-JenkinsConfiguration)
* [Key Notes](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI#ProUI-KeyNotes)

definethefeaturenameas

Distribution List

@Digital EngSS Tools FTE <EngSSToolsFTE@[ge.com](http://ge.com/)>

**System Requirements**

| **Component** | **Value** |  |
| --- | --- | --- |
| Node | 4.4.7 or greater | **Mandatory** |
| NPM | 3.9.0 or greater | **Mandatory** |
| Python | 2.7.11 | Optional  (required only to test HighCharts) |
| Visual Studio (Express) | 2015 Community Edition (Windows Only) | Optional  (required only to test HighCharts) |
| Xcode Command Line Tools | Latest (Mac Only) | Optional  (required only to test HighCharts) |

**Installation & Execution**

**Current Util Version**

* 1.1.66

**Introductory Video**

* <https://ge.box.com/s/4537mko684s1kn5v4ucujiit6nby6yai>

\*\*Please download video for higher quality\*\*

**Instructions**

* Should have Github account
* Make sure to have SSH key generated as part of your GitHub account (<https://help.github.com/articles/generating-an-ssh-key/>)
* Open new terminal and clone repo from GitHub to the local system ( ProUI repo is PUBLIC now with Read-only access)

|  |
| --- |
| git clone git@github.build.ge.com:MicroTester/ProUI.git |

* In case of issues using the SSH key, clone it using https
* git clone <https://github.build.ge.com/MicroTester/ProUI.git>
* Open IDE (WebStorm) and select File-> Open and navigate to cloned Project
* Run the commands below to install the dependencies:  
  *Prerequisite: Should have NPM installed.*

|  |
| --- |
| npm install -g grunt --proxy=http://sjc1intproxy02.crd.ge.com:8080  npm config set strict-ssl false  npm install --only=dev --proxy=http://sjc1intproxy02.crd.ge.com:8080  npm install grunt-execute --save-dev |

* Execution commands
* commands

**File Structure**

 Folder Structure

***\*\*\*People using Jasmine instead of Cucumber will not have a Feature folder\*\*\****

**Types of Locators**

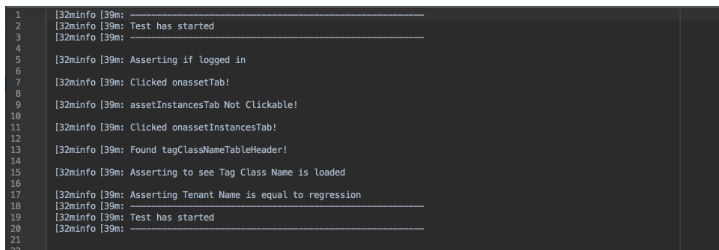
Different locator types are supported to detect locators as per:

| **id** | **idArray** | **css** | **cssArray** | **cssAll** | **cssContainingText** | **name** | **className** | **xPath** | **linkText** | **buttonText** | **binding** | **exactBinding** | **repeater** | **exactRepeater** | **partialLinkText** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

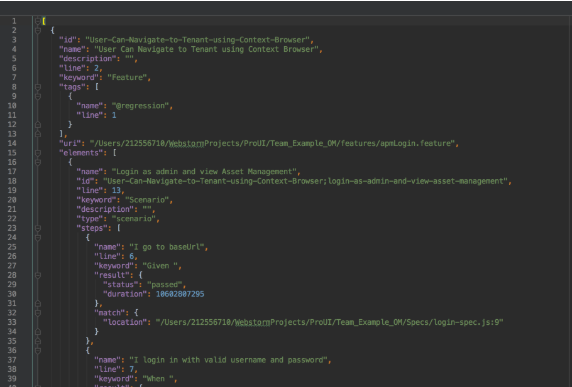
| **id** | **idArray** | **css** | **cssArray** | **cssAll** | **cssContainingText** | **name** | **className** | **xPath** | **linkText** | **buttonText** | **binding** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

**Logging/Reports**

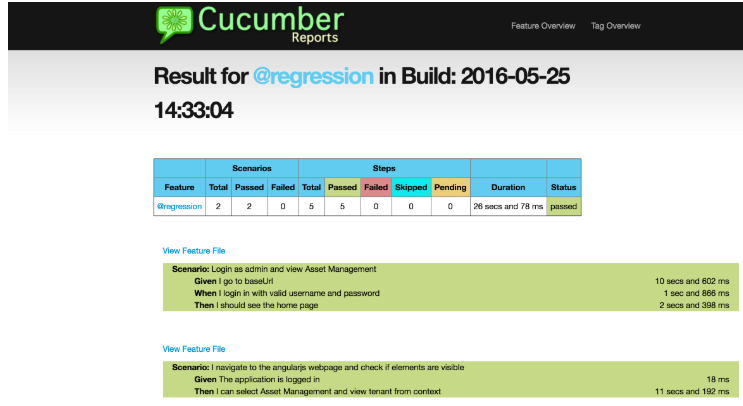
* ProUI captures the test execution logs under the Reports directory:



* It also generates a JSON report for the cucumber reports:

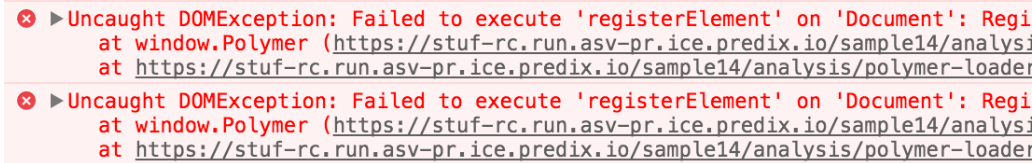


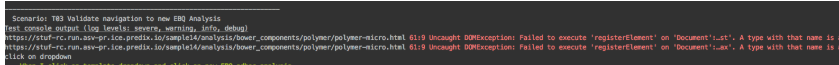
* In addition, from the JSON reports, HTML reports can be generated as well.



**Browser logging:**

* This implementation will be useful if you want to log browser console logs to protractor logs. This will help you to monitor reasons for test failures.
* Logs in like below will be posted,



* They will be posted inlogsas following,
* 

**Implementation:**

* + - 1. Do,

|  |
| --- |
| npm i -D protractor-console  2. Add following to config file you wish to run, |

|  |
| --- |
| plugins: [      {          package: 'protractor-console',          logLevels: ['severe', 'warning', 'info', 'debug']      }  ], |

[**log levels**](https://github.com/SeleniumHQ/selenium/wiki/Logging)**:**

* + - * + **OFF**: Turns off logging
        + **SEVERE**: Messages about things that went wrong. For instance, an unknown command.
        + **WARNING**: Messages about things that may be wrong but was handled. For instance, a handled exception.
        + **INFO**: Messages of an informative nature. For instance, information about received commands.
        + **DEBUG**: Messages for debugging. For instance, information about the state of the driver.
        + **ALL**: All log messages. A way to collect all information regardless of which log levels that are supported.

**Tags from grunt:**

* This implementation will be helpful if you want to call tags from command line instead of hard coding it in grunt file.
* Addfollowingobject to grunt file,

|  |
| --- |
| cucumberOpts: {      "tags": grunt.option('tags')  }, |

* **following are the combinations can now be performed to run tagged tests,**

--tags=@foo          # Only runs foo scenario from whole suite

--tags=~@foo         # Runs all tests except foo scenario

--tags=@foo,@bar           # Runs both scenarios (Scenarios with @foo OR @bar)

* **Scenarios can be tagged as following,**

@foo2

**Scenario:**foo scenario

**When** foo given

**Then** foo then

**Common Utils**

TestHelper

Represents a library of canned expected conditions that will make testing easier such has adding the ability to dynamically wait for elements to load.

Object Manager

Object Manager allows us to manage existing objects available in ProUI along with interaction with multiple json files on the fly.

**The first thing after creating a step definition is to create an object of Object Manager.**

Below is the syntax to create an object of Object Manager (Define this in your config file):

|  |
| --- |
| var objManager = require('ProUI-Utils').ObjectManager;  var objectManager = new objManager(); |

Below are the available methods in Object Manager:

**1.  getLocalObjectManager: U**sed to call all the ProUI functions and to access the locators which are stored locally at module level in Locators folder.

Syntax for using **getLocalObjectManager:**

|  |
| --- |
| var objManager = objectManager.getLocalObjectManager("LCC", "test",”SampleTestData”); |

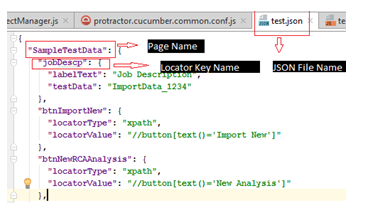
In above declaration:

1st Argument (LCC):                               Module Name,

2nd Argument (test):                              JSON File Name

3rd Argument (SampleTestData):          Page Name in test.Json file

       Below is the screen shot from test.json file



**getLocalObjectManager** **function returns an object which internally contains Objects of “Test**

**Helper, RestHelper and Element** **Manager”**.

Same thing needs to be done if we want to access Test Helper (provides capability to interact with page locators), Rest Helper (provides limited capability to make http requests)

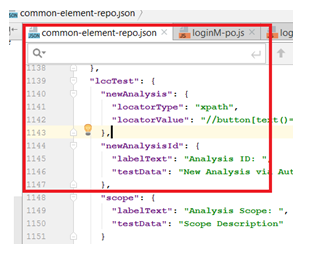
and Element Manger (provides capability to use page locators) functions as shown below:

**Below is the syntax to get Object of Test Helper:**

Let’s say we want to use “**isElementPresent”** function from Test Helper then first get an object of Test Helper:

|  |
| --- |
| var ObjTestHelper = objManager.TestHelper;  ObjTestHelper.isElementPresent(“lccTest”, "newAnalysis").then(function () {      console.log("new analysis button is visible");  }).catch(function(err) {      console.log(“Not able to find newAnalysisbutton”);  }); |

