[FalconUI]

[Documentation]

Index:

..

Class: Browser

Methods:

1.public boolean captureScreenshot()
2.public boolean clearAllCookies()
3.public boolean clearCookieByName(String cookieName)
4.public boolean closeBrowser()
5.public CheckBox getCheckBox()
6.public Click getClick()
7.public String getCurrentPageTitle()
8.public String getCurrentURL()
9.public WebDriver getDriver()
10.public Find getFindFromBrowser()
11.public Key getKey()
12.public String getLoadedPageSource()
13.public MouseMove getMouse()
14.public Navigate getNavigate()
15.public PageHelper getPageHelper()
16.public Scroll getPageScroll()
17.public WebTable getTable()
18.public TextField getTextField()
19.public WaitElement getWait()
20.public WindowHandler getWindowHandler()
21.public boolean maximizeWindow()
22.public boolean openPage(String url, BrowserType browserType)
23.public boolean openPage(String url, BrowserType browserType, String profileName)
24.public void openPageCommons(String url)
25.public boolean openURL(String url, String os, String osVersion, String browser, String browserVersion)
26.public boolean openURL(String url, String os, String osVersion, String browser, String browserVersion,DesiredCapabilities capabilities)
27.public boolean quitBrowser()
28.public boolean refreshBrowser()
29.private void setupDriverForRequiredClasses()
30.setupDriverForRequiredClassesForParallel()
31.private boolean uploadImage(File file)
32.private boolean validateURL(String url)

Method Name:

public boolean captureScreenshot()

Purpose:

This method will take Screenshot of browser.This method is used to upload screenshot of the web site to the report server whenever the test method is failed.

Returns:

Returns True if Successful or returns False

Usage:

```
browser.captureScreenshot();
```

Method Name:

public boolean clearAllCookies()

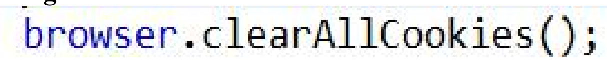
Purpose:

This method will clear all browser Cookies.

Returns:

Returns True if Successful or returns False

Usage:



```
browser.clearAllCookies();
```

Method Name:

public boolean clearCookieByName(String cookieName)

Purpose:

This method will clear browser Cookies as per cookies name provided to this method.

Returns:

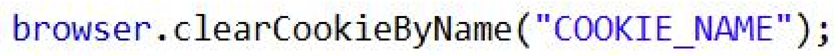
Returns True if Successful or returns False

Throws:

NullPointerException

Usage:

I have a cookie with name “COOKIE_NAME”
Go to settings -->(left side) settings-->Advance
-->Privacy and Security-->Content Setting-->Cookies



```
browser.clearCookieByName("COOKIE_NAME");
```

Method Name:

public boolean closeBrowser()

Purpose:

This method will close the browser.

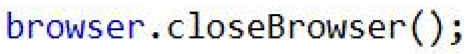
Returns:

Returns True if Successful or returns False

Throws:

NullPointerException

Usage:



```
browser.closeBrowser();
```

Method Name:

public CheckBox getCheckBox()

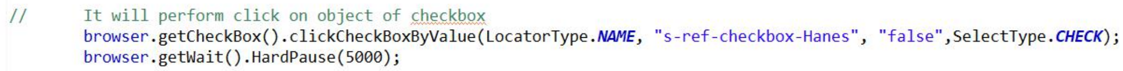
Purpose:

This method is to get the CheckBox class object from Browser class.This class is used to select checkbox By CssSelector, Id, Value.It is also used to check checkbox is selected or not.

Returns:

Returns CheckBox Object.

Usages:



```
//      It will perform click on object of checkbox
        browser.getCheckBox().clickCheckBoxByValue(LocatorType.NAME, "s-ref-checkbox-Hanes", "false",SelectType.CHECK);
        browser.getWait().HardPause(5000);
```

Method Name:

public Click getClick()

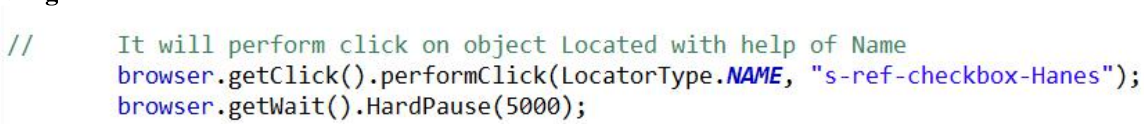
Purpose:

This method is to get the Click class object from Browser class.Also Click class is used to click button by name, image by xpath, double click on element by xpath and by id, right click on element by xpath.

Returns:

Returns Click Object.

Usages:



```
//      It will perform click on object Located with help of Name
        browser.getClick().performClick(LocatorType.NAME, "s-ref-checkbox-Hanes");
        browser.getWait().HardPause(5000);
```

Method Name:

public String getCurrentPageTitle()

Purpose:

This method will get the title of currently opened page

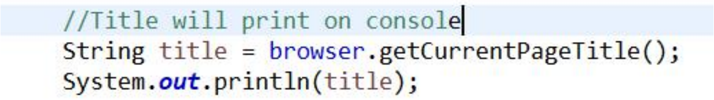
Throws:

NullPointerException.

Returns:

Returns Title of page if found if not then Throws Exception.

Usage:



```
        //Title will print on console
        String title = browser.getCurrentPageTitle();
        System.out.println(title);
```

Method Name:

public String getCurrentURL()

Purpose:

This method will get the currently opened page URL.

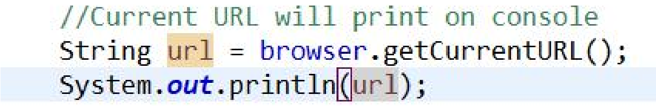
Throws:

NullPointerException.

Returns:

Returns Title of page if found if not then Throws Exception.

