Contents

Web Application Testing using selenium with java

1. Function to Initialize the driver

```
private void initializeDriver() {
    String browser = System.getProperty("browserName");
    ReportLog.LOG("Launching browser: " + browser);
    System.out.println("Launching driver: " + browser);
    if (browser.toUpperCase().equalsIgnoreCase("IE")) {
        ReportLog.LOG("Setting up Internet explorer options");
        InternetExplorerOptions capabilities = new InternetExplorerOptions();
        // capabilities.setCapability("browser.download.dir","c:\\downloads");
        capabilities.setCapability("ignoreZoomSetting", true);
        capabilities.setCapability(InternetExplorerDriver.INTRODUCE_FLAKINESS_BY_IGNORING_SECURITY_DOMAINS, true);
        capabilities.setCapability(InternetExplorerDriver.NATIVE_EVENTS, false);
        capabilities.setCapability(InternetExplorerDriver.INTRODUCE_FLAKINESS_BY_IGNORING_SECURITY_DOMAINS, true);
        capabilities.setCapability(InternetExplorerDriver.IE_ENSURE_CLEAN_SESSION, true);
        setDriver(new InternetExplorerDriver(capabilities));
        ReportLog.LOG("IE Driver is started");

    } else if (browser.equalsIgnoreCase("chrome")) {
        //System.setProperty("webdriver.chrome.driver","S:\\Technology\\TTGACMX\\QA\\QA All\\Selenium and Eclipse Files\\IEDriverServer_Win32_3.4.0\\
        setDriver(new ChromeDriver());
        ReportLog.LOG("Chrome Driver is started");
```

2. Functions to locate the element, click element, send Input text.

1. Function: - getElements() : Line 62-74. To locate the element by passing the input xpath and returns the list of webelements. An Explicit wait is used to wait until the element is visible. Also logs in the extent report with name passed to function as input.
2. Function: - click() : Line 77-82. To click the element by passing the input xpath and click the element. Also logs in the extent report with name passed to function as input.
3. Function: - setText() : Line 85-90. To pass data i.e. input content to web elements such as text boxes by using the input xpath. Also logs in the extent report with name passed to function as input.

```
60      // to get an element
61      @SuppressWarnings("deprecation")
62      protected List<WebElement> getElements(By by, String controlName) {
63          ReportLog.LOG("Finding elements: " + controlName);
64          wait = new WebDriverWait(getDriver(), Long.valueOf(System.getProperty("explicitwait")));
65          wait.until(ExpectedConditions.visibilityOfElementLocated(by));
66          List<WebElement> elements = getDriver().findElements(by);
67          JavascriptExecutor js = (JavascriptExecutor) getDriver();
68          String script = ("arguments[0].setAttribute('style','border:2px solid " + (System.getProperty("higlightcolor"))
69                  + "');");
70          for (WebElement each : elements) {
71              js.executeScript(script, each);
72          }
73          return elements;
74      }
75
76      // to click on element
77      protected void click(By by, String controlName) {
78          WebElement ele = getElement(by, controlName);
79          ReportLog.LOG("clicking on element: " + controlName);
80          ele.click();
81          ReportLog.LOG("clicked");
82      }
83
84      // to enter data
85      protected void setText(By by, String content, String controlName) {
86          WebElement ele = getElement(by, controlName);
87          ReportLog.LOG("Entering text as:  " + content);
88          ele.sendKeys(content);
89          ReportLog.LOG("Text entered");
90      }
91
```

3.Functions to verify the element present in page, get text from elements, element of dropdown.