Below is the screen shot from json file where we have stored “**newAnalysis**” locator in the page name **lccTest**:



Example for using Element Manager function from **getLocalObjectManager**:

Let’s say we want to use **findElement** function from ProUI Element Manager then first get an object ofElement Manager**:**

Below is the syntax to get Object ofElement Manager**:**

|  |
| --- |
| var ObjEleManager = objManager.ElementManager;  ObjEleManager. findElement(lccTest, “newAnalysis”).elementToBeClickable();.then(function () {      console.log("clicked on the new analysis");  }).catch(function(err) {      console.log(“Not able to click on newAnalysis button”);  }); |

Example for using Rest Helper function from **getLocalObjectManager** :

Let’s say we want to use **executeGetRequest** function from ProUI Rest Helper then first get an object ofRest Helper**:**

Below is the syntax to get Object ofRest Helper**:**

|  |
| --- |
| ObjRestHelper.executeGetRequest(url, opts.headers, function (err, res) {      var uuid = JSON.parse(JSON.stringify(res.body)).dataObject[0].uuid;      console.log('Found user : ' + uuid);      Logger.info('Found user : ' + uuid);      ObjRestHelper.executeDeleteRequest(opts.url + '/' + uuid, opts.headers, function (err, res) {          Logger.info('Deleted user : ' + uuid);          console.log('Deleted user : ' + uuid);      });  }); |

**2.  getGlobalObjectManager (Used only at framework level and not at module level)**:

* It will only interact with **common-element-repo.json**file, which is located under **Common\_Template** folder.
* getGlobalObjectManager can be used to call all the ProUI functions and to access the locators which are stored globally in json file named “**common-element-repo.json”**

**Note**:  **common-element-repo.json** is only used to store the locators which are common to all the modules for reusability.

Below is the syntax to call getGlobalObjectManager function:

**var** globalObjectManager = objectManager.getGlobalObjectManager(**"lccTest"**);

In above declaration:

1st Argument (lccTest):                          Page Name

**Note**: Here object manager internally contains objects of TestHelper, ElementManger, and RestHelper

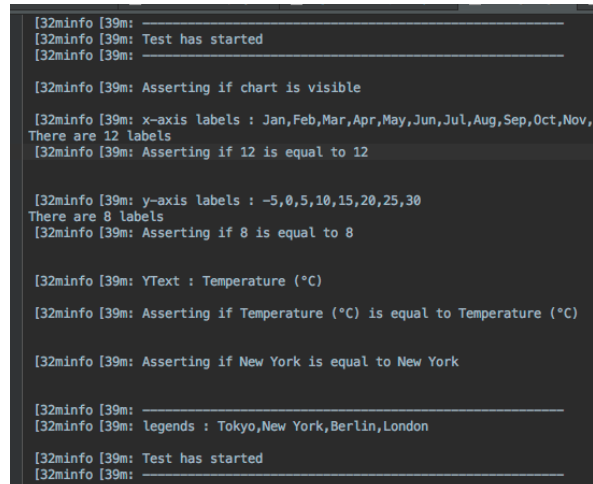
Highchart

**Simple highchart validation example:**

Represents a library of methods for testing Highchart data models.

|  |
| --- |
| this.Given(/^The chart is loaded/, function (callback) {  // Setting the highchart to retrieve information from it  highchartDemo.setChart();        //Checking to see if the highchart is visible      highchartDemo.isChartVisibile().then(function (result) {          Logger.info("Asserting if chart is visible\n");          TestHelper.assertTrue(result, 'Chart is not available');          callback();      })  });    this.Then(/^I can view the chart details$/, function (callback) {    // Retrieving the X-Axis Labels      highchartDemo.getXAxisLabels().then(function (xlabels) {          Logger.info('x-axis labels : ' + xlabels + '\n' + 'There are ' + xlabels.length + ' labels');          console.info('x-axis labels : ' + xlabels + '\n' + 'There are ' + xlabels.length + ' labels');            // Asserting if there are 12 labels on the X-Axis         TestHelper.assertEqual(12, xlabels.length, callback);            //Retrieving Y-Axis Labels          highchartDemo.getYAxisLabels().then(function (ylabels) {              Logger.info('y-axis labels : ' + ylabels + '\n' + 'There are ' + ylabels.length + ' labels');              console.info('y-axis labels : ' + ylabels + '\n' + 'There are ' + ylabels.length + ' labels');                // Asserting if there are 8 labels on the Y-Axis              TestHelper.assertEqual(8, ylabels.length, callback);                //Retrieving Y-Axis Text              highchartDemo.getYaxisText().then(function (ytext) {                  Logger.info('YText : ' + ytext + '\n');                  console.log('YText : ' + ytext + '\n');                    // Asserting if the Y-Axis text is equal to a string                  TestHelper.assertEqual(ytext, 'Temperature (°C)', callback);                    //Retrieving the Legends                  highchartDemo.getLegends().then(function (legends) {                        // Asserting to see if the second legend is equal to New York                      TestHelper.assertEqual('New York', legends[1], callback);                      Logger.info('legends : ' + legends + '\n');                      console.log('legends : ' + legends + '\n');                      callback();                  });              });          });      })  }); |
|  |

**Sample Output:**



**Highchart data validation example:**

For Data Validation the RobotJS is required for this part, it will move the mouse to trigger the highcharts tooltip and then we can get the information

Please ensure you have the required dependencies before installing:

* Windows
  + Visual Studio 2013 (Express works fine).
  + Python (v2.7.3 recommended, v3.x.x is not supported).
* Mac
  + Xcode Command Line Tools.
* Linux
  + Python (v2.7 recommended, v3.x.x is not supported).
  + make.
  + A C/C++ compiler like GCC.
  + libxtst-dev and libpng++-dev (sudo apt-get install libxtst-dev libpng++-dev)

For APM part:

|  |
| --- |
| //set up the mouse start position, and the scroll range and direction  // you can use page ruler chrome plugin to see the position  // If you are using Mac, you can use shift + cmd + 4 then you can see the mouse position on the screen    var startx = 300  var starty = 450  var endx = 400  var rollrange = 5  var rolldirection ="down"  // test case 1: verify a tag exit in a time stamp    var Timestamp1 ="07/11/16 08:30:00 AM";  var Tagname\_1 = "EXHAUST\_TEMP\_T48B\_EE-43-TI-11243B\_Actual"  apmHighChartPage.getTooltipInfo(Timestamp1,Tagname\_1, true, startx, starty, endx, rollrange, rolldirection)  .then(function(data){      console.log("data\_2 is " +data)      TestHelper.assertTrue(data, callback);      callback()  })    // test case 2: verify two tags in a time stamp  rollrange = 5;  Timestamp1= "07/12/16 12:45:00 AM"  Tagname\_1 = "EXHAUST\_TEMP\_T48B\_EE-43-TI-11243B\_Actual,EXHAUST\_TEMP\_T48B\_EE-43-TI-11243B\_Actual\_Smoothed"  apmHighChartPage.getTooltipInfo(Timestamp1,Tagname\_1, true, startx, starty, endx, rollrange, rolldirection)  .then(function(data){      console.log("data\_3 is :" + data);      TestHelper.assertTrue(data, callback);      callback()  })    //test case 3: get data in in a time range    Timestamp1 ="07/11/16 08:30:00 AM,07/11/16 02:00:00 PM";  apmHighChartPage.getTooltipInfo(Timestamp1,Tagname\_1, false, startx, starty, endx, rollrange, rolldirection)  .then(function(timerange\_data){      console.log("timerange\_data is :" + timerange\_data);      apmHighChartPage.compareData(timerange\_data, "./testRange\_data.csv").then(function (result) {          console.log("result is "+ result)          TestHelper.assertTrue(result, callback);          callback()      })  })    // test case 4: if a tag is not exit in a time stamp, try to get the tooltip information  //             show "no data" under the not exit tag    var Timestamp0 ="07/12/16 08:00:00 AM"  var Tagname\_0 = "EXHAUST\_TEMP\_T48B\_EE-43-TI-11243B\_Actual, test\_teg"  apmHighChartPage.getTooltipInfo(Timestamp0,Tagname\_0, false, startx, starty, endx, rollrange, rolldirection)  .then(function(data){      console.log("data\_1 is "+data)      var int\_1 = data.toString().replace(/\.[0-9]\d\*/g, " ")      TestHelper.assertEqual(int\_1.toString(), "07/12/16 08:00:00 AM, 1344  temperature,no data", callback);      callback()  })    // test case 5: get all data and compare to the data in the file    rollrange= 5  var index\_data = "a"  apmHighChartPage.getApmChartData(startx, starty, endx,index\_data, rollrange, rolldirection)  .then(function(testdata){      //console.log("whole data is "+ testdata);      apmHighChartPage.compareData(testdata, "./original\_data.csv").then(function (result) {          console.log("result is "+ result)          TestHelper.assertTrue(result, callback);          callback()        })  })    // test case 6: find a tooltip tag in the highchart, the mouse will stay in that point and return ture    rollrange = 0;  var index\_data = "07/11/16 06:00:00 AM,EXHAUST\_TEMP\_T48A\_EE-43-TI-11243A\_Estimate: 0 temperature,EXHAUST\_TEMP\_T48B\_EE-43-TI-11243B\_Actual: 1357 temperature,EXHAUST\_TEMP\_T48B\_EE-43-TI-11243B\_Actual\_Smoothed: 1357 temperature,EXHAUST\_TEMP\_T48A\_EE-43-TI-11243A\_Actual\_Smoothed: 1355.5 temperature";  apmHighChartPage.getApmChartData(startx, starty, endx, index\_data, rollrange, rolldirection)  .then(function(testdata){      if(typeof testdata === "boolean"){          TestHelper.assertTrue(testdata, callback);          callback();      }  }) |