Usage:



```
        //Current URL will print on console
        String url = browser.getCurrentURL();
        System.out.println(url);
```

Method Name:

public WebDriver getDriver()

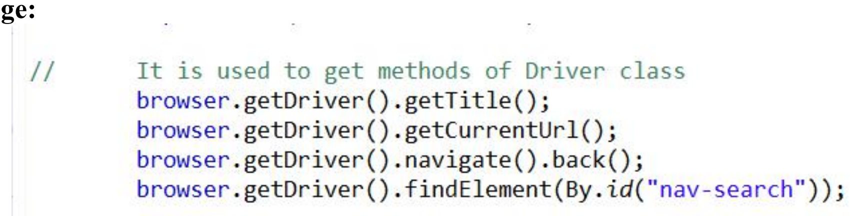
Purpose:

This method is to get the driver object from Browser class.

Returns:

Returns current driver object

Usage:



```
//      It is used to get methods of Driver class
        browser.getDriver().getTitle();
        browser.getDriver().getCurrentUrl();
        browser.getDriver().navigate().back();
        browser.getDriver().findElement(By.id("nav-search"));
```

Method Name:

public Find getFindFromBrowser()

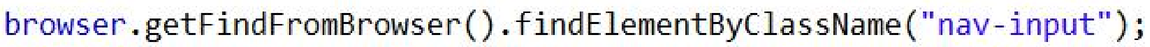
Purpose:

This method is to get the Find class object from Browser class.

Returns:

Returns find object

Usage:



```
browser.getFindFromBrowser().findElementByClassName("nav-input");
```

Method Name:

public Key getKey()

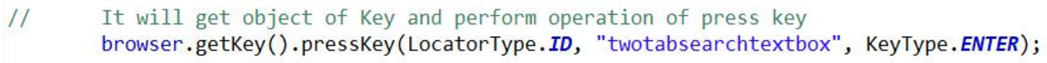
Purpose:

Method is to get the Tab class object from Browser class

Returns:

Returns key object

Usage:



```
//      It will get object of Key and perform operation of press key
        browser.getKey().pressKey(LocatorType.ID, "twotabsearchtextbox", KeyType.ENTER);
```

Method Name:

public String getLoadedPageSource()

Purpose:

This method will Load page source.

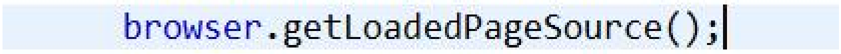
Throws:

NullPointerException.

Returns:

Returns current page loaded source if found if not then Throws Exception.

Usage:



```
        browser.getLoadedPageSource();
```

Method Name:

public MouseMove getMouse()

Purpose:

This method is to get the MouseMove class object from Browser class

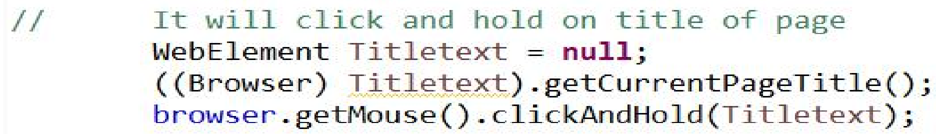
Throws:

NullPointerException

Returns:

Returns mouseMove object

Usage:



```
//      It will click and hold on title of page
        WebElement Titletext = null;
        ((Browser) Titletext).getCurrentPageTitle();
        browser.getMouse().clickAndHold(Titletext);
```

Method Name:

public Navigate getNavigate()

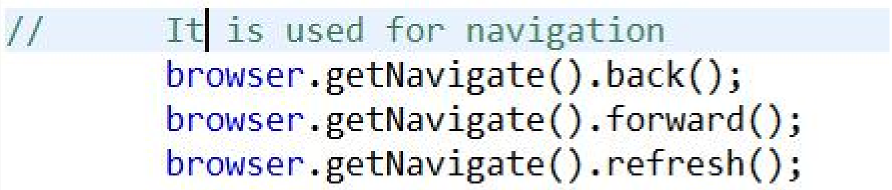
Purpose:

This method is to get the Navigate class object \n from Browser class

Returns:

Returns navigate object

Usage:



```
//      It is used for navigation
        browser.getNavigate().back();
        browser.getNavigate().forward();
        browser.getNavigate().refresh();
```

Method Name:

public PageHelper getPageHelper()

Purpose:

This method is to get the PageHelper class object from Browser class. PageHelper class provides utilities for a web page such as getElementsAttributeList(Returns list of WebElements Attribute by given 'By' instance).

Returns:

Returns pageHelper object

Method Name:

public Scroll getPageScroll()

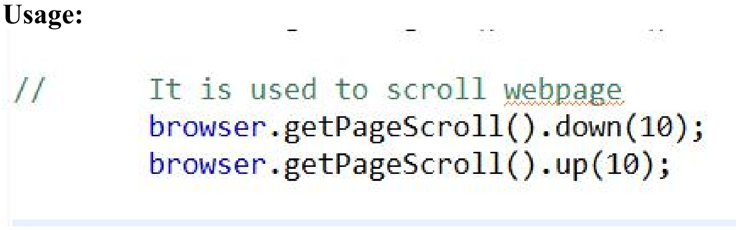
Purpose:

This method is to get the Scroll class object from Browser class.

Returns:

Returns Scroll object

Usage:



```
//      It is used to scroll webpage
        browser.getPageScroll().down(10);
        browser.getPageScroll().up(10);
```

Method Name:

public WebTable getTable()

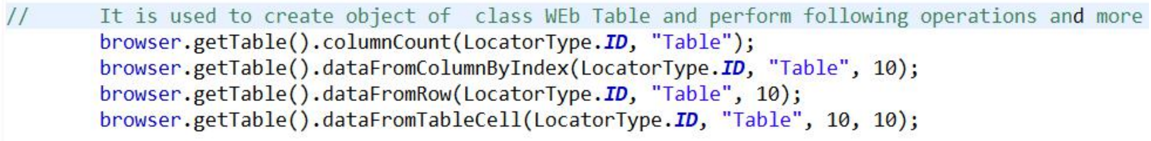
Purpose:

This method is to get the table class object from Browser class and set the find object.