1. Function: - IsElementPresent() : Line 92-104. To verify the element present or not on the page by passing the input xpath and return the element if present. Also logs in the extent report with name passed to function as input.
2. Function: - getDataFromElementList() : Line 107-114. To get the text of elements by using the input xpath. Also logs in the extent report with name passed to function as input.
3. Function: - SelectDropdownByText() : Line 117-123. To select any option from the dropdown by using the input xpath and option title. Also logs in the extent report with name passed to function as input.

```
92      // verify element present or not on the page
93      protected boolean IsElementPresent(By by, String controlName) {
94          boolean elementPresent = true;
95          @SuppressWarnings("unused")
96          WebElement ele;
97          ReportLog.LOG("Verifying element availability: " + controlName);
98          try {
99              ele = getElement(by, controlName);
100         } catch (Exception ex) {
101             elementPresent = false;
102         }
103         ReportLog.LOG("Element is available");
104         return elementPresent;
105     }
106
107     // get text from list of elements
108     protected List<String> getDataFromElementList(By by, String controlName) {
109         List<WebElement> list = getElements(by, controlName);
110         List<String> data = new ArrayList<String>();
111         for (WebElement eachElement : list) {
112             data.add(eachElement.getText());
113         }
114         return data;
115     }
116
117     protected void SelectDropdownOptionByText(By by, String optionTitle, String controlName) {
118         Select dropdown = new Select(getElement(by, controlName));
119         Assert.assertNotNull(dropdown, "Dropdown control is null");
120         ReportLog.LOG("Selecting option from dropdown: " + optionTitle);
121         dropdown.selectByVisibleText(optionTitle);
122         ReportLog.LOG("Dropdown is Selected");
123     }
124
```

4. Functions to generate the Random string and Random number.

1.Function: - getRandomString() : Line 125-132. To generate the random string of size equal to the length passed in input.
2. Function: - getRandomNumber() : Line 135-144. To generate the random number of size equal to the length passed in input.

```
124
125     protected String getRandomString(int length) {
126         String AB = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
127         SecureRandom rnd = new SecureRandom();
128
129         StringBuilder sb = new StringBuilder(length);
130         for (int i = 0; i < length; i++)
131             sb.append(AB.charAt(rnd.nextInt(AB.length())));
132         return sb.toString();
133     }
134
135     protected String getRandomNumber(int length) {
136         String AB = "0123456789";
137         SecureRandom rnd = new SecureRandom();
138
139         StringBuilder sb = new StringBuilder(length);
140         for (int i = 0; i < length; i++)
141             sb.append(AB.charAt(rnd.nextInt(AB.length())));
142         return sb.toString();
143     }
144 }
145
```

5. Functions to Verify the page navigation text, Success message and String to date conversion.

1. Function: - verifyPageNavigationText () : Line 47-52. To verify the text by passing the driver and title and using assert to verify the expected and actual. Also logs in the extent report with title passed to function as input.
2. Function: - getSuccessMessage() : Line 77-82. To verify the successmessage by passing the driver and title and using assert to verify the expected and actual. Also logs in the extent report with title passed to function as input.
3. Function: - convertStringToDate() : Line 85-90. To convert the string into the specifies format passed as input and returns converted date.

```
    }

    @SuppressWarnings("deprecation")
    public static void verifyPageNavigationText(WebDriver driver, String title) {
        ReportLog.LOG("Verifying navigation title: " + title);
        WebDriverWait wait = new WebDriverWait(driver, Long.valueOf(System.getProperty("explicitwait")));
        wait.until( ExpectedConditions.visibilityOfElementLocated( byNavigationText));
        WebElement naviagationElement = driver.findElement(byNavigationText);
        Assert.assertTrue(naviagationElement.getText().contains(title), "Expected (contains): " + title + " | Actual: " + naviagationElement.getText());
    }

    @SuppressWarnings("unchecked")
    public static <T extends BasePage> T  clickAddNew(Class<?> cls, WebDriver driver) throws Exception{
        ReportLog.LOG("Clicking on ADD NEW button");
        @SuppressWarnings("deprecation")
        WebDriverWait wait = new WebDriverWait(driver, Long.valueOf(System.getProperty("explicitwait")));
        wait.until( ExpectedConditions.visibilityOfElementLocated( byAddNew));
        driver.findElement(byAddNew).click();

        //creating generic object to return
        Class <?> servicename = Class.forName(cls.getName());
        Constructor<?>[] constructors = servicename.getDeclaredConstructors();
        return (T)constructors[1].newInstance(driver);
    }

    public static String getSuccessMessage(WebDriver driver, String title) {
        ReportLog.LOG("Getting "+ title +" text");
        @SuppressWarnings("deprecation")
        WebDriverWait wait = new WebDriverWait(driver, Long.valueOf(System.getProperty("explicitwait")));
        wait.until( ExpectedConditions.visibilityOfElementLocated( bySuccessMessage));
        WebElement successElement = driver.findElement(bySuccessMessage);
        Assert.assertNotNull(successElement, "Success element is not displayed");
        return successElement.getText();
    }

    public static Date convertStringToDate(String sDate, String sFormate) throws Exception {
        DateFormat df = new SimpleDateFormat(sFormate);
        Date date = df.parse(sDate);
        return date;
    }
}
```

