Creating test case guide

Muhammad U. Bari

2015

Table of Contents

[Summary 2](#_Toc438650522)

[Copy template test case file 2](#_Toc438650523)

[Creating workspace 2](#_Toc438650524)

[Deleting workspace 2](#_Toc438650525)

[Importing a Master Data 2](#_Toc438650526)

[Creating a calculation measure 2](#_Toc438650527)

[Deleting a calculation measure 2](#_Toc438650528)

[Using graph and metric filters 2](#_Toc438650529)

[Creating and validating Dashboard graph 2](#_Toc438650530)

[Deleting Dashboard graph 2](#_Toc438650531)

[Creating and validating Dashboard metric 2](#_Toc438650532)

[Deleting Dashboard metric 2](#_Toc438650533)

[Loading and validating opportunity graph 2](#_Toc438650534)

[Loading and validating measure graph 2](#_Toc438650535)

# Summary

This file is a guide on how to create a test case for the Automation framework.

# Copy template test case file

* Open eclipse (if not opened)
  + Make sure the project is up to date. If not, right click the project > Team > Pull
* Open templates package in test folder
* Copy TestCaseTemplate.java and paste it in testCases package
* Rename the copied file to a reasonable name.

# Creating workspace

* Open the test case file in testCases package
* Go to test(BrowserType, String) method
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Use the following method/s
  + EsiActivity.loadEsiActivity(WebDriver, String)
  + EsiActivity.createWorkspace(WebDriver, String, String)

# Deleting workspace

* Open the test case file in testCases package
* Go to test(BrowserType, String) method
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Use the following method/s
  + EsiActivity.deleteWorkspace(WebDriver, String)

# Importing a Master Data

* Open the test case file in testCases package
* Go to test(BrowserType, String) method
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Use the following method/s
  + EsiActivity.importMasterData(WebDriver, String)

# Creating a calculation measure

* Open the test case file in testCases package
* Go to test(BrowserType, String) method
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Use the following method/s
  + EsiActivity.createMeasure(WebDriver, String)

# Deleting a calculation measure

* Open the test case file in testCases package
* Go to test(BrowserType, String) method
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Use the following method/s
  + EsiActivity.deleteMeasure(WebDriver, String)

# Using graph and metric filters

* Used for setting up the graph with certain filters before viewing the graph/metric
* Ex: Display a graph of oil production represented in a column graph
  + Use Selenium IDE to record the user’s actions for loading the graph with the filters
  + Based on the target type (xpath, css, link, etc), create an ArrayList <Object []>.
    - Check javadocs for EsiActivity.createDashboardGraph(WebDriver, ArrayList<Object []>) for details
    - Check javadocs for EsiActivity.createDashboardMetric(WebDriver, ArrayList<Object []>) for details

# Creating and validating Dashboard graph

* Open the test case file in testCases package
* Go to test(BrowserType, String) method
* Create a new private method that doesn’t return anything
* Use the following commands:
  + String graphName = <graph name>;
  + String excelFileLocation = <relative path to answer key excel file>;
  + String sheetName = <excel sheet name>;
  + String reporterGraphValidateName = <the name of expand/collapse html section>;
    - Used for displaying the name of the expand/collapse when the report is generated
    - Ex: for validating a column graph in the dashboard for production  
      “Dashboard > Chart > production > column”
  + EsiActivity.goToDashboard(WebDriver)
  + Create an ArrayList <Object []> filters.
    - Check [link](#_Using_graph_and)
  + EsiActivity.createDashboardGraph(WebDriver, String, ArrayList<Object []>)
  + Thread.sleep(WAIT\_TIME)
  + WebElement graph = EsiActivity.getDashboardGraph(WebDriver, String)
  + EsiActivity.maximizeDashboardGraph(WebDriver, graph)
  + EsiActivity.validateGraph(WebDriver, WebElement, String, ChartType, String, String, String, int, int, int)
  + EsiActivity.restoreDashboardGraph(WebDriver, WebElement)
* Go to test(BrowserType, String)
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Call the newly created creation.

