24/03/25

Automation

Client and server communication

Selenium

1. Selenium is a open source automated testing tool
2. Used for **automating web application testing** across different browsers and platforms.
3. Selenium is primarily built using **Java**, but it supports multiple programming languages Ruby, Python, C#, JavaScript, Kotlin
4. Selenium is often integrated with various testing frameworks to enhance test automation, reporting, and maintainability.
   Selenium with Java framework is,
   a. TestNG (Here we are using TestNG)
   b. Cucumber
   c. Junit
   d. Spring boot testing
5. Selenium supports wide range of browsers including
   a. Google chrome
   b. Mozilla Firefox
   c. Microsoft Edge
   d. Safari
   e. Opera
   f. Internet Explorer

Difference between Manual and Automation testing

Manual testing	**Automation testing**
Testing done manually by humans without automation tools.	Testing performed using automation tools/scripts.
Slower, as testers execute test cases manually.	Faster, as scripts run automatically.
Occures human errors.	Highly accurate (if scripts are written correctly).
Test coverage limited due to time constraints.	High test coverage due to fast execution.
Lower initial cost but higher long-term cost due to manual effort.	Higher initial cost (tools, setup) but cost-effective long-term.

Test cases need to be executed manually every time.	Test scripts can be reused multiple times.
Human involvement is high (requires human observation & decision-making).	Human involvement is low (scripts run independently).
Parallel execution is not possible (one tester runs one test at a time).	Parallel execution is possible (multiple tests can run simultaneously).

Advantage and Disadvantage of Selenium

Advantage

1. Open Source , free to use
2. Support multiple browsers
3. Supports CICD tool (Continuous integration and continuous)

Disadvantages

1. Supporting only web application testing

Test Suit component

Selenium is a test suit component

1. Selenium IDE (Selenium integrated development environment)
    - Record and playback tool
    - Firefox chrome plugin (selenium ide only supported with Firefox and Chrome, this is the major disadvantages)
    - Automatic generated TestScript
    - No direct interaction with web element
    - No need programming language
2. Selenium remote control (RC)
    - No longer use Selenium RC
3. Selenium WebDriver
    - Selenium Webdriver is updated version of selenium RC
    - Direct interaction to Webelement
4. Selenium Grid
    - Hub-node architecture
    - Parallelly execute with IDE and Webdriver (Selenium grid is mostly used for parallel execution of multiple browsers or multipls Os)

Architecture of Selenium

2 versions,

1. Selenium 3 (Json wire protocol using data transfer)
2. Selenium 4 (W3C protocol using data transfer)

WebDriver Initializing

Webdriver is a predefined interface class

Global declaration is,

```
Public WebDriver driver;
```

- Chrome corresponds reference class is ChromDriver
- Edge reference class is EdgeDriver

Close and Quit

Close: Close single window at a time

Quit: Close more than one window at a time

Browser launch code

```
public void initialiseBrowser()
        {
                driver=new ChromeDriver();
                driver.get("https://selenium.qabible.in/");
                driver.manage().window().maximize();
        }
```

Launching URL in different browsers

```
1.  public void initialiseBrowser()
        {
                driver=new EdgeDriver();
                driver.get("https://selenium.qabible.in/");
                driver.manage().window().maximize();
        }
```

```
3.  public void initialiseBrowser()
            {
                driver=new MozillaDriver();

                 driver.get("https://selenium.qabible.in/");

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

            }
```

26/03/25

Browser and navigation commands

Browser commands

1. Title (getTitle())
2. URL (getCurrentUrl())
3. window handle (getWindowHandle())
4. page source (getPageSource())

Navigation commands

1. to (navigate().to();)
2. back (navigate().back();)
3. forward (navigate().forward();)
4. refresh (navigate().refresh();)

27/03/25

Web elements

- Any component of a webpage that can be interacted with, such as buttons, text boxes, links, images, checkboxes, dropdowns, etc

DOM structure

Document object model

The DOM represents an HTML document as a tree of nodes:

- The root node is <html>, containing:
    1. <head> (metadata, title, styles)

    2. <body> (visible content of the page)

- Inside <body>, elements like <div>, <p>, <a>, <button>, etc., form child nodes.