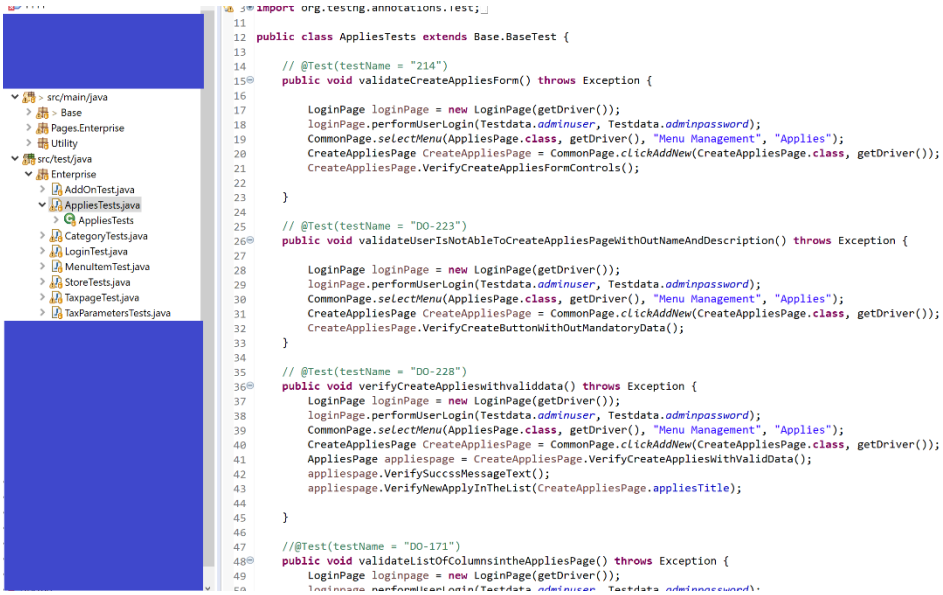
6. Function To take screenshot and save in the destination folder

```
    }

    public static void take_screenshot(String name) throws IOException {
        //take screenshot and store it in a file format
        File file = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
        //copy the screenshot to the destination folder
        String new_date = new SimpleDateFormat("yyyy-mm-dd-HH-mm-ss").format(new Date());
        FileUtils.copyFile(file,new File("user.dir"+name+new_date+".jpg"));

    }
```

7. Function To select a particular date from calendar

```
@Test(priority=3)
public void check_out_date(){
    driver.findElement(By.id("d2-btn")).click();
    String Month = "June 2021";
    String date = "2";


    while(true) {

  String text = driver.findElement(By.xpath("//h2[@class='uitk-date-picker-month-name uitk-type-medium']")).getText();
    //comparing the months with expected month and if condition is true then break
  if(text.equals(Month)) {
        break;
    }
        else {
            driver.findElement(By.xpath("//button[@class='uitk-button uitk-button-small uitk-flex-item uitk-button-paging'][2]")).click();
        }
    }
    //loop to select the date
    List<WebElement> alldates = driver.findElements(By.xpath("//div[@class='uitk-new-date-picker-month'][1]//table[@class='uitk-date-picker-weeks']//tr//td//b
    for(WebElement ele:alldates)
    {
        String date1 = ele.getAttribute("data-day");

        //System.out.println(date1);
        if(date1.equals(date)) {
            ele.click();
            ReportLog.LOG("clicked");
            break;
        }
    }

    driver.findElement(By.xpath("//button[@data-stid='apply-date-picker']")).click();

}
```