# Deleting Dashboard graph

* Open the test case file in testCases package
* Go to test(BrowserType, String) method
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Make sure the workspace is opened and the dashboard graph exists.
* Use the following commands
  + EsiActivity.deleteDashboardGraph(WebDriver, String);

# Creating and validating Dashboard metric

* Open the test case file in testCases package
* Go to test(BrowserType, String) method
* Create a new private method that doesn’t return anything
  + String metricName = <metric name>;
  + String excelFileLocation = <relative path to answer key excel file>;
  + String sheetName = <excel sheet name>;
  + String reporterMetricValidateName = <the name of expand/collapse html section>;
    - Used for displaying the name of the expand/collapse when the report is generated
    - Ex: for validating a column graph in the dashboard for production  
      “Dashboard > Metric > production”
  + EsiActivity.goToDashboard(WebDriver)
  + Create an ArrayList <Object []> filters.
    - Check [link](#_Using_graph_and)
  + EsiActivity.createDashboardMetric(WebDriver, String, ArrayList<Object []>)
  + EsiActivity.validateMetric(WebDriver,String, String, String, String, String, int, int)
* Go to test(BrowserType, String)
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Call the newly created creation.

# Deleting Dashboard metric

* Open the test case file in testCases package
* Go to test(BrowserType, String) method
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Make sure the workspace is opened and the dashboard metric exists.
* Use the following commands
  + EsiActivity.deleteDashboardMetric(WebDriver, String);

# Loading and validating opportunity graph

* Open the test case file in testCase package
* Go to test(BrowserType, String) method
* Create a new private method that doesn’t return anything.
  + String opportunityName = <opportunity name>
  + String excelFileLocation = <relative path to answer key excel file>
  + String sheetName = <excel sheet name>
  + String reporterGraphValidateName = <the name of expand/collapse html section>;
    - Used for displaying the name of the expand/collapse when the report is generated
    - Ex: for validating a column graph in the dashboard for production  
      “Opportunites > <opportunity name> > production”
  + EsiActivity.goToOpportunities(WebDriver)
  + Create an ArrayList <Object []> filters.
    - Check [link](#_Using_graph_and)
  + EsiActivity.loadOpportunityGraph(WebDriver, String, ArrayList<Object []>)
  + Thread.sleep(WAIT\_TIME)
  + WebElement graph = driver.findElement(By.cssSelector(“svg”))
  + EsiActivity.validateGraph(WebDriver, WebElement, String, ChartType, String, String, String, int, int, int)
  + Thread.sleep(WAIT\_TIME)
  + Find the xpath to the button that closes the opportunity and execute it using
    - driver.findElement(By.xpath(<xpath to the button location>)).click();
* Go to test(BrowserType, String)
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Call the newly created creation.

# Loading and validating measure graph

* Open the test case file in testCase package
* Go to test(BrowserType, String) method
* Create a new private method that doesn’t return anything.
  + String measureName = <opportunity name>
  + String excelFileLocation = <relative path to answer key excel file>
  + String sheetName = <excel sheet name>
  + String reporterGraphValidateName = <the name of expand/collapse html section>;
    - Used for displaying the name of the expand/collapse when the report is generated
    - Ex: for validating a column graph in the dashboard for production  
      “Calculation > Measure > <measure name>”
  + EsiActivity.goToCalculations(WebDriver)
  + Create an ArrayList <Object []> filters.
    - Check [link](#_Using_graph_and)
  + EsiActivity.loadMeasureGraph(WebDriver, String, ArrayList<Object []>)
  + Thread.sleep(WAIT\_TIME)
  + WebElement graph = driver.findElement(By.cssSelector(“svg”))
  + EsiActivity.validateGraph(WebDriver, WebElement, String, ChartType, String, String, String, int, int, int)
  + Thread.sleep(WAIT\_TIME)
  + Find the xpath to the button that closes the opportunity and execute it using
    - driver.findElement(By.xpath(<xpath to the button location>)).click();
* Go to test(BrowserType, String)
* After the line that says   
  *driver* = *webi*.instantiateBrowser(browserType);
* Call the newly created creation.