Returns:

Returns webTable object

Usage:



```
//      It is used to create object of  class WEb Table and perform following operations and more
        browser.getTable().columnCount(LocatorType.ID, "Table");
        browser.getTable().dataFromColumnByIndex(LocatorType.ID, "Table", 10);
        browser.getTable().dataFromRow(LocatorType.ID, "Table", 10);
        browser.getTable().dataFromTableCell(LocatorType.ID, "Table", 10, 10);
```

Method Name:

public TextField getTextField()

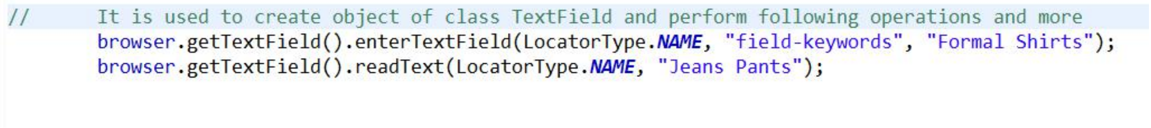
Purpose:

This method is to get the TextField class object from Browser class and set the find object.

Returns:

Returns textfield object

Usage:



```
//      It is used to create object of class TextField and perform following operations and more
        browser.getTextField().enterTextField(LocatorType.NAME, "field-keywords", "Formal Shirts");
        browser.getTextField().readText(LocatorType.NAME, "Jeans Pants");
```

Method Name:

public WaitElement getWait()

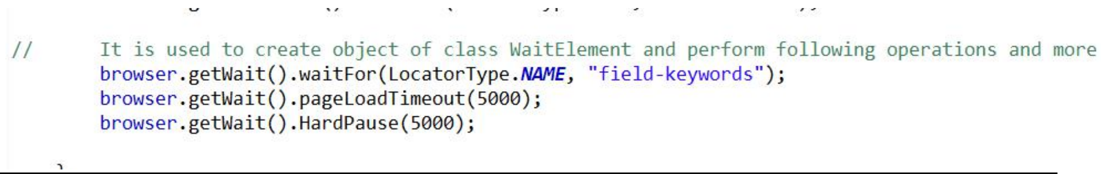
Purpose:

This method is to get the WaitElement class object from Browser class and set the find object.

Returns:

Returns waitElement object

Usage:



```
//      It is used to create object of class WaitElement and perform following operations and more
        browser.getWait().waitFor(LocatorType.NAME, "field-keywords");
        browser.getWait().pageLoadTimeout(5000);
        browser.getWait().HardPause(5000);
```

Method Name:

public WindowHandler getWindowHandler()

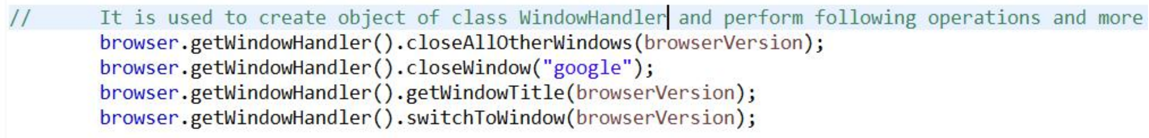
Purpose:

This method is to get the WindowHandler class object from Browser class and set the find object.

Returns:

Returns WindowHandler object

Usage:



```
//      It is used to create object of class WindowHandler and perform following operations and more
        browser.getWindowHandler().closeAllOtherWindows(browserVersion);
        browser.getWindowHandler().closeWindow("google");
        browser.getWindowHandler().getWindowTitle(browserVersion);
        browser.getWindowHandler().switchToWindow(browserVersion);
```

Method Name:

public boolean maximizeWindow()

Purpose:

This method will maximize your browser.

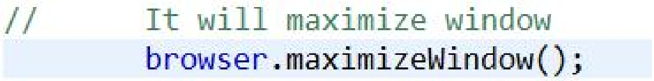
Throws:

NullPointerException

Returns:

Returns true if Successful or Returns false

Usage:



```
//      It will maximize window
        browser.maximizeWindow();
```

Method Name:

public boolean openPage(String url, BrowserType browserType)

Purpose:

This method will Open the web page with Required Parameters

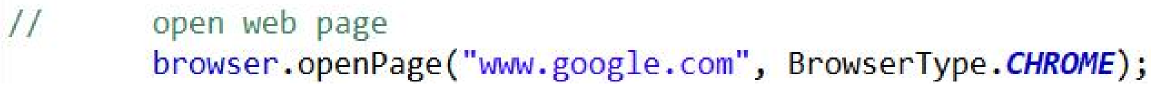
Throws:

NullPointerException

Returns:

Returns true if Successful or Returns false

Usage:



```
//      open web page
        browser.openPage("www.google.com", BrowserType.CHROME);
```

Method Name:

public boolean openPage(String url, BrowserType browserType, String profileName)

Purpose:

This method will Open the web page with Profile. It supports FIREFOX browser only.

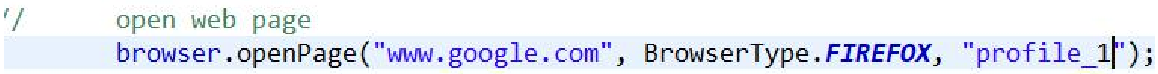
Throws:

NullPointerException

Returns:

Returns true if Successful or Returns false

Usage:



```
'/      open web page
        browser.openPage("www.google.com", BrowserType.FIREFOX, "profile_1");
```

Method Name:

public void openPageCommons(String url)

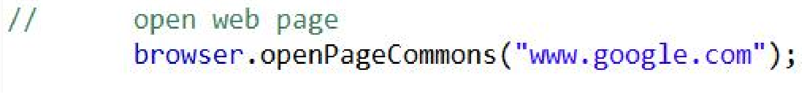
Purpose:

This method is using in both openPage with profile and openPage without profile methods To avoid the code repetition, providing a new method which get the driver from driverManager and set the driver for required classes

Returns:

Returns nothing

Usage:



```
//      open web page
        browser.openPageCommons("www.google.com");
```

Method Name:

public boolean openURL(String url, String os, String osVersion, String browser, String browserVersion)

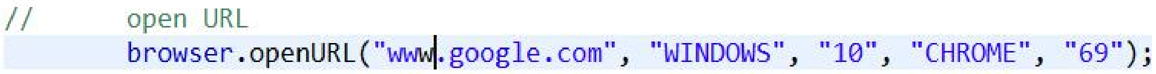
Purpose:

This method will open url with given parameter

Returns:

Returns nothing

Usage:



```
//      open URL
        browser.openURL("www.google.com", "WINDOWS", "10", "CHROME", "69");
```

Method Name:

public boolean openURL(String url, String os, String osVersion, String browser, String browserVersion,DesiredCapabilities capabilities)

Purpose:

This method will open url with given parameter

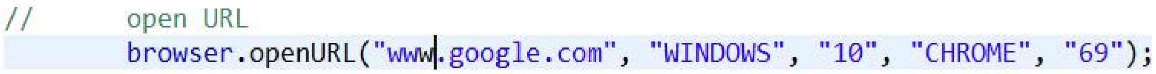
Throws:

IllegalArgumentException

Returns:

Returns nothing

Usage:



```
//      open URL
        browser.openURL("www.google.com", "WINDOWS", "10", "CHROME", "69");
```

Method Name:

public boolean quitBrowser()

Purpose:

This method will quit Browser.