For OO Chart

|  |
| --- |
| this.Then(/^I should be able to see all the OO DashBoard chart's details$/, function (callback) {      //apmHighChartPage.checkChartPage().then(function () {      console.log("Header loaded. Proceeding to set chart");      // Setting the Highchart to retrieve information from it        console.log("After setting section")        function getTimestamp(n) {          var date = new Date().getTime()          console.log("date is" + date)          var date\_now = date + n\*1000          var new\_date = new Date(date\_now).toUTCString().replace(/ GMT$/, "")          console.log("new\_date is" + new\_date);          var p1 = new\_date.split(",");          p1 = p1[1];          console.log("p1 is" + p1);          p1 = p1.split(" ");          console.log("p1 is" + p1.toString());          var res = p1[p1.length - 1] + " +0000" + p1[2] + " " + p1[1] + "  " + p1[3];          console.log("res is" + res.toString())          return res;      }  // defining a timestamp that is 15 seconds and 16 seconds later      var timestamp\_1 = getTimestamp(15)      var timestamp\_2 = getTimestamp(16)        var time\_stamp = timestamp\_1+","+ timestamp\_2      var endx = 600      var startx = 310      var starty = 450      var loopTime = 3      var rollrange = 5      var rolldirection ="down"       //testcase 1: expecting the get data action to stop at the time stamp       ooHighChartPage.getOOChartData(time\_stamp, loopTime, startx, starty, endx, rollrange, rolldirection)  .then(function (data) {          console.log("data is " + data)          TestHelper.assertTrue((data.length > 0));          callback()      });        rollrange = 5      var timestamp\_3 = getTimestamp(550)      var timestamp\_4 = getTimestamp(551)      time\_stamp = timestamp\_3+","+ timestamp\_4    //testcase 2: expecting the get data action to stop after maximum loop times      ooHighChartPage.getOOChartData(time\_stamp, loopTime, startx, starty, endx, rollrange, rolldirection)  .then(function (data) {          console.log("data is " + data)          TestHelper.assertTrue((data.length > 0));          callback();      });      }); |

**RestHelper**

RestHelper provides basic REST API call support, like GET, POST, DELETE operations. Here is an example of using RestHelper to create a user and retrieve the user and finally delete the user. Note that RestHelper  requires .***env***file in the run time path (the root directory of your project) to pickup the proxy configuration environment variable, e.g. httpProxy=http://sjc1intproxy01.crd.ge.com:8080.

*.*

**Create and delete a user using RestHelper**

|  |
| --- |
| var myStepDefinitionsWrapper;  myStepDefinitionsWrapper = function () {      'use strict';        var fs = require('fs');      var rStream = fs.createReadStream(path.resolve(\_\_dirname, '../TestData/user.json'));      var data = '';      rStream.setEncoding('utf8');      rStream.on('data', function(chunk) {          data+=chunk;      });      rStream.on('end', function() {          Logger.info('Read user data');          Logger.info(data);      });        var buffer = new Buffer('ingestor.7fd4b91593c34c0e8f4bac2084131a79:');      var base64s = buffer.toString('base64');      Logger.info(base64s);        var adminUserOpts = {          url: '<https://4ebc5a24-c5ec-481e-a371-57ce006de141.predix-uaa-staging.grc-apps.svc.ice.ge.com/oauth/token'>,          headers:{              'Authorization': 'Basic ' + base64s,              'Content-Type': 'application/x-www-form-urlencoded'          },          body:{              'grant\_type' : 'password',              'username' : 'acs0516-admin',              'password' : 'test'          }      }        this.Given(/^A user is created$/, function (callback) {          RestHelper.getAccessToken(adminUserOpts.url, adminUserOpts.headers, adminUserOpts.body,  function(err, token){              var opts = {                  url: '<http://apm-user-management-qa.grc-apps.svc.ice.ge.com/services/v1/umgmt/users'>,                  headers: {                      'Content-Type': 'application/json',                      'tenant':'4D37997D31354D36A40FBBF9AADB4BE0',                      'Authorization':'bearer ' +  token                  },                  body : data              };              RestHelper.executePostRequest(opts.url, opts.headers, opts.body, function(err, res){                  Logger.info("Created user : " + JSON.stringify(res.body));                  console.log("Created user : " + JSON.stringify(res.body));                  callback();              });          });      });        this.Then(/^I can retrieve the user and delete the user$/, function (callback) {          RestHelper.getAccessToken(adminUserOpts.url, adminUserOpts.headers, adminUserOpts.body,   function (err, token) {              //console.log("TOKEN=" + token);              var opts = {                  url: '<http://apm-user-management-qa.grc-apps.svc.ice.ge.com/services/v1/umgmt/users'>,                  headers: {                      'Content-Type': 'application/json',                      'tenant': '4D37997D31354D36A40FBBF9AADB4BE0',                      'Authorization': 'bearer ' + token                  }              };              var url = '[http://apm-user-management-qa.grc-apps.svc.ice.ge.com/services/v1/umgmt/users?username=tobedeleted'](http://apm-user-management-qa.grc-apps.svc.ice.ge.com/services/v1/umgmt/users?username=tobedeleted%27);              RestHelper.executeGetRequest(url, opts.headers, function (err, res) {                  var uuid = JSON.parse(JSON.stringify(res.body)).dataObject[0].uuid;                  console.log('Found user : ' + uuid);                  Logger.info('Found user : ' + uuid);                  RestHelper.executeDeleteRequest(opts.url + '/' + uuid, opts.headers, function (err, res) {                      Logger.info('Deleted user : ' + uuid);                      console.log('Deleted user : ' + uuid);                        callback();                  });              });          });      });  };  module.exports = myStepDefinitionsWrapper; |

**RestHelper for getting Highchart series data:**

This is an example for getting Highcharts series data by utilizing the RestHelper from ProUI. The data will be saved in "RestData.csv" and you can use this file to compare with the data you retrieve from the tooltip. Currently, the tooltip information is affected by the size of the chart, so make sure to modify the size of chart to render correct tooltip information when comparing with the "RestData.csv".

|  |
| --- |
| var myStepDefinitionsWrapper;  myStepDefinitionsWrapper = function () {      'use strict';        var fs = require('fs');      var rStream = fs.createReadStream(path.resolve(\_\_dirname, '../TestData/user.json'));      var data = '';      rStream.setEncoding('utf8');      rStream.on('data', function(chunk) {          data+=chunk;      });      rStream.on('end', function() {          Logger.info('Read user data');          Logger.info(data);      });        var buffer = new Buffer('ingestor.12f037011a4d4892903b9cb8f1d0d446:');      var base64s = buffer.toString('base64');      Logger.info(base64s);        var adminUserOpts = {          url: '<https://91069a10-02cc-41e3-a325-e34b2539905a.predix-uaa.run.asv-pr.ice.predix.io/oauth/token'>,          headers:{              'Authorization': 'Basic ' + base64s,              'Content-Type': 'application/x-www-form-urlencoded'          },          body:{              'grant\_type' : 'password',              'username' : '4EAE92DABCF643C4BACF9D976B5A2E81\_ingestor',              'password' : 'Pa55w0rd'          }      }        this.Given(/^All information provided and I will generate all the data$/, function (callback) {          RestHelper.getAccessToken(adminUserOpts.url, adminUserOpts.headers, adminUserOpts.body,  function(err, token){              console.log("token is " + token)                    var opts = {                  url: '<https://apm-timeseries-services-qa2.run.asv-pr.ice.predix.io/v1/data'>,                  headers: {                      'tenant': '4EAE92DABCF643C4BACF9D976B5A2E81',                      'Authorization': 'Bearer ' + token                  }              };              var url = '[https://apm-timeseries-services-qa2.run.asv-pr.ice.predix.io/v1/data?operation=interpolated&responseFormat=KairosDB&tagList=AVD-CHART-ASSET-CHILD02-ID.EXHAUST\_TEMP\_T48A\_EE-43-TI-11243A\_Actual\_Smoothed\_id,AVD-CHART-ASSET-CHILD02-ID.EXHAUST\_TEMP\_T48B\_EE-43-TI-11243B\_Actual\_Smoothed\_id,AVD-CHART-ASSET-CHILD02-ID.EXHAUST\_TEMP\_T48B\_EE-43-TI-11243B\_Actual\_id,AVD-CHART-ASSET-CHILD02-ID.EXHAUST\_TEMP\_T48A\_EE-43-TI-11243A\_Estimate\_id&startTime=2016-07-11T05%3A00%3A00.000&endTime=2016-07-14T05%3A00%3A00.000&sampleCount=300'](https://apm-timeseries-services-qa2.run.asv-pr.ice.predix.io/v1/data?operation=interpolated&responseFormat=KairosDB&tagList=AVD-CHART-ASSET-CHILD02-ID.EXHAUST_TEMP_T48A_EE-43-TI-11243A_Actual_Smoothed_id,AVD-CHART-ASSET-CHILD02-ID.EXHAUST_TEMP_T48B_EE-43-TI-11243B_Actual_Smoothed_id,AVD-CHART-ASSET-CHILD02-ID.EXHAUST_TEMP_T48B_EE-43-TI-11243B_Actual_id,AVD-CHART-ASSET-CHILD02-ID.EXHAUST_TEMP_T48A_EE-43-TI-11243A_Estimate_id&startTime=2016-07-11T05%3A00%3A00.000&endTime=2016-07-14T05%3A00%3A00.000&sampleCount=300%27);              RestHelper.executeGetRequest(url, opts.headers, function (err, res) {                  //console.log(JSON.stringify(res.body.sample\_size))                        var jres = JSON.stringify(res.body)                  //console.log(jres);                  console.log(JSON.parse(jres).queries.length)                  var len = JSON.parse(jres).queries.length                    var res = new Array()                  for(var i = 0; i< len;){                      var test = JSON.parse(jres).queries[i].results[0].values                      var name = JSON.parse(jres).queries[i].results[0].name                      var testlen = test.length                      for(var a = 0; a<testlen;){                          test[a][1] = name+":"+ test[a][1]                          a++                          if(a === testlen){                              if(res.length === 0){                                  res = test                                  i++                                  if(i === len){                                      console.log(res)                                      callback()                                  }                              }else{                                  for(var j = 0; j<testlen;){                                      var index = test[j].indexOf(res[j][0])                                      test[j].splice(index, 1)                                      res[j] = res[j].concat(test[j])                                      j++                                      if( j === testlen){                                          i++                                          if(i === len){                                              console.log(res)                                              for(var k = 0; k< res.length;){                                                  res[k][0] += 7\*60\*60\*1000                                                  res[k][0] = new Date(res[k][0]).toLocaleString()                                                  res[k][0] = res[k][0].replace(/,/g, "")                                                  k++                                                  if(k === res.length){                                                      var finalres = res.join("\n")                                             fs.writeFile("./RestData.csv", finalres, function (err) {                                                          if (err)console.log(err);                                                      })                                                      callback()                                                  }                                              }                                            }                                      }                                  }                              }                          }                      }                      }                  });            });      });  };  module.exports = myStepDefinitionsWrapper; |