HTML tag

Opening tag and Closing tag

Opening tag Eg: <head>, <body>

Closing tag Eg: </head>, </body>

Attribute and Attribute value

Eg:

type="button"

type is Attribute and “Button” is Attribute value

Locators

Locators are used to identify and interact with web elements on a webpage.

- To find single element use **, findelement(By.)**
- To find multiple elements use, **findelements(By.)**
- How to create a Web element? Web element is a interface so,

```
WebElement element name= driver. Findelement(By.id(“attribute value”));
```

Eg: WebElement showmessagebutton= driver.findElement(By.*id*("button-one"));

- By is a Abstract class

- How to locate an Element,

```
driver.findeelement(By.id(“attribute value”));
```

eg: 1. driver.findElement(By.*id* ("button-one"));
2.driver.findElement(By.*className*("btn btn-primary"));

Total 8 locators

1. Id
2. Class namw
3. Tag name
4. Name
5. Linktext
6. Partial linktext
7. Css selector
8. X-path

1. id- driver.findElement(By.*id*("button-one"));
2. Class name- driver.findElement(By.*className*("btn btn-primary"));
3. Tag name- driver.findElement(By.*tagName*("button"));
4. Name- driver.findElement(By.*name*("viewport"));
5. LinkText- driver.findElement(By.*linkText*("Radio Buttons Demo"));
6. Partial linktext- driver.findElement(By.*partialLinkText*("Radio Buttons"));
7. Css selector-

   Providing top to down search

   Syntax:

   ```
   Tag name[attribute='value']
   ```

   Eg:-

   a. driver.findElement(By.*cssSelector*("input[id='single-input-field']"));
   b. driver.findElement(By.*cssSelector*("button[id='button-one']"));
8. X-path –

   Providing both top to down and down to top search

   Syntax:

   ```
   //tagname[@attribute='value']
   ```

   // represents current node.

   X-path finding plugins:- Selectors hub, X-path helper, Firebug, Chropath.

Two types of X-path

1. Relative X-path

   Relative X-path starts from Current node, We are using relative X-path.
2. Absolute X-path

   Absolute X-path starts from Root node

- Using text(linktext) in X-path
  //button[text()='show message']
- If link text is length, we can use starting words,
  //button[starts-with(text(),'show')]
- "and" case

```
driver.findElement(By.xpath("//button[@id='button-one' and @type='button']"));
```

- "or" case
  driver.findElement(By.*xpath*("//button[@id='button-one' or @id='button-one-electronics']"));

X-path axes mathods

1. Parent
2. Child
3. Following
4. Preceding
5. Ancestor
6. Descendant

Eg:

1. Parent:
   driver.findElement(By.*xpath*("//div[contains(text(),'Single Input Field')]//parent::div[@class='card']"));
2. Child:
   driver.findElement(By.*xpath*("//div[@class='card']//child::button[@id='button-one']"));
3. Following:
   driver.findElement(By.*xpath*("//button[@id='button-one']//following::div[@class='card']"));
4. Preceding:
   driver.findElement(By.*xpath*("//button[@id='button-one']//preceding::div[@class='card']"));
5. Ancestor:
   driver.findElement(By.*xpath*("/button[@id='button-one']//ancestor::div"));
6. Descendant:
   driver.findElement(By.*xpath*("//div[@class='card']//descendant::div"));

01/04/25

Webelement Commands

1. SendKeys
2. Click
3. getText
4. Clear
5. getCssValue

1.SendKeys()

Used for input a value to any box (text box)

Eg:

```
WebElement messagebox= driver.findElement(By.id("single-input-field"));

messagebox.sendKeys("Welcome");
```

2. Click()

Eg:

```
WebElement showmessagebutton= driver.findElement(By.id("button-one"));

showmessagebutton.click();
```

3. getText()

Used to print any message or test in console

Eg:

```
WebElement yourmessage= driver.findElement(By.id("message-one"));

System.out.println(yourmessage.getText());
```

4. Clear()

Eg:

```
messagebox.clear();
```

5. getCssValue()

Eg:

String bordercolourofshowmessage= showmessagebutton.getCssValue("border-color");

System.***out***.println(bordercolourofshowmessage);