Throws:

NullPointerException

Returns:

Returns true if Successful or Returns false

Method Name:

public boolean refreshBrowser()

Purpose:

This method will refresh Browser.

Throws:

NullPointerException

Returns:

Returns true if Successful or Returns false

Usage:

```
browser.refreshBrowser();
```

Method Name:

private void setupDriverForRequiredClasses()

Purpose:

This method helps to share the same driver to all required classes in framework Set the driver for find class Set the driver for waitElement class Set the driver for click class Set the driver for in ThreadPool class Set the driver for mouseMove class

Returns:

Returns nothing

Method Name:

private void setupDriverForRequiredClassesForParallel()

Purpose:

This method helps to share the same driver to all required classes in framework Set the driver for find class Set the driver for waitElement class Set the driver for click class Set the driver for in ThreadPool class Set the driver for mouseMove class

Returns:

Returns nothing

Method Name:

private boolean uploadImage(File file)

Purpose:

This method is used to upload screenshot of the web site to the report server whenever the test method is failed.

Returns:

Returns true if Successful or Returns false

Method Name:

private boolean validateURL(String url)

Purpose:

This method will helps to validate the provided URL is correct or not It accepts url as a string And returns boolean.

Throws:

IOException

Returns:

Returns true if Successful or Returns false

..

Class :

CheckBox :

Methods:

1. clickCheckBoxById(LocatorType,String,SelectType) (boolean)
2. clickCheckBoxByValue(LocatorType,String String,SelectType)(boolean)
3. clickCheckBoxByCssSelector(LocatorType,String,SelectType)(boolean)
4. isCheckBoxSelectedById(LocatorType,String)(boolean)
5. selectCheckBoxByValue(LocatorType,String,String)(boolean)
6. isCheckBoxSelectByValue(LocatorType,String,String)(boolean)
7. isCheckBoxSelectByCssSelector(LocatorType,String)(boolean)
8. setFinder(IFind)(void)

Description:

A. **Method Name**:

public boolean clickCheckBoxById(LocatorType locator,String elementId,SelectType selectType)

Purpose :

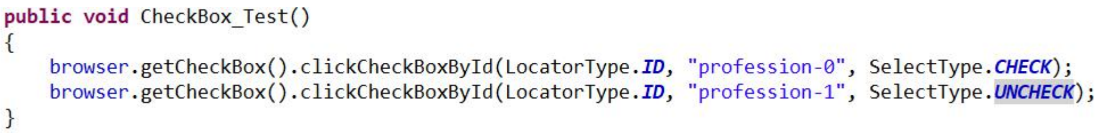
1. This Method will identify element by there ID and perform click operation on element.

Parameters: This methods having three parameters:

- Locator- It identifies the WebElement.
- ElementId- It is Value of locator type.
- SelectType- It is selectorType to select the check box(CHECK/UNCHECK)

Returns: This method returns True if element is Clicked or False if not clicked.

Uses :



```
public void CheckBox_Test()
{
    browser.getCheckBox().clickCheckBoxById(LocatorType.ID, "profession-0", SelectType.CHECK);
    browser.getCheckBox().clickCheckBoxById(LocatorType.ID, "profession-1", SelectType.UNCHECK);
}
```

B. **Method Name**:

public boolean clickCheckBoxByValue(LocatorType locator, String elementName, String elementValue, SelectType selectType)

Purpose :

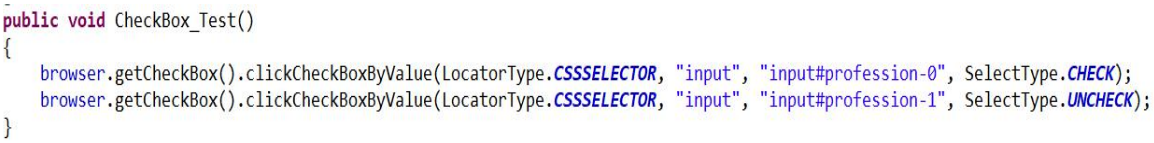
1. This method performs click operation on element by there value.

Parameters: This method having four parameters:

- Locator - It is identifier of WebElement.
- elementName - It is name of locator type.
- elementValue - It is value of locator type.
- SelectType - It is select type for selection of check box(CHECK/UNCHECK).

Returns: This method returns true if checked or false if not.

Uses :



```
public void CheckBox_Test()
{
    browser.getCheckBox().clickCheckBoxByValue(LocatorType.CSSSELECTOR, "input", "input#profession-0", SelectType.CHECK);
    browser.getCheckBox().clickCheckBoxByValue(LocatorType.CSSSELECTOR, "input", "input#profession-1", SelectType.UNCHECK);
}
```

C. **Method Name**:

Public boolean clickCheckBoxByCssSelector(LocatorType locator,String elementCssSelector,SelectType selectType)

Purpose :

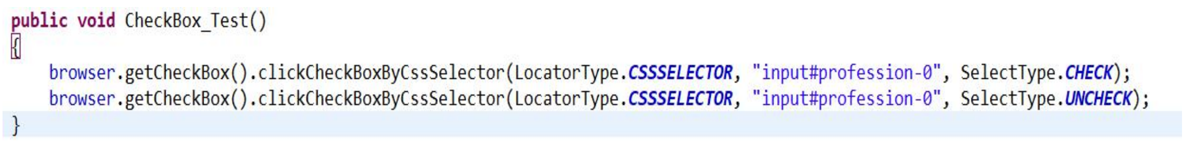
1. This method performs click operation on check box by using CSS Selector.