Protractor Image Comparison (protractor-image-comparison)

<https://github.com/wswebcreation/protractor-image-comparison>

Protractor Image Comparison is a node module library which allows image comparison on screens or elements.

You can:

* save or compare screens/elements against a baseline
* save or comparea full page screenshot against a baseline
* automatically create a baseline when no baseline is there
* disablecssanimationsbydefault
* ignore anti-aliasing differences
* compare images by ignoring their colors (do a grayscale comparison)
* blockoutcustomregionsduringcomparison (all)
* increasetheelementdimenisionsscreenshots(all)
* provide custom iOS and Android offsets for status-/address-/toolbar (mobile only)
* automaticallyexcludeastatusbarduringscreencomparison(mobile only)

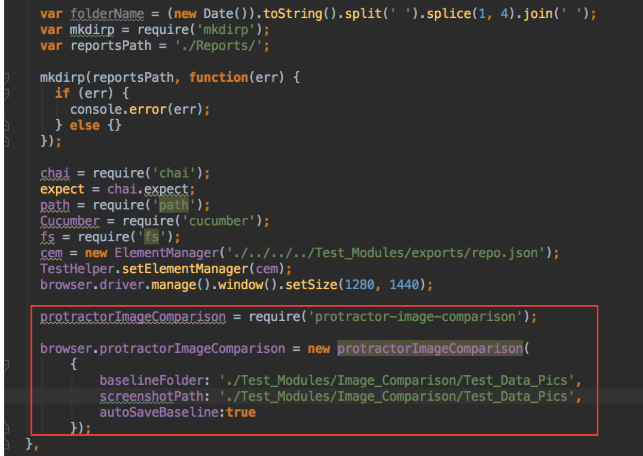
An example is in the ProUI Github inside the Test\_Modules/Image\_Comparison folder

**Steps to use:**

1) Download the node\_module

|  |
| --- |
| npm install protractor-image-comparison --save |

2) Initialize an instance inside your protractor conf.js file



3) Use it in your step definitions

|  |
| --- |
| this.Given(/^I check the image at the homepage$/, function (callback) {        //Navigate to element you want to compare      TestHelper.sendKeys('Predix\_ProUI\_TestPage', 'Predix\_ProUI\_Object\_0', 'neoteric-matrix').then(function () {       TestHelper.sendKeys('Predix\_ProUI\_TestPage', 'Predix\_ProUI\_Object\_1', 'neo@2017').then(function () {       TestHelper.elementToBeClickable('Predix\_ProUI\_TestPage', 'Predix\_ProUI\_Object\_2').then(function () {                  TestHelper.isElementPresent(element(by.css('.pxh-drawer-header'))).then(function () {                        //Provide the element that needs to be checked, and a name for the image      browser.protractorImageComparison.checkElement(element(by.css('.pxh-drawer-header')), 'test')  .then(function (percentDifferent) {                            //The function returns a promise with the percentage that shows difference                          console.log("The difference: " + percentDifferent)             //Asserting if the percentage is 0 meaning that there are no differences between the two images                          TestHelper.assertEqual(0, percentDifferent, callback)                          callback();                      });                  });              });          });      });  }); |

For the full list of methods, please visit:

[https://github.com/wswebcreation/protractor-image-comparison/blob/master/docs/index.md](https://github.com/wswebcreation/protractor-image-comparison/blob/master/docs/index.md#saveElement)

**Constructing Tests**

Conf File

**\*\*\*Take note where these three items are in the config file\*\*\***

1. **Suites:**(Where each team will define their own suite and add their own feature files)
2. **Page Objects:** (Where new page objects are defined and used globally)
3. **CucumberOpts:**(Where step definition files are defined)

|  |
| --- |
| /\*\*   \* Created by 212556710 on 5/9/16.   \*/  exports.config = {       // ---- While testing locally     //seleniumAddress: ' <http://127.0.0.1:4444/wd/hub',>       sauceUser: null,     sauceKey: null,     sauceSeleniumAddress: null,       directConnect: false,     firefoxPath: null,       // ---------------------------------------------------------------------------     // ----- What tests to run ---------------------------------------------------     // ---------------------------------------------------------------------------       // Spec patterns are relative to the location of this config.     specs: [],       // Patterns to exclude.     exclude: [],       // Organize spec files into suites. To run specific suite, --suite=<name of suite>     suites: {   asset: ['../Features/apmLogin.feature','../../Test\_Modules/Assets/Features/asset-example.feature'],          chart: ['../../Team\_Example/Chart/Features/wfvChartExample.feature'],          test: ['../../Team\_Example/Tub/Features/test.feature'],          highchart: ['../../Test\_Modules/Highchart/Features/Highchart.feature'],          rest: ["../../Test\_Modules/RestAPIExample/Features/rest-example.feature"]    },       // Hooks running in the background     plugins: [{        path: '../../node\_modules/proui-utils/Utils/GeneralHook.js',     }],       // Browser options     multiCapabilities: [        // {        // browserName: 'internet explorer',        // platform: 'ANY',        // version: '11'        // },        // {        // browserName: 'firefox',        // },        {           browserName: 'chrome',            proxy : {                          proxyType:'manual',                      httpProxy: 'sjc1intproxy01.crd.ge.com:8080',                          sslProxy: 'sjc1intproxy01.crd.ge.com:8080',           count: 1,           shardTestFiles: false,           maxInstances: 1,           'chromeOptions': {              args: ['--no-sandbox', '--test-type=browser'],              // Set download path and avoid prompting for download even though              // this is already the default on Chrome but for completeness              prefs: {                 'download': {                    'prompt\_for\_download': false,                    'directory\_upgrade': true,                    'default\_directory': 'C:/Jenkins/sharedspace/public/test/e2e-reboot/steps/'                 }              }           }        }       ],       params: {        env: 'dev'     },       maxSessions: -1,       allScriptsTimeout: 250000,       // How long to wait for a page to load.     getPageTimeout: 650000,       // Before launching the application     beforeLaunch: function () {     },       // Application is launched but before it starts executing     onPrepare: function () {          // Create reports folder if it does not exist        var folderName = (new Date()).toString().split(' ').splice(1, 4).join(' ');        var mkdirp = require('mkdirp');        var reportsPath = "./Reports/";          mkdirp(reportsPath, function (err) {           if (err) {              console.error(err);           } else {           }        });          browser.manage().deleteAllCookies();        browser.manage().timeouts().pageLoadTimeout(50000);        browser.manage().timeouts().implicitlyWait(50000);        browser.driver.manage().window().setSize(1280, 1440);          chai = require('chai');        expect = chai.expect;        path = require('path');        Cucumber = require('cucumber');        fs = require('fs');          // Initializing page object variables        loginPage = require('../PageObjects/apm-login-po.js');        loginSpec = require('../Specs/login-spec.js');        apmlandingPage = require('../PageObjects/apm-landing-po.js');        wfvPage = require('../PageObjects/wfv-po.js');        assetTenantPage = require('../PageObjects/asset-tenant-po.js');          // Initializing necessary utils from ProUI-Utils module        TestHelper = require('ProUI-Utils').TestHelper;        ElementManager = require('ProUI-Utils').ElementManager;        Logger = require('ProUI-Utils').Logger;        cem = new ElementManager('../../../Common\_Modules/common-element-repo.json');        TestHelper.setElementManager(cem);          RestHelper = require('ProUI-Utils').RestHelper;          commonTestData = require('../TestData/common-test-data.json').data;     },       // A callback function called once tests are finished     onComplete: function () {       },         // A callback function called once tests are cleaning up     onCleanUp: function (exitCode) {       },       // A callback function after tests are launched     afterLaunch: function () {       },    // Browser parameters for feature files.     params: {        login: {           baseUrl: '<https://apm-qa.grc-apps.svc.ice.ge.com/tenant0609/'>,           "username": "tenant0609-smokeuser",           "password": "test",        }     },       resultJsonOutputFile: null,       // If true, protractor will restart the browser between each test.     // CAUTION: This will cause your tests to slow down drastically.     restartBrowserBetweenTests: false,       // Custom framework in this case cucumber     framework: 'custom',     frameworkPath: require.resolve('protractor-cucumber-framework'),     cucumberOpts: {          // define your step definitions in this file        require: [           '../Specs/env.js',           '../Specs/login-spec.js',           '../../Test\_Modules/Assets/Specs/\*',           '../../node\_modules/proui-utils/Utils/Reporter.js',           '../../Test\_Modules/Highchart/Specs/\*',              '../../Test\_Modules/RestAPIExample/Specs/rest-example-step-def.js'        ],          format: 'pretty'     }  }; |