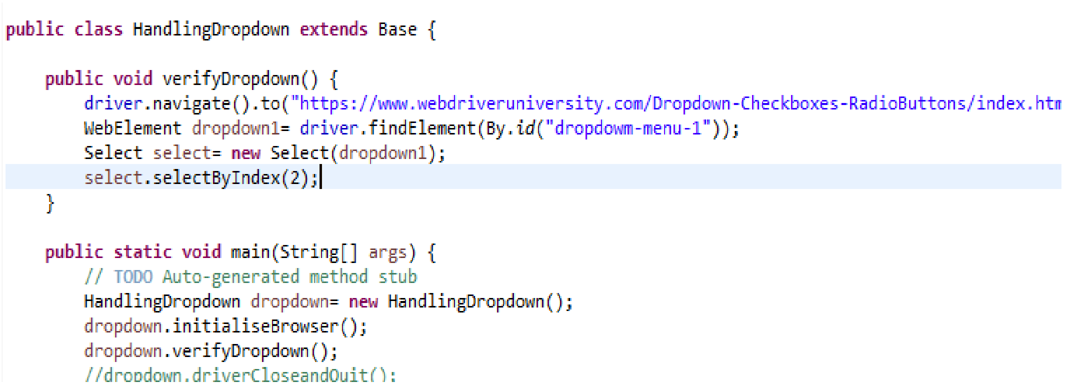
Handling Dropdown

1. Select

Select is a pre-defined class

a. SelectByIndex

Eg:



```
public class HandlingDropdown extends Base {

    public void verifyDropdown() {
        driver.navigate().to("https://www.webdriveruniversity.com/Dropdown-Checkboxes-RadioButtons/index.htm
        WebElement dropdown1= driver.findElement(By.id("dropdowm-menu-1"));
        Select select= new Select(dropdown1);
        select.selectByIndex(2);
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        HandlingDropdown dropdown= new HandlingDropdown();
        dropdown.initialiseBrowser();
        dropdown.verifyDropdown();
        //dropdown.driverCloseandQuit();
```

b. SelectByValue

Eg:

select.selectByValue("python");

c. SeclectByVisible

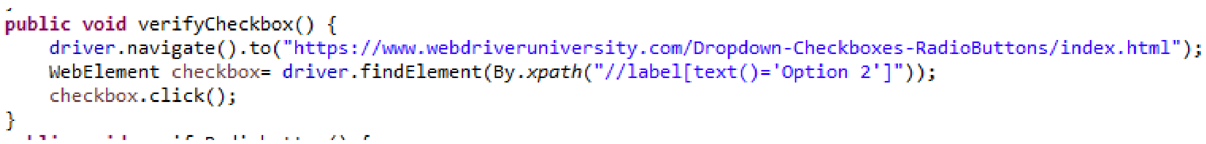
Eg:

select.selectByVisibleText("SQL");

Check Box:

Multiple selection allows at a time

Eg:

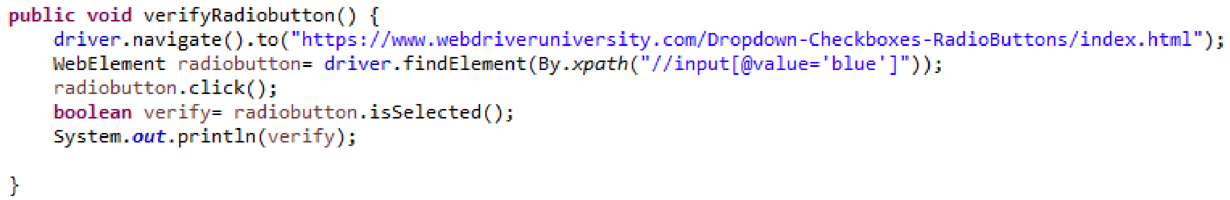


```
public void verifyCheckbox() {
    driver.navigate().to("https://www.webdriveruniversity.com/Dropdown-Checkboxes-RadioButtons/index.html");
    WebElement checkbox= driver.findElement(By.xpath("//label[text()='Option 2']"));
    checkbox.click();
}
```

Radio Button:

Single selection done at a time

Eg:



```
public void verifyRadiobutton() {
    driver.navigate().to("https://www.webdriveruniversity.com/Dropdown-Checkboxes-RadioButtons/index.html");
    WebElement radiobutton= driver.findElement(By.xpath("//input[@value='blue']"));
    radiobutton.click();
    boolean verify= radiobutton.isSelected();
    System.out.println(verify);

}
```

Methods

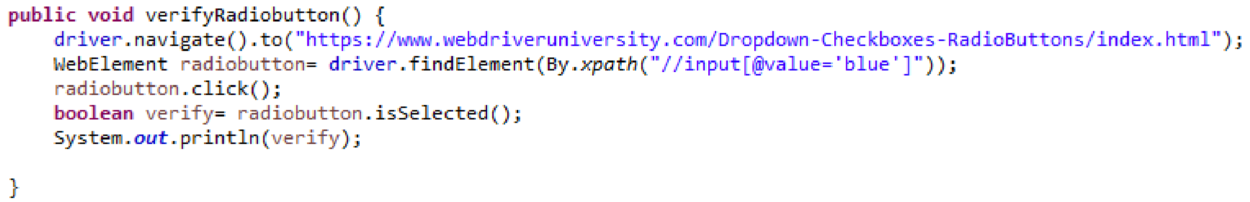
- isEnabled

  Eg:

  System.***out***.println(showmessagebutton.isEnabled());

- isSelected

  Eg:



```
public void verifyRadiobutton() {
    driver.navigate().to("https://www.webdriveruniversity.com/Dropdown-Checkboxes-RadioButtons/index.html");
    WebElement radiobutton= driver.findElement(By.xpath("//input[@value='blue']"));
    radiobutton.click();
    boolean verify= radiobutton.isSelected();
    System.out.println(verify);

}
```

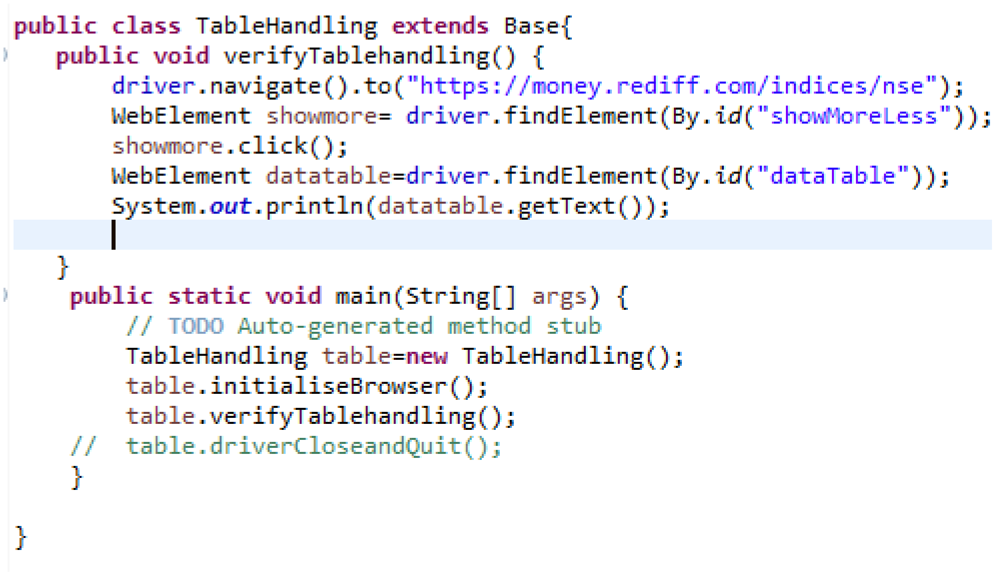
- isDisplayed

  Eg:

  System.out.println(showmessagebutton.isDisplayed());

Table Handling

- Showmore Clicking
- Full data table



```
public class TableHandling extends Base{
   public void verifyTablehandling() {
       driver.navigate().to("https://money.rediff.com/indices/nse");
       WebElement showmore= driver.findElement(By.id("showMoreLess"));
       showmore.click();
       WebElement datatable=driver.findElement(By.id("dataTable"));
       System.out.println(datatable.getText());
       
   }
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        TableHandling table=new TableHandling();
        table.initialiseBrowser();
        table.verifyTablehandling();
    //  table.driverCloseandQuit();
    }

}
```