8. Tests Page in PageObjectModel

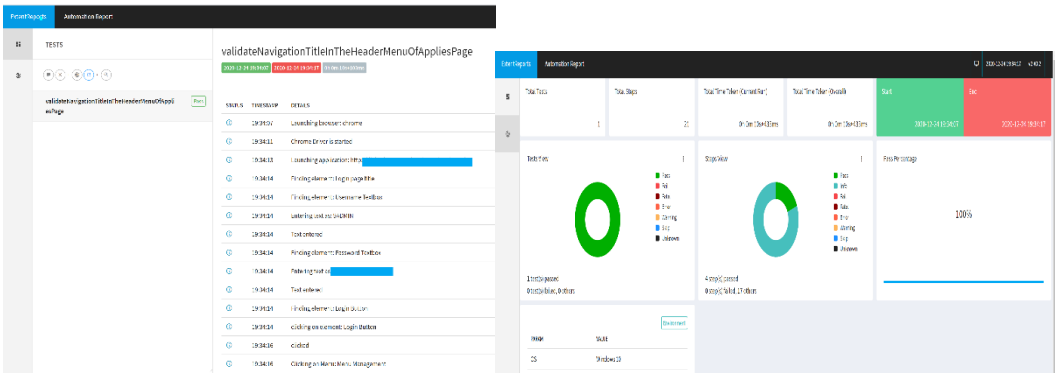



```java
import org.testng.annotations.Test;

public class AppliesTests extends Base.BaseTest {

    // @Test(testName = "214")
    public void validateCreateAppliesForm() throws Exception {

        LoginPage loginPage = new LoginPage(getDriver());
        loginPage.performUserLogin(Testdata.adminuser, Testdata.adminpassword);
        CommonPage.selectMenu(AppliesPage.class, getDriver(), "Menu Management", "Applies");
        CreateAppliesPage CreateAppliesPage = CommonPage.clickAddNew(CreateAppliesPage.class, getDriver());
        CreateAppliesPage.VerifyCreateAppliesFormControls();

    }

    // @Test(testName = "DO-223")
    public void validateUserIsNotAbleToCreateAppliesPageWithOutNameAndDescription() throws Exception {

        LoginPage loginPage = new LoginPage(getDriver());
        loginPage.performUserLogin(Testdata.adminuser, Testdata.adminpassword);
        CommonPage.selectMenu(AppliesPage.class, getDriver(), "Menu Management", "Applies");
        CreateAppliesPage CreateAppliesPage = CommonPage.clickAddNew(CreateAppliesPage.class, getDriver());
        CreateAppliesPage.VerifyCreateButtonWithOutMandatoryData();
    }

    // @Test(testName = "DO-228")
    public void verifyCreateApplieswithvalidata() throws Exception {
        LoginPage loginPage = new LoginPage(getDriver());
        loginPage.performUserLogin(Testdata.adminuser, Testdata.adminpassword);
        CommonPage.selectMenu(AppliesPage.class, getDriver(), "Menu Management", "Applies");
        CreateAppliesPage CreateAppliesPage = CommonPage.clickAddNew(CreateAppliesPage.class, getDriver());
        AppliesPage appliespage = CreateAppliesPage.VerifyCreateAppliesWithValidData();
        appliespage.VerifySuccssMessageText();
        appliespage.VerifyNewApplyInTheList(CreateAppliesPage.appliesTitle);

    }

    //@Test(testName = "DO-171")
    public void validateListOfColumnsintheAppliesPage() throws Exception {
        LoginPage loginpage = new LoginPage(getDriver());
```

9. Extent Report Functions

```java
import java.text.SimpleDateFormat;

public class ReportLog {
    public static ExtentTest extentTest;
    public static ExtentReports extentReport;

    public static void initializeReport() {
        String currentDateTime =  String.valueOf(new SimpleDateFormat("MMddyyyyhhmmss").format(new Date()));
        extentReport = new ExtentReports(System.getProperty("user.dir")+"\\Reports\\ExtentReportResults_"+ currentDateTime +".html");
    }

    public static void startTest(String testTitle) {
        extentTest = extentReport.startTest(testTitle);
    }

    public static void LOG(String message) {
        extentTest.log(LogStatus.INFO, message);
    }

    public static void PASS(String stepName) {
        extentTest.log(LogStatus.PASS, stepName);
    }

    public static void FAIL(String stepName) {
        extentTest.log(LogStatus.FAIL, stepName);
    }

    public static void WARNING(String stepName) {
        extentTest.log(LogStatus.WARNING, stepName);
    }
}
```

10. Report generated using Extend Reports



Rest API test using postman using openweather API

Created a Postman collection that is checking the weather in New York tomorrow

1) Added a test on the max_temperature: if max_temperature > 20degree Celsius, the test should fail.

2)To check the Response code status

3)To check the content type in JSON format

4)To check the response time.