Feature File Example (ApmLogin)

|  |
| --- |
| @regression  Feature: TS20:User Can Login to APM as Admin      Scenario Outline: US7355:Login as admin and view Asset Management      Optional description goes here        Given I login to "<env>"      When  I authenticate with valid "<user\_name>" and "<password>"      Then  I should see the landing page        Examples:        | env | user\_name | password |        | baseUrl | username  | password | |

\*\*\* **TS\_\_\_**  is equal to Test Set which the Test Cases will be linked to. **US\_\_\_** is equal to User Story which the Test Cases will be linked to. Both fields are optional. \*\*\*

\*\*\* If using Rally Integration your feature file may not contain special characters (# $ % "  ') except for Scenario Outline variables, which must be defined with quotation marks: **"<env>"** \*\*\*

Asset-Step-Def

|  |
| --- |
| /\*\*   \* Main step definition. Add all glue code here in this class   \*/  var myStepDefinitionsWrapper;  myStepDefinitionsWrapper = function () {      'use strict';          this.Given(/^The application is authenticated$/, function (callback) {            //Using login page object method to check if user has logged in.          loginPage.checkHomePage().then(function (completed) {                //Calling logger to write specific details              Logger.info("Asserting if logged in\n");                // Using the promise returned from checkHomePage as a parameter in TestHelper class to assert if true              TestHelper.assertTrue(completed, 'Not logged in');                //Callback once test is complete              callback();          });      });        this.Then(/^I can select Asset Management and view tenant from context$/, function (callback) {            // Using landing page object method to select assets tab          apmlandingPage.selectAssets().then(function () {                // Using landing page object method to select assets instances              apmlandingPage.selectAssetInstances().then(function () {                    // Using asset page object method to wait for table to load                  assetTenantPage.waitForAssetTenantLabel();                    // Using asset page object method to retrieve data from table headers                  assetTenantPage.getCustomerInformation().then(function (tenantName) {                      Logger.info("Asserting to see Tag Class Name is loaded\n");                        // Asserting if the data retrieved is equal to a specific string                      TestHelper.assertEqual(tenantName, "Tag Class Name", callback);                        // Asserting if tenant name is equal to regression                      assetTenantPage.checkTenantName().then(function (tenantName) {                          var name = tenantName.split("\n");                          Logger.info("Asserting Tenant Name is equal to regression");                          TestHelper.assertEqual(name[3], "regression", callback);                            // Logging out and checking if the user is actually logged out                          apmlandingPage.logoutAPM().then(function () {                              loginPage.checkLogout().then(function (completed) {                                  assert.isTrue(completed, 'Not logged in');                                  console.log("LOGGED OUT");                                  callback();                              });                          });                      });                  });              });          });      });  };  module.exports = myStepDefinitionsWrapper; |

Page Object Example (Asset-Tenant-PO)

|  |
| --- |
| /\*\*   \* Class encapsulating all functions in main landing page   \*/  (function () {      'use strict';          var TIMEOUT = 5000;          var AssetTenantPage = function () {            // Defining currentPage variable for retrieving elements          var currentPage = 'assetTenantPage'            return {                // Retrieving table header from asset instances              getTableHeader: function () {                  var customerInfo= elementManager.findElement(currentPage,'customerInfo');                  return customerInfo.getText();              },                // Retrieving tenant name              checkTenantName: function() {                  var tenantName = elementManager.findElement(currentPage,'tenantName');                  return tenantName.getText();              },                // Waiting for spinner to finish loading table              waitForAssetTenantLabel : function (){                  TestHelper.isElementPresent(currentPage,'tagClassNameTableHeader');              }              };      };        module.exports = new AssetTenantPage();    }()); |

**iFrame Example**

If you have certain elements that are stored in iFrames, hereisanexampleonhowtoapproachit:

|  |
| --- |
| /\*\*   \* Main step definition. Add all glue code here in this class   \*/  var myStepDefinitionsWrapper;  myStepDefinitionsWrapper = function () {        'use strict';      this.Given(/^The application is authenticated$/, function (callback) {          browser.ignoreSynchronization = true;          var test;          browser.driver.get("[http://www.w3schools.com/html/html\_iframe.asp"](http://www.w3schools.com/html/html_iframe.asp)).then(function () {              cem.findElement('geSSOPage', 'username').sendKeys(commonTestData.sso);              cem.findElement('geSSOPage', 'password').sendKeys(commonTestData.password);              cem.findElement( 'geSSOPage', 'password').sendKeys(protractor.Key.ENTER)                    //Switching to an iFrame locator which is defined in object repository                  TestHelper.iFrameSwitch('testDialogPage', 'iframe').then(function(){                      console.log("Switched container")                      callback();              });      });      });        this.Then(/^I can select Asset Management and view tenant from context$/, function (callback) {              //After switching to iFrame, I can do anything with the elements inside that iFrame          TestHelper.getText('testDialogPage','headerCheck').then(function(text){              console.log("Here is the text: "+ text);                //When finished using iFrame, switch back to default content              browser.switchTo().defaultContent();              callback();              });          });    };  module.exports = myStepDefinitionsWrapper; |

**The key note to take from this is:**

1) Establish the iFrame locator in your object repository

2) Use "TestHelper.IFrameSwitch(page, element)" to switch to your iFrame locator

3) When the window has been switched to iFrame, you can continue using regular methods.

4) After you are done with the iFrame and want togobacktoregularscreen, use "browser.switchTo().defaultContent()"

**Drag and Drop Example**

If you have certain elements that you would like to drag and drop, hereisanexampleonhowtoapproachit:

|  |
| --- |
| this.When(/^the user drags a tag to a new chart area$/, function (callback) {      console.log('\_\_\_\_\_\_\_\_Dragging to a new chart area')        // Checking to see if the list of item headers to drag are present      TestHelperPO.isElementPresent(element.all(by.css('li.asset-list-item h4')).get(0)).then(function () {          element.all(by.css('li.asset-list-item h4')).get(0).getText()              .then(function (txt) {                  tagname = txt;                  console.log(txt);              })            // Checking to see if the list of items to drag are present          TestHelperPO.isElementPresent(element.all(by.css('li.asset-list-item')).get(0)).then(function () {                // Checking to see if container that the items will be dragged to are present.              TestHelperPO.isElementPresent(element(by.css('#analysis-drop-container > div:nth-child(1)'))).then(function () {                    // Defining the element that's going to be dragged                  var field = (element.all(by.css('li.asset-list-item')).get(0));                    // Defining the container where the elements will be dragged to.                  var src = element(by.css("div[class^='drop-target drop-zone-new-chart']:nth-child(1)"));                    // Calling the drag script by passing in the two elements we defined above.                 TestHelperPO.dragAndDrop(field, src).then(function () {                        // Checking to see if the elements were actually dragged or not.                      analysisPage.checkForPlottedData().then(function (result) {                          TestHelperPO.isElementPresent(result).then(function () {                              callback();                          })                      })                  })              })          })      })  }); |

**Rally Integration**

**Overview**

Rally Integration is a module currently used inside the ProUI framework, which allows users to run functional front-end testing. This feature allows users to automate the process of posting to rally by uploading results after tests are run.

**Use Cases Covered**

**1. Test Set (variable)**  
      a. Test Set has two ways it can be customized in the .env file, and one in the feature file.

1. Associate one test set for all Feature Files w/o a test set already defined.
2. Only post the test cases that have test sets defined – used when a user only wants a specific TestSet to contain the results (Release/Version updates).
3. Add TSXXXX before the Feature File name to add a test set to a specific test folder.

**2. User Story**  
      a. Similar to test set, User Story has two ways it can be customized

1. Associate a user story for all Feature File Test Cases w/o a user story already defined.
2. Add USXXXX: before the Scenario name to add a user story to a specific test case.

**3. Test Case ID**  
      a. Test Case ID can be only customized in the feature file.

1. Utilized if the test case is already created manually in rally and you want to associate a specific scenario to that test case.
2. Add TCXXXX: before the Scenario name to add scenario to a specific test case.

**4. Empty Step File**  
      a. This use case allows all test folders and test cases to be posted to rally, without displaying any results. This will occur on a user’s first run of their tests on ProUI if no step files have been defined.

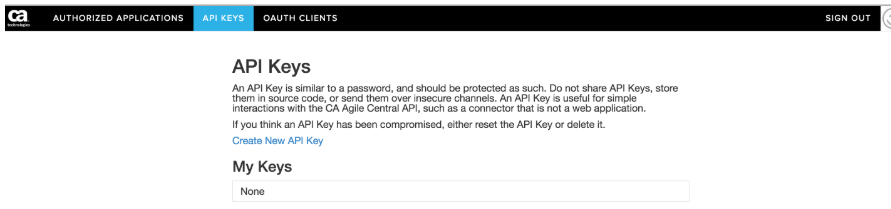
**5. API Key**

1. The API key is parameterized in the .env file. Each user will create their own API key, which will allow Rally to associate an owner to the test cases created.
2. If no api key is provided, there will be an error message indicating that an API key needs to be provided

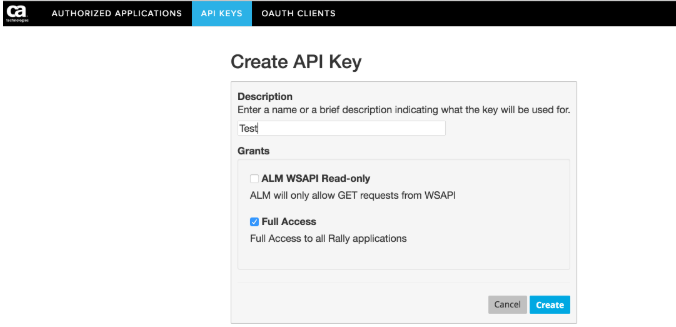
**Steps to generate API Key**

1) Log onto Rally

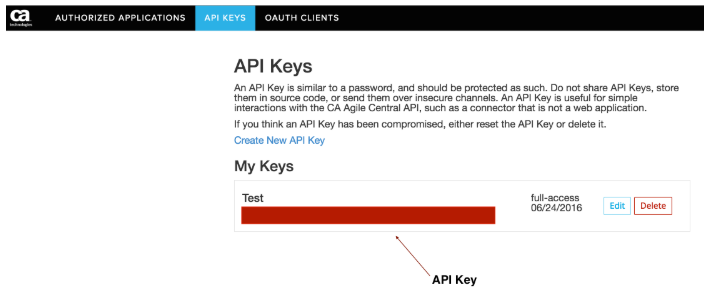
2) Navigate to <https://rally1.rallydev.com/login/accounts/index.html#/keys>



3) Click on "Create new API Key"



4) Write any description and click create



5) Copy and paste the API key into your .env file (located in root folder)

**External Data Parameterization**

1. **.env file**

Note: The syntax for all parameters must use the exact parameter names (case sensitive), and avoid putting spaces between the ‘=’  
The .env file has the following customizable parameters:

if you have same variable set in system variable, please remove them in system variable in order to make the variable in .env file work.