Parameters: This method having three parameters.

- Locator - It is identifier of WebElement
- CssSelector - It is value for locator.
- SelectType - It is select type for selection of check box(CHECK/UNCHECK).

Returns: This method returns true for if clicked and false for if not.

Uses:



```
public void CheckBox_Test()
{
    browser.getCheckBox().clickCheckBoxByCssSelector(LocatorType.CSSSELECTOR, "input#profession-0", SelectType.CHECK);
    browser.getCheckBox().clickCheckBoxByCssSelector(LocatorType.CSSSELECTOR, "input#profession-0", SelectType.UNCHECK);
}
```

D. **Method Name**:

public boolean isCheckBoxSelectedById(LocatorType locator,String elementId)

Purpose:

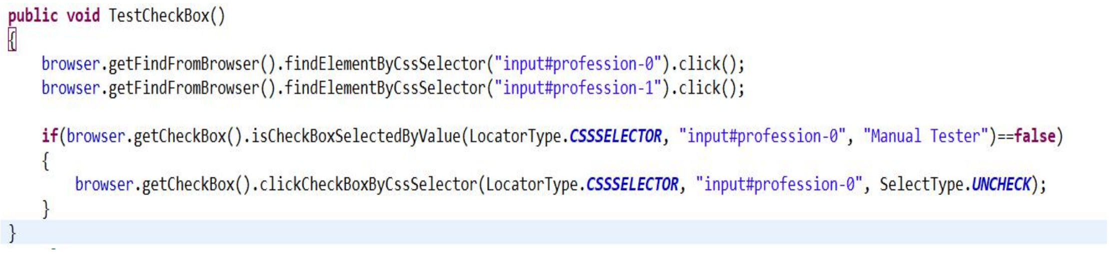
1. This method checks whether check box is clicked or not by ID.
2. If check box is clicked it returns true otherwise returns false.

Parameters: This method having two parameters:

- Locator: It is identifier of WebElement.
- elementId: It is value for locator.

Return: This method returns true if checked otherwise returns false.

Uses:



```
public void TestCheckBox()
{
    browser.getFindFromBrowser().findElementByCssSelector("input#profession-0").click();
    browser.getFindFromBrowser().findElementByCssSelector("input#profession-1").click();

    if(browser.getCheckBox().isCheckBoxSelectedByValue(LocatorType.CSSSELECTOR, "input#profession-0", "Manual Tester")==false)
    {
        browser.getCheckBox().clickCheckBoxByCssSelector(LocatorType.CSSSELECTOR, "input#profession-0", SelectType.UNCHECK);
    }
}
```

E. **Method Name**:

public boolean selectCheckBoxByValue(LocatorType locator,String elementName,String Value)

Purpose:

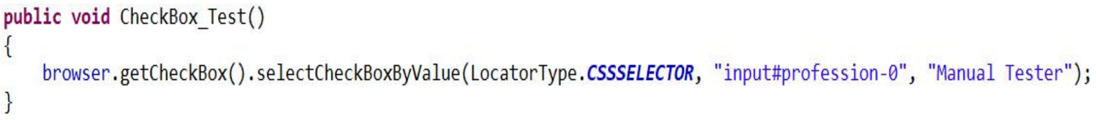
1. This method performs selection on check box by there value.

Parameters: This method having three parameters

- Locator- It is identifier of WebElement.
- elementName- It is value for locator.
- Value- Value of that element.

Returns: This method returns true if check box is selected and false if not.

Uses:



```
public void CheckBox_Test()
{
    browser.getCheckBox().selectCheckBoxByValue(LocatorType.CSSSELECTOR, "input#profession-0", "Manual Tester");
}
```

F. **Method Name**:

public boolean isCheckBoxSelectByValue(LocatorType locator,String elementName,String value)

Purpose:

1. This method checks if check box is selected or not by there value.

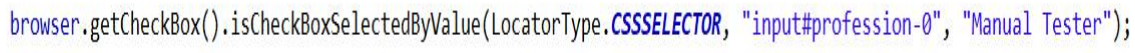
Parameters: This method have three parameters

- Locator- It is identifier of WebElement
- elementName- It is value for locator
- value- It is value of that element.

Returns: This method returns true if checkbox is selected otherwise false.

Exception: noSuchElementException

Uses:



```
browser.getCheckBox().isCheckBoxSelectedByValue(LocatorType.CSSSELECTOR, "input#profession-0", "Manual Tester");
```

G. **Method Name**:

public boolean isCheckBoxSelectByCssSelector(LocatorType locator,String elementCssSelector)

Purpose:
1. This method checks if check box is selected or not by the CssSelector.

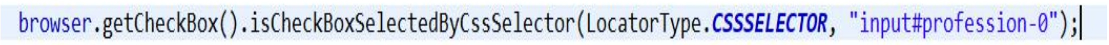
Parameters: This method having two parameters

- Locator - It is identifier of WebElement which is CssSelector.
- elementCssSelector - It is value for locator.

Returns: It returns true if check box is selected and false if not.

Exception: noSuchElementException

Uses:



```
browser.getCheckBox().isCheckBoxSelectedByCssSelector(LocatorType.CSSSELECTOR, "input#profession-0");
```

H. Method Name:

public void setFinder(IFind finder)

Purpose:
1. This method accepts the reference variable of IFind interface.
2. This method is use to refer instance variable.

Returns:
It will returns Nothing.

..

