**QA Assignment Documentation**

Table of Contents

[Software’s Requirement 3](#_heading=h.gjdgxs)

[Steps to create a project 4](#_heading=h.30j0zll)

[Steps to Create test automation scripts that perform CRUD operations. 7](#_heading=h.1fob9te)

[Run the program 9](#_heading=h.3znysh7)

[Test Result 10](#_heading=h.2et92p0)

# Software’s Requirement

|  |  |
| --- | --- |
| **Software:** | Eclipse IDE |
| **Windows:** | x86\_64 |
| **Mac:** | x86\_64 |
| **Linux:** | x86\_64 |
| **Plugin:** | TestNG, Maven |
| **Framework:** | TestNG |
| **Tool:** | Selenium Driver |
| **Language** | Java |
| **Browser:** | chromedriver.exe |

# Steps to create a project in your local machine

Step1: Create a new JAVA project and provide the below details:

Graphical user interface, text, application, email

Description automatically generated

Project name: backbase-qa-assignment

Proceed by clicking Next

Graphical user interface, text, application, email

Description automatically generated

Step2: Navigate to Libraries tab and add external JARs of Selenium Client and WebDriver for Java

Graphical user interface, text

Description automatically generated

Step 3: Click on Finish

Step 4: Once the project is created, click on project(backbase-qa-assignment) and create TestNG class

Source Folder: /backbase-qa-assignment/src/test/java

Package Name: backbaseQA

ClassName: crudArticles

And proceed by clicking Finish

A picture containing text, screenshot, computer, indoor

Description automatically generated

Step5: A new java file will be created with the name: crudArticles.java

# 

# Steps to Create test automation scripts that perform CRUD operations.

Step1: Create a handler method that sets driver to ChromeDriver()

***“@Test(priority=1, description="Login into alert popup to access BBlog URL")***

***public void handleAlert() {***

***System.setProperty("webdriver.chrome.driver", "./Drivers/chromedriver.exe");***

***driver= new ChromeDriver();***

***//launch the chrome browser to open the application url and give username and password in alert popup***

***driver.get("https://candidatex:qa-is-cool@qa-task.backbasecloud.com/");***

***//maximize the browser window***

***driver.manage().window().maximize();***

***}”***

***Login to*** [***https://candidatex:qa-is-cool@qa-task.backbasecloud.com***](https://candidatex:qa-is-cool@qa-task.backbasecloud.com) ***by specifying username & Password***

Step2: Login to Conduit with username and password

***Username: testadmin@testadmin.com***

***Password: testadmin***

***@Test(priority=2, description="Login to conduit")***

***public void loginConduit() {***

***//Click on Sign In button***

***driver.findElement(By.xpath("//a[@routerlink='/login']")).click();***

***driver.findElement(By.xpath("//input[@formcontrolname='email']")).sendKeys("testadmin@testadmin.com");***

***driver.findElement(By.xpath("//input[@formcontrolname='password']")).sendKeys("testadmin");***

***driver.findElement(By.xpath("//button[@type='submit']")).click();***

***}***

Step3: Create new article by clicking a navigation link : New article

***P.s Enter tags text area doesn’t work as we can provide value but when we view the article TagName isn’t visible.***

***Created a method to create new article: createNewArticle()***

* ***driver.findElement(By.xpath("//a[@routerlink='/editor']")).click();***
* ***Identify all the input fields in the page and fill in values by using .*** ***sendKeys()***
* ***Publish article by: driver.findElement(By.xpath("//button[normalize-space()='Publish Article']")).click();***

Step4: Read article which we recently created

* ***driver.navigate().to("https://candidatex:qa-is-cool@qa-task.backbasecloud.com"); 🡪 Home Screen Navigation***
* ***Navigate to Global Feeds***
* ***Identify all the article by h1 tags in the page***
* ***And loop through all the identified values to the Title that was created to find the exact article.***
* ***driver.findElement(By.xpath("//a/h1[contains(text(),'Covid-20 Article')]")).click();***

Step5: Update Article

* ***Perform actions same until step 3***
* ***Identify the Edit article button and append text to the existing article***

***Ex:***

***WebElement element = driver.findElement(By.xpath("//textarea[@formcontrolname='body']"));***

***element.sendKeys(Keys.HOME + "Update Body: ");***

Step 6: Delete Article

* ***driver.navigate().to("https://qa-task.backbasecloud.com");*** 🡪 ***Navigate to Home screen***
* ***Create a new article: Following step 3***
* ***Wait until 3 seconds to validate the new article is created***
* ***And proceed with delete Article***
* ***Once the article is deleted, close the window.***
* ***//close the browser***

***driver.close();***

# Run the program

Right Click on,

1. Run the pom.xml file > Maven clean,

2. Run the pom.xml file > Maven test, and finally

2. Run the testng.xml file > "Run as Test Suite"

**Note:** I have converted the project into maven to get supported jar files which includes all dependencies mentioned in pom.xml and created testng.xml to run the scripts.

# Test Result

It can be found in Suite folder : **test-output/Suite/Test.html**