**rallyProxy**

* Default: [http://502635365:RIU.8272.ge@PITC-Zscaler-Americas-](http://502635365:RIU.8272.ge@pitc-zscaler-americas-/) [Alpharetta3pr.proxy.corporate.ge.com](http://alpharetta3pr.proxy.corporate.ge.com/):80
* If a user wants to use the Rally Integration feature outside the GE Network, the proxy must be changed or set to an empty string ‘’.

**postToRally**

* Default: false
* To enable the feature, postToRally must be changed to true

**~~postToRallyWithoutTS~~ (Deprecated. No longer needed)**

* ~~Default: false~~
* ~~If a user wants to post all results regardless of TestSet being defined, change postToRallyWithoutTS=true~~

**apiKey**

* Default: N/A
* User specific key needed to post to Rally.

**user\_story**

* Default: US0000 (dummy user story variable)
* If user\_story is defined in the .env file, all test cases without user stories associated with them will have this user story appended. The user story must already exist in Rally in order to be appended.

**testset**

* Default: TS0000 (dummy test set variable)
* If testset is defined in the .env file, all test cases without test sets associated with them will have this test set appended to them. The test set must already exist in Rally in order to be appended.

**rallyworkspace**

* Example: GE Digital
* Workspace needs to be defined in order for Rally to know where to post results to.

**project**

* QE Automation Engineering (dummy test set variable)
* This is your specific team's name. This also needs to be defined in order for Rally to know where to post results to.  
    
  **type**
* Regression
* This is a parameter to label your tests with a specific type

**overwrite\_US**

* Boolean
* This parameter when set to true, will overwrite the User Story number defined in the feature files with the provided user\_story env parameter.

**overwrite\_TS**

* Boolean
* This parameter when set to true, will overwrite the Test Set number defined in the feature files with the provided testset env parameter.

**overwrite\_RF**

* Boolean
* This parameter when set to true, will overwrite the rally root folder defined in the feature files with the provided rallyrootfolder env parameter.

**overwrite\_PN**

* Boolean
* This parameter when set to true, will overwrite the project name defined in the feature files with the provided project env parameter.

**overwrite\_result**

* Boolean
* This parameter when set to true, will overwrite the latest results for the testcases in rally instead of keeping a history of all runs.

**rallyrootfolder**

* Default: N/A
* Example (Q4Release)
* User needs to specify the root folder name and folder structure so all the test cases could be placed in.

**tag**

* Default: N/A
* Example (Automatable)
* User can specify a tag he/she wants to associate with each test case. \*\*\*Can only specify one tag at a time\*\*\*

**method**

* Default: Automated
* Example (Manual)
* This is the type of method. It can either be Automated or Manual

2.**Feature file**

**For Root Folder:**

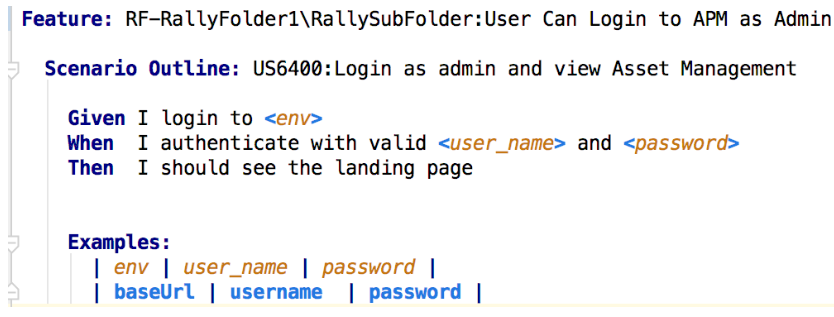
In the feature file if you define the feature name as  :

Feature**:**RF-RallyFolder1\RallySubFolder:User Can Login to APM as Admin,

   Your test result will be saved in the RallyFolder1 -> RallySubFolder and rallyrootfolder defined in the .env file will not work on this feature

   Besides, if you want to provide a specific user story for each scenario, you can put the user story before the scenario description, it will not be affected by the user\_story set in the .env file if the overwrite\_US set to false

  Here is an example:



**For TestSet:**

Youcan define the feature file as:

Feature: **TSXXXX:RF-RallyFolder1\RallySubFolder:User Can Login to APM as Admin**

  Your test  result will be associated with TestSet XXXX

**For Project Name:**

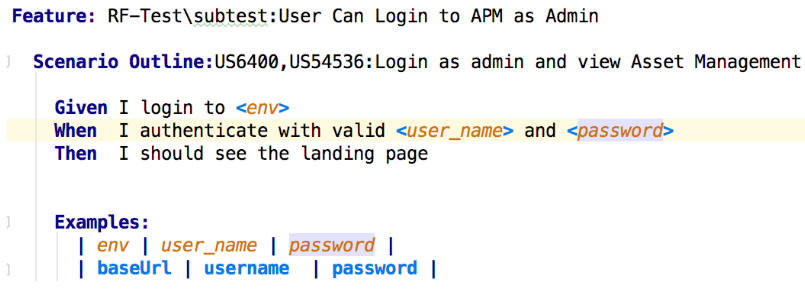
  Youcan define the feature file as:

Feature: **PN-XXXX:TSXXXX:RF-RallyFolder1\RallySubFolder:User Can Login to APM as Admin**

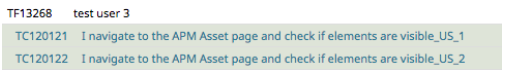
 The test result for this feature will be saved into the project name defined in this feature file.

**For User Story:**

  Youcan define the feature file as:



  User Story 6400 and User Story 54536 will be associated with the test case. In rally we will create a test case for each user story:

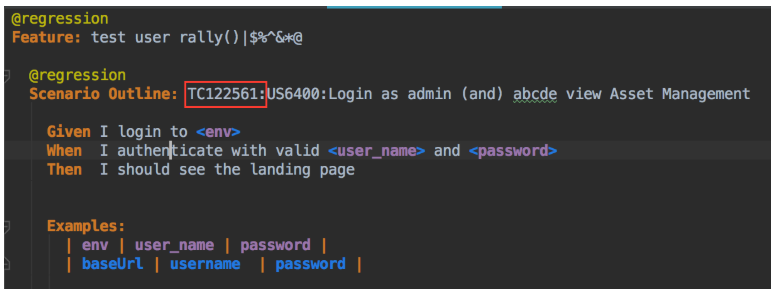


**For Test Case ID:**

  Youcan define the feature file as:

  Scenario: **TCXXXX:Scenario Name**

 The test result for this scenario will be saved into the specific Test Case Number provided.



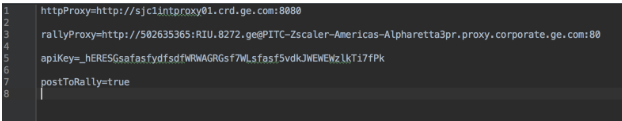
**Steps to enable this feature**

1) Navigate to .env file (located in root directory)

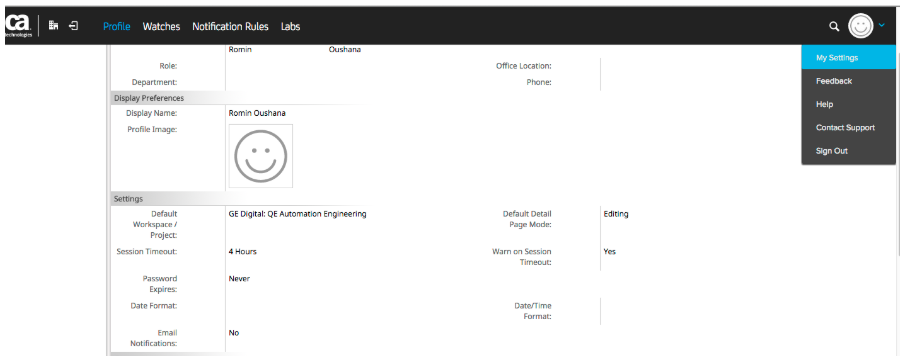
\*\*\*\* If you don't have a .env file in your project root, just create one \*\*\*\*

2) Set postToRally to "true"