Class:

Click:

Methods:

1. clickButtonByName(LocatorType, String) (boolean)
2. clickImageByXpath(LocatorType, String) (boolean)
3. doubleClickOnElementByXpath(LocatorType, String) (boolean)
4. doubleClickOnImageById(LocatorType, String) (boolean)
5. performClick(LocatorType, String) (boolean)
6. PerformDoubleClick(LocatorType, String) (boolean)
7. performRightClick(LocatorType, String) (boolean)
8. rightClickOnElementByXpath(LocatorType, String) (boolean)
9. setDriverForActionClick(WebDriver) (void)
10. setFinder(IFind)

A. Method Name:

public boolean clickButtonByName(LocatorType locator, String button)

Purpose:

1. This method perform click on button by there name
2. We have to provide button name to this method.

Parameters: This method having Two parameters:

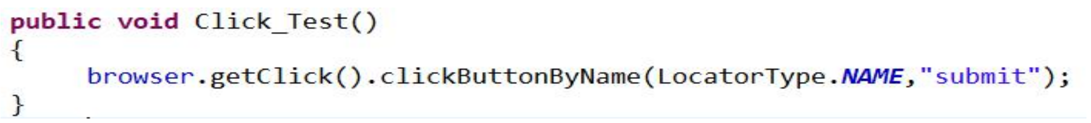
- Locator - It is identifier of element.
- Botton - It contains value of locator which means button name.

Returns:

It returns true if button gets click otherwise return false.

Exceptions: NoSuchElementException

Uses:



```
public void Click_Test()
{
    browser.getClick().clickButtonByName(LocatorType.NAME,"submit");
}
```

B. Method Name:

public boolean clickImageByXpath(LocatorType locator, String image)

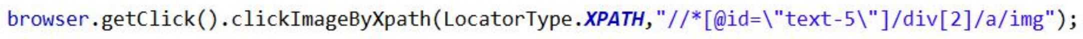
Purpose:
1. This method select image by Xpath and perform click operation.

Parameters: This method having two parameters
- ➢ Locator- It is identifier of element.
- ➢ Image- It is the value for locator type.

Return:
This method returns true if clicked the image otherwise false.

Exceptions: NoSuchElementException

Uses:



```
browser.getClick().clickImageByXpath(LocatorType.XPATH,"//*[@id=\"text-5\"]/div[2]/a/img");
```

C. Method Name:

public boolean doubleClickOnElementByXpath(LocatorType locator, String value)

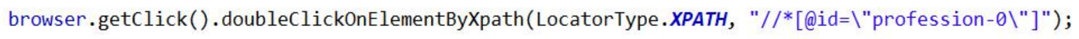
Purpose:
1. This method perform double click on web element by using there Xpath.

Parameters: This Method having two parameters:
- ➢ Locator - It is identifier of web element.
- ➢ Value - It is value of locator type.

Return:
It method returns true if double click successfully done otherwise returns false.

Exceptions: NoSuchElementException

Uses:



```
browser.getClick().doubleClickOnElementByXpath(LocatorType.XPATH, "//*[@id=\"profession-0\"]");
```

D. Method Name:

public boolean doubleClickOnImageById(LocatorType locator, String image)

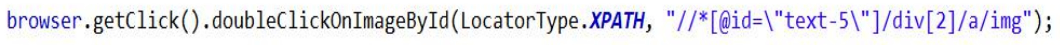
Purpose:
1. This method performs double click operation on given image by there Id.

Parameters: This method having Two parameters:
- ➢ Locator- It is identifier of element.
- ➢ Image- It is value for locator type.

Return:
It returns true if image is clicked otherwise returns false.

Exceptions: NoSuchElementException

Uses:



```
browser.getClick().doubleClickOnImageById(LocatorType.XPATH, "//*[@id=\"text-5\"]/div[2]/a/img");
```

E. Method Name:

public boolean performClick(LocatorType locator, String element)

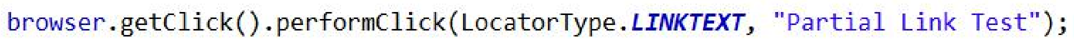
Purpose:
1. This method performs click operation on given element.
2. We can click element by using any identifier of element.

Parameters: This method having two parameters:
- Locator- It is identifier of web element.
- Element- It is value for locator type.

Return:
This method returns true if element gets click otherwise return false.

Exceptions:
- NullPointerException
- illegalArgumentException

Uses:



```
browser.getClick().performClick(LocatorType.LINKTEXT, "Partial Link Test");
```

F. Method Name:

public boolean PerformDoubleClick(LocatorType loactor, String element)

Purpose:
1. This method use to perform double click operation on element.
2. We can perform double click by using any identifier of element.

Parameters: This method having two parameters:
- Locator- It is identifier of element.
- Element- It is string value for locator type.

Exceptions:
- NullPointerException
- illegalArgumentException

Return:
This method returns true if operation successfully done otherwise returns false.

G. Method Name:

public boolean performRightClick(LocatorType locator, String element)

Purpose:
1. This method use to perform right click on any element.
2. We can use any identifier for performing this operation.

Parameters: This method have two parameters:

- Locator- It is identifier for web element.
- Element- It is string value for locator type.

Exceptions:

- NullPointerException
- illegalArgumentException

Return:
It returns true on successfully execution otherwise returns false.

H. Method Name:

public boolean rightClickOnElementByXpath(LocatorType, String)

Purpose:
1. This method performs right click operation on element.
2. We can perform right click operation by using Xpath of that web element.

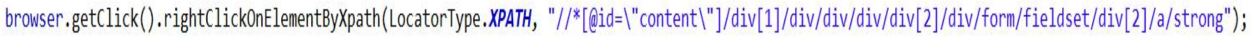
Parameters: This method having two parameters :

- Locator- It is identifier of web element.
- Element- It is value for locator type.

Return:
This method returns true on successfully execution otherwise returns false.

Exceptions: NoSuchElementException

Uses:



```
browser.getClick().rightClickOnElementByXpath(LocatorType.XPATH, "//*[@id=\"content\"]/div[1]/div/div/div/div[2]/div/form/fieldset/div[2]/a/strong");
```

I. Method Name:

public void setDriverForActionClick(WebDriver driver)

Purpose:
1. This method accepts reference variable of WebDriver Interface, which will initialize to FirefoxDriver or InternetExplorerDriver or the ChromeDriver class constructor.
2. This method is use to refer current instance variable.

Returns:
It will returns nothing.

J. Method Name:

Public void setFinder(IFind finder)

Purpose:

3. This method accepts the reference variable of IFind interface.
4. This method is use to refer instance variable.

Returns:
It will returns Nothing.

..