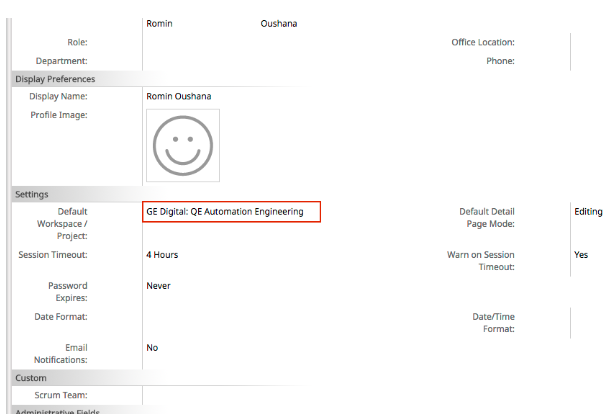
3) Make sure to use your own API Key



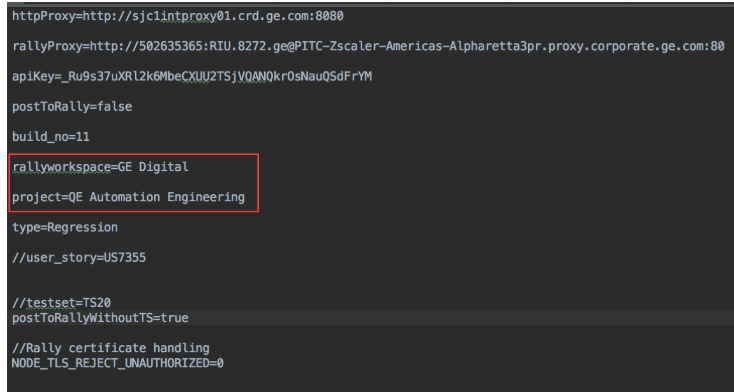
4) Make sure to set your default workspace in rally. To do so, open up your rally webpage. Click on the profile picture and in the dropdown, click My Settings.



5) Scroll down to Settings section and change the Default Workspace/Project to your rally project team.



6) Add the rally workspace and project parameters in the .env file like this:



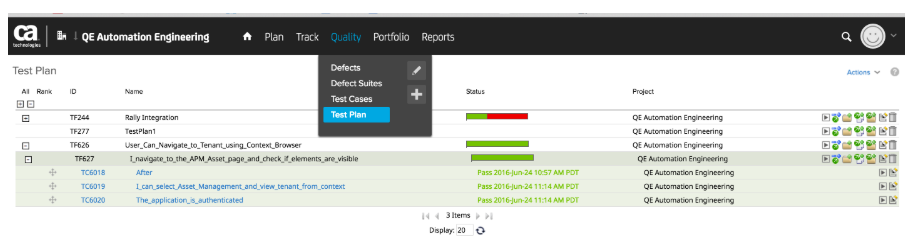
7) Run any of your tests using the above grunt commands.

8) View results on Rally.

**Viewing Results on Rally After Test Execution**

1) Log onto Rally

2) Click Quality and Test Plan

**Jasmine Rally Configuration**

You can follow the example configuration on the ProUI Github.

1) Make sure to include this code block inside the jasmine configuration in the OnPrepare method:

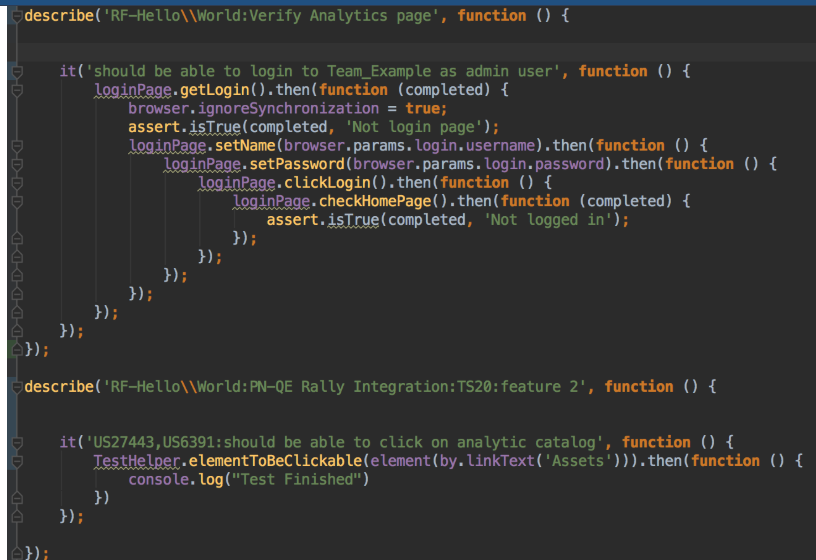
|  |
| --- |
| var jasmineReporters = require('jasmine-reporters');  jasmine.getEnv().addReporter(new jasmineReporters.JUnitXmlReporter({      consolidateAll: true,      savePath: '/Users/212556710/Desktop/Automation Framework/pushProUI/ProUI/Reports',      filePrefix: 'jasmine'    })); |

2) Make sure to include this code block inside the jasmine configuration in the OnComplete method:

|  |
| --- |
| // Creating a promise to run this synchronously  var deferred = protractor.promise.defer();  protractor.promise.controlFlow().execute(function () {        // Calling the Jasmine reporter post to rally function      require('proui-utils').JasmineReporter.postToRally().then(function () {          deferred.fulfill();      })  });  return deferred; |

3) Since Jasmine does not have feature files, treat Features as describes and Scenarios as it statements.

Ex.



**Rally Debugging**

**1. Check the rallylog.log inside the Reports folder**

a. See if rally threw any errors.

**2. Is your default workspace set to the same space you want to post to?**

a. To set your default workspace, navigate to Rally. Click on the profile icon located on the right side of the page. Navigate to ‘My Settings’. Scroll down to the section: ‘Settings’ and change the default workspace/project to the location you want the results posted.

**3. Did you generate your own api key with full access?**  
a. Navigate to:<https://rally1.rallydev.com/login/accounts/index.html#/keys> to generate your ApiKey with full access

**4. Does the .env file have postToRally=true?**

**Windows Configuration**

For windows, if you want to be able to use the Highchart Utility, you have to install Microsoft C++ Build Tools.

1) Once you install <http://landinghub.visualstudio.com/visual-cpp-build-tools>

2) Then type:

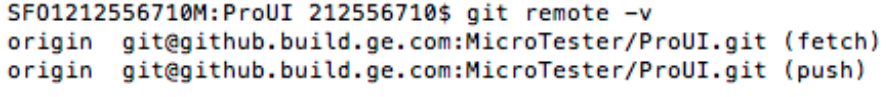
|  |
| --- |
| npm config set --global msvs\_version 2015 |

**Internet Explorer Configuration**

* The IEDriverServer executable must be [downloaded](http://selenium-release.storage.googleapis.com/index.html) and placed in your [PATH](http://en.wikipedia.org/wiki/PATH_(variable)).
* On IE 7 or higher on Windows Vista or Windows 7, you must set the Protected Mode settings for each zone to be the same value. The value can be on or off, as long as it is the same for every zone. To set the Protected Mode settings, choose "Internet Options..." from the Tools menu, and click on the Security tab. For each zone, there will be a check boxatthebottomof the tab labeled "Enable Protected Mode".
* Additionally, "Enhanced Protected Mode" must be disabled for IE 10 and higher. This option is found in the Advanced tab of the Internet Options dialog.
* The browser zoom level must be set to 100% so that the native mouse events can be set to the correct coordinates.
* For IE 11 *only*, you will need to set a registry entry on the target computer so that the driver can maintain a connection to the instance of Internet Explorer it creates. For 32-bit Windows installations, the key you must examine in the registry editor isHKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE\_BFCACHE. For 64-bit Windows installations, the key isHKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE\_BFCACHE. Please note that the FEATURE\_BFCACHE subkey may or may not be present, and should be created if it is not present. Important: Inside this key, create a DWORD value named iexplore.exe with the value of 0.

**Pushing to own Repository/branch**

Once you clone ProUI,yourremoteurlwillbesetto our Microtester repository.



We want to keep this repository as clean as possible because it will be used for example purposes. Therefore, you have to switch to your own repository. Just follow these steps to in order to do so:

|  |
| --- |
| git remote rm origin  git remote add origin <url of your repo> |

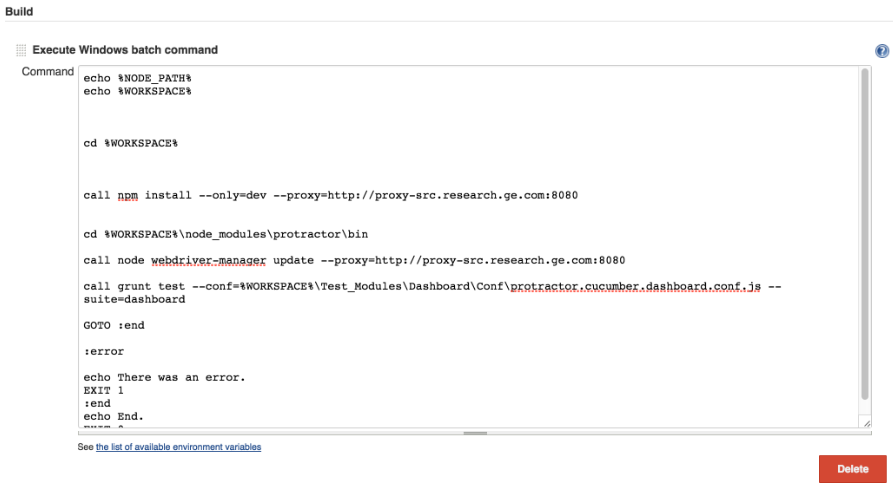
Once you have set your repo, you should do a push.

|  |
| --- |
| git push origin <branch name> |

**Jenkins Configuration**

In the configuration, add a windows batch command and put:

|  |
| --- |
| cd %WORKSPACE%  call npm install --only=dev --proxy=http://sjc1intproxy02.crd.ge.com:8080    cd %WORKSPACE%\node\_modules\protractor\bin  call node webdriver-manager update --proxy=http://sjc1intproxy02.ge.com:8080  call grunt test --conf= <path\_of\_config> --suite= <suite\_name>  GOTO :end  :error  echo There was an error.  EXIT 1  :end  echo End.  EXIT 0 |