Class: CustomListener

Methods:

1. public void afterInvocation(IInvokedMethod method, ITestResult result)

2. public void beforeInvocation(IInvokedMethod method, ITestResult result)

3. private void captureScreenShot(ITestResult result)

4. private void createJIRABug(ITestResult result)

5. public void onExecutionFinish()

6. public void onExecutionStart()

7. public void onStart(ITestContext context)

8. public void onTestFailure(ITestResult result)

9. private void uploadImage(File file)

Method Name:

public void afterInvocation(IInvokedMethod method,
ITestResult result)

Purpose:

This method invoke after configuration method(@before and @after).It will check invoked method is configuration method or not if not then it will create a failure message with all failures and stack trace if it is configuration method then it will check assertion failure and verification failure.

Returns:

Returns nothing

Usage:

Used to check status of Configuration method.

Method Name:

public void beforeInvocation(IInvokedMethod method,
ITestResult result)

Purpose:

This method invoke before configuration method (@before and @after). It will check invoked method is configuration method or not if not then it will create a failure message with all failures and stack trace if it is configuration method then it will check assertion failure and verification failure.

Returns:

Returns nothing

Usage:

Used to check status of Configuration method.

Method Name:

private void captureScreenShot(ITestResult result)

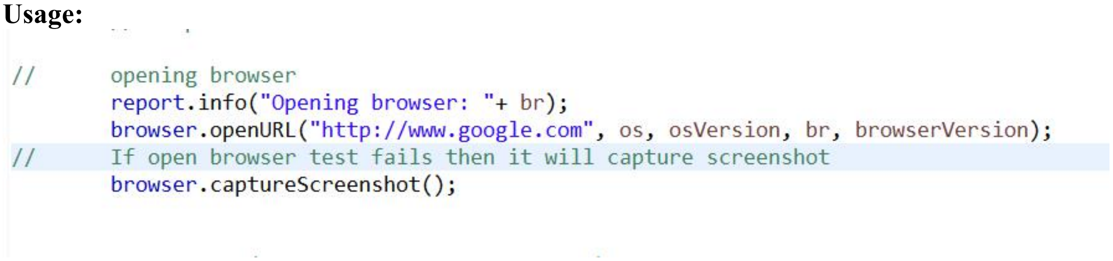
Purpose:

This method is used to take the screenshot whenever the test script is fail.

Returns:

Returns nothing

Usage:



```
//      opening browser
        report.info("Opening browser: "+ br);
        browser.openURL("http://www.google.com", os, osVersion, br, browserVersion);
//      If open browser test fails then it will capture screenshot
        browser.captureScreenshot();
```

Method Name:

private void createJIRABug(ITestResult result)

Purpose:

This method will create JIRA bug after failure.If you want to create JIRA Bug. Please set value of key 'automatic.bug.creation' to 'true' in config.properties file otherwise it will not create JIRA bug.

Returns:

Returns nothing

Usage:

Used to JIRA bug.

Method Name:

public void onExecutionFinish()

Purpose:

After finishing execution of Script if you want to execute some code then this listener is used.

Returns:

Returns nothing

Method Name:

public void onExecutionStart()

Purpose:

After starting execution of Script if you want to execute some code then this listener is used.

Returns:

Returns nothing

Method Name:

public void onStart(ITestContext context)

Purpose:

It is used to get Suit name

Returns:

Returns nothing

Method Name:

public void onTestFailure(ITestResult result)

Purpose:

This method will captures Screen shot on Failure and also create JIRA bug.

Returns:

Returns nothing

Method Name:

private void uploadImage(File file)

Purpose:

This method will upload captures Screen shot on Failure and also create JIRA bug.

Returns:

Returns nothing

..

Class: DriverManager

Methods:

1.public WebDriver getDriver()

2.protected void linCaps()

3.protected void macCaps()

4. protected void winCaps()

5.private WebDriver setFirefoxDriver()

6.private WebDriver setChromeDriver()

7.public void setDriver(BrowserType browserType)

8.public void setDriver(BrowserType browserType,
String profileName)

9.protected WebDriver
setHeadlessDriverCapabilities()

Method Name:

public WebDriver getDriver()

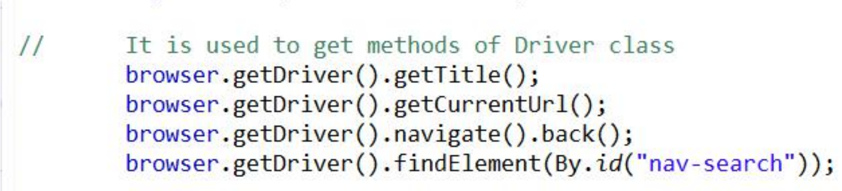
Purpose:

This method is to get the driver object from Browser class.

Returns:

Returns current driver object

Usage:



```
//      It is used to get methods of Driver class
        browser.getDriver().getTitle();
        browser.getDriver().getCurrentUrl();
        browser.getDriver().navigate().back();
        browser.getDriver().findElement(By.id("nav-search"));
```

Method Name:

protected void linCaps()

Purpose:

This method checks Linux capabilities i.e OS Arch, OsVersion, etc.

Returns:

Returns nothing

Usage:

Used to check all required capabilities for Linux OS.

Method Name:

protected void macCaps()

Purpose:

This method checks mac capabilities i.e OS Arch, OsVersion, etc.

Returns:

Returns nothing

Usage:

Used to check all required capabilities for Mac OS.

Method Name:

protected void winCaps()

Purpose:

This method checks Windows capabilities i.e OS Arch, OsVersion, etc.

Returns:

Returns nothing

Usage:

Used to check all required capabilities for Windows OS.

Method Name:

private WebDriver setFirefoxDriver()

Purpose:

This method will set the path for driver.Path will depend on currently using OS.

Returns:

Returns object of FireFoxDriver

Usage:

Used to set Firefox Driver.

Method Name:

private WebDriver setChromeDriver()

Purpose:

This method will set the path for driver.Path will depend on currently using OS.

Returns:

Returns object of setChromeDriver

Usage:

Used to set Chrome Driver.

Method Name:

public void setDriver(BrowserType browserType)

Purpose:

This method will set the path for driver.Path will depend on currently using OS.

Returns:

Returns nothing

Usage:

Used to set Driver.

Method Name:

public void setDriver(BrowserType browserType, String profileName)

Purpose:

This method will set the path for driver.Path will depend on currently using OS and Name of driver AND Profile name is given through browserType and profileName.

Only works for Firefox Browser.

Returns:

Returns nothing

Throws:

NullPointerException

IllegalArgumentException

Usage:

Used to set Driver.

Method Name:

protected WebDriver setHeadlessDriverCapabilities()

Purpose:

A headless browser is a web-browser without a graphical user interface. This method is used to set capabilities for browser but will not show any GUI.

Returns:

Returns current driver object

Throws:

NullPointerException

IllegalArgumentException

Usage:

Used to set Headless Driver.

..

Class: Find

Methods:

1. public WebElement findElementByClassName(String className)
2. public WebElement findElementByCssSelector(String cssSelector)
3. public WebElement findElementById(String id)
4. public WebElement findElementByLinkText(String linkText)
5. public WebElement findElementByPartialLinkText(String linkText)
6. public WebElement findElementByName(String name)
7. public WebElement findElementByTagName(String name)
8. public WebElement findElementByXpath(String name)
9. public List<WebElement> findElementsBy(By by)
10. public List<WebElement> ~~findElementsByClassName~~(String className)
11. public List<WebElement> ~~findElementsByCssSelector~~(String cssSelector)
12. public List<WebElement> ~~findElementsById~~(String id)
13. public List<WebElement> ~~findElementsByLinkText~~(String linkText)
14. public List<WebElement> ~~findElementsByName~~(String name)
15. public List<WebElement> ~~findElementsByPartialLinkText~~(String partialLinkText)
16. public List<WebElement> ~~findElementsByTagName~~(String tagName)
17. public List<WebElement> ~~findElementsByXpath~~(String xpath)
18. public void setDriverForFind(WebDriver driver)

Method Name:

public WebElement findElementByClassName(String className)

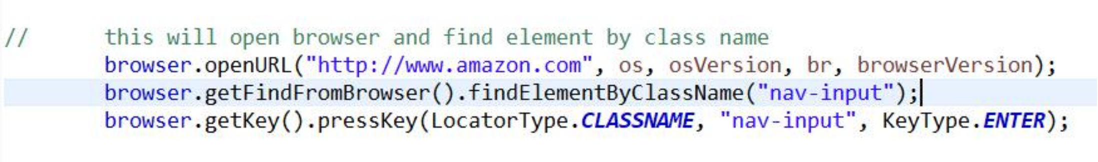
Purpose:

This method will find webElement by class name with help of webElement object.

Returns:

Returns webElement object

Usage:



```
//      this will open browser and find element by class name
        browser.openURL("http://www.amazon.com", os, osVersion, br, browserVersion);
        browser.getFindFromBrowser().findElementByClassName("nav-input");
        browser.getKey().pressKey(LocatorType.CLASSNAME, "nav-input", KeyType.ENTER);
```

Method Name:

public WebElement findElementByCssSelector(String cssSelector)

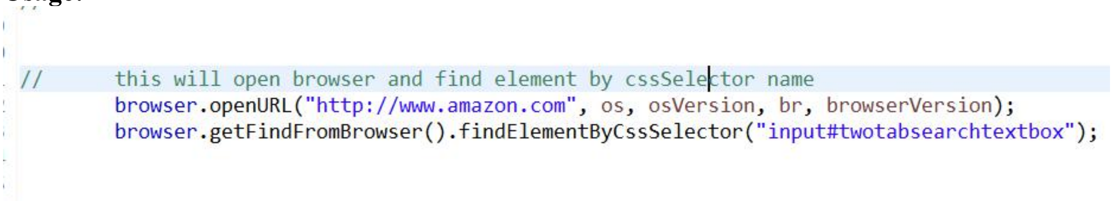
Purpose:

This method will find webElement by cssSelector with help of webElement object.

Returns:

Returns webElement object

Usage:



```
//      this will open browser and find element by cssSelector name
        browser.openURL("http://www.amazon.com", os, osVersion, br, browserVersion);
        browser.getFindFromBrowser().findElementByCssSelector("input#twotabsearchtextbox");
```

Method Name:

public WebElement findElementById(String id)

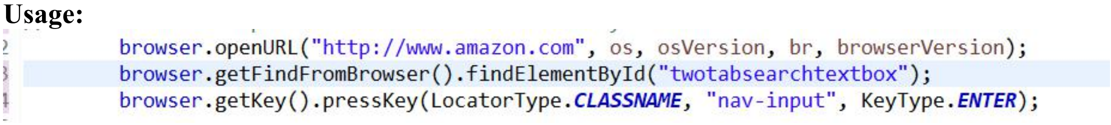
Purpose:

This method will find webElement by ID with help of webElement object.

Returns:

Returns webElement object

Usage:



```
        browser.openURL("http://www.amazon.com", os, osVersion, br, browserVersion);
        browser.getFindFromBrowser().findElementById("twotabsearchtextbox");
        browser.getKey().pressKey(LocatorType.CLASSNAME, "nav-input", KeyType.ENTER);
```

Method Name:

public WebElement findElementByLinkText(String linkText)

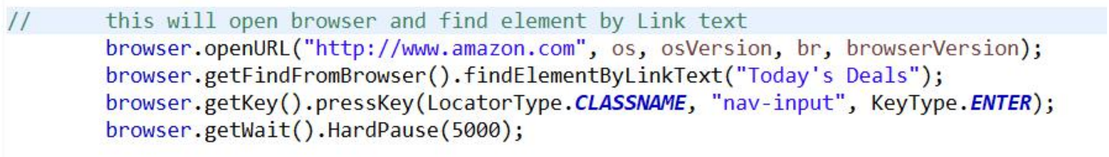
Purpose:

This method will find webElement by LinkText with help of webElement object.

Returns:

Returns webElement object

Usage:



```
//      this will open browser and find element by Link text
        browser.openURL("http://www.amazon.com", os, osVersion, br, browserVersion);
        browser.getFindFromBrowser().findElementByLinkText("Today's Deals");
        browser.getKey().pressKey(LocatorType.CLASSNAME, "nav-input", KeyType.ENTER);
        browser.getWait().HardPause(5000);
```

Method Name:

public WebElement findElementByPartialLinkText(String linkText)