To publish HTML reports:

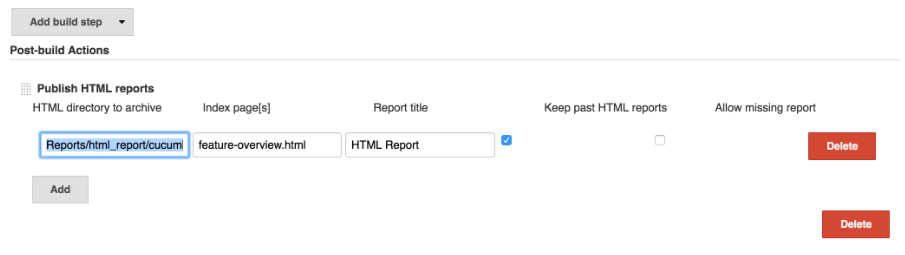
Go into Post-build Actions and add "Publish HTMLreports"

**HTML directory to archive:**

|  |
| --- |
| Reports/html\_report/cucumber-html-reports |

**Index Page:**

|  |
| --- |
| feature-overview.html |



\*\*\*\*IMPORTANT\*\*\*\*

 If you want your browser to pop up on your Jenkin's machine while you are running your test, make sure to have these settings for your browser:

|  |
| --- |
| capabilities: {            'browserName': 'chrome',            'chromeOptions': {                        // This is the key. Make sure "--no-sandbox" is not part of your args              'args': ['--disable-web-security', '--ignore-autocomplete-off-autofill']            }      }, |

**Key Notes**

* Cucumber is no longer included by default as of version 3.0. You can integrate Cucumber with Protractor with the custom framework option.

Change framework from “cucumber” to “**custom**” and add the new framework path

* If you are creating a new feature file, make sure you **DO NOT have the same wording** as other feature files because it will through an ambiguous error.
* Protractor sometimes does not work well with polymer components so if necessary, use old methods.

**var**objManager = ***require***(**'ProUI-Utils'**).**ObjectManager**;  
**var**objectManager = **new**objManager();

[Like8 people](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI) like this

* [rally](https://devcloud.swcoe.ge.com/devspace/label/KKQTL/rally)
* [integration](https://devcloud.swcoe.ge.com/devspace/label/KKQTL/integration)
* [Edit Labels](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI)

11 Comments

[**Prema, Raghi (Raghi P)**](https://devcloud.swcoe.ge.com/devspace/display/~105034975)

* + In the Key Notes "Protractor sometimes does not work well with polymer components so if necessary, use old methods" - Which "Old methods" is being referred here?
  + [Reply](https://devcloud.swcoe.ge.com/devspace/pages/replycomment.action?commentId=1107200790&pageId=1094052848)
  + [Like](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI)
  + [Aug 26, 2016](https://devcloud.swcoe.ge.com/devspace/display/KKQTL/ProUI?focusedCommentId=1107200790#comment-1107200790)

[**Oushana, Romin Bob**](https://devcloud.swcoe.ge.com/devspace/display/~212556710)

Hi Raghi. Sorry for a late reply and not being clear. What I meant by old methods was, use selenium web-driver methods rather than using protractor. Protractor is built on top of selenium to better support angular applications. However, it may not work perfectly with polymer components, so you would have to use regular selenium methods. Thanks again for the comment.

* + - [Reply](https://devcloud.swcoe.ge.com/devspace/pages/replycomment.action?commentId=1124549101&pageId=1094052848)
    - [Like](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI)
    - [Nov 09, 2016](https://devcloud.swcoe.ge.com/devspace/display/KKQTL/ProUI?focusedCommentId=1124549101#comment-1124549101)

[**Khandewale, Sandeep**](https://devcloud.swcoe.ge.com/devspace/display/~502620637)

Hi, This is awesome documentation and interesting piece of software. While I am just getting introduced to this, Its just today I have got dashboard tests app working. However I would like to suggest a small change, (Which I have tried and it works for me)

For some dependencies (I think robotjs framework) It says Visual Studio is needed. But IMO it is not needed. IMHO it should not be needed. (On windows platform) robotjs postinstall script will try to build a vc++ project and it will need c++ compiler and linker only. There is a distribution of this tool-chain MS provides its called "Microsoft C++ Build Tools." Just install that and it will work. If you are using 2015 version of this dist, make sure you do npm config set --global msvs\_version 2015.

More information can be found on <https://github.com/nodejs/node-gyp#installation>

again, respect and compliments for this kind of work!

Cheers,

sandeep

* + [Reply](https://devcloud.swcoe.ge.com/devspace/pages/replycomment.action?commentId=1123744947&pageId=1094052848)
  + [Like](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI)
  + [Nov 08, 2016](https://devcloud.swcoe.ge.com/devspace/display/KKQTL/ProUI?focusedCommentId=1123744947#comment-1123744947)

[**Oushana, Romin Bob**](https://devcloud.swcoe.ge.com/devspace/display/~212556710)

Hi Sandeep. Thank you very much for your comment! I will try this out and update the changes.

* + - [Reply](https://devcloud.swcoe.ge.com/devspace/pages/replycomment.action?commentId=1124549083&pageId=1094052848)
    - [Like](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI)
    - [Nov 09, 2016](https://devcloud.swcoe.ge.com/devspace/display/KKQTL/ProUI?focusedCommentId=1124549083#comment-1124549083)

[**Sykes, Benjamin James (Ben)**](https://devcloud.swcoe.ge.com/devspace/display/~108010087)

Please make this github repo public read.

* + [Reply](https://devcloud.swcoe.ge.com/devspace/pages/replycomment.action?commentId=1124806384&pageId=1094052848)
  + [Like](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI)
  + [Nov 14, 2016](https://devcloud.swcoe.ge.com/devspace/display/KKQTL/ProUI?focusedCommentId=1124806384#comment-1124806384)

[**Khanna, Kapil**](https://devcloud.swcoe.ge.com/devspace/display/~212452470)

Repo is Public now.

* + - [Reply](https://devcloud.swcoe.ge.com/devspace/pages/replycomment.action?commentId=1128204300&pageId=1094052848)
    - [Like](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI)
    - [Nov 30, 2016](https://devcloud.swcoe.ge.com/devspace/display/KKQTL/ProUI?focusedCommentId=1128204300#comment-1128204300)

[**Kumaran, RAJEESH Palakkalparambu**](https://devcloud.swcoe.ge.com/devspace/display/~212605158)

I am facing web driver time out issue with direct Connect : false, however its working when direct Connect is true.

I can use the above solution for chrome but for ie direct connect is not supported.

Can some one help

* + [Reply](https://devcloud.swcoe.ge.com/devspace/pages/replycomment.action?commentId=1206804377&pageId=1094052848)
  + [Like](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI)
  + [Nov 07, 2017](https://devcloud.swcoe.ge.com/devspace/display/KKQTL/ProUI?focusedCommentId=1206804377#comment-1206804377)

[**Khanna, Kapil**](https://devcloud.swcoe.ge.com/devspace/display/~212452470)

**You need to remove/comment out directConnect:**

exports.config = { multiCapabilities:[ { 'browserName': 'internet explorer' } ], ... }

directConnect: true - Lets script communicates directly to Chrome Driver or Firefox Driver, bypassing any Selenium Server. If this is true, settings for seleniumAddress and seleniumServerJar will be ignored.

**If you attempt to use a browser other than Chrome or Firefox an error will be thrown**.

* + - [Reply](https://devcloud.swcoe.ge.com/devspace/pages/replycomment.action?commentId=1207716798&pageId=1094052848)
    - [Like](https://devcloud.swcoe.ge.com/devspace/pages/viewpage.action?spaceKey=KKQTL&title=ProUI)
    - [Nov 09, 2017](https://devcloud.swcoe.ge.com/devspace/display/KKQTL/ProUI?focusedCommentId=1207716798#comment-1207716798)

[**Vadrevu, Ravindra**](https://devcloud.swcoe.ge.com/devspace/display/~502710906)

**Please help as we are stuck on this**

We are facing an issue with File Upload component while creating test suites for one of the microapps. There is already one file upload component used in one of the existing pages, which is tested through the TestHelper.uploadFile() method.

We have created an additional file upload component in a modal panel. When we use the TestHelper.uploadFile() method, it is uploading the file in the first component instead the one in the modal panel.

**Error or Warning:** W/element - more than one element found for locator By(css selector, input[type="file"]) - the first result will be used

As we see internally the TestHelper.uploadFile method is using $('input[type="file"]') for locating the web element (file upload) and in case of multiple upload components in the UI, it is not working.

We also tried various approaches for doing file upload. Below are few of them:

var uploadElement = element(by.xpath("//div[@class='email-case-modal ng-scope']/..//following-sibling::div//following-sibling::div//following-sibling::evidence-upload//a/div"));  
var uploadElement = element(by.xpath("/html/body/label[2]/input"));

1. uploadElement.sendKeys(absolutePath)  
2. uploadElement.getWebElement().sendKeys(absolutePath)  
2. TestHelperPO.elementToBeClickable(uploadElement.getWebElement())  
3. browser.actions().mouseMove(uploadElement).click().perform()  
4. browser.driver.actions().mouseDown(uploadElement.getWebElement()).click().perform()  
5. browser.actions().mouseMove(uploadElement).click().perform()

and then

var path = require('path');  
var robot = require('robotjs');  
var fileToUpload1 = filePath;  
var absolutePath = path.resolve(fileToUpload1);  
robot.typeString(filePath);  
browser.sleep(3000).then(function () {  
robot.keyTap("enter");  
deferred.fulfill(browser.sleep(10000));  
});

For the above options neither the click / sendKeys seems to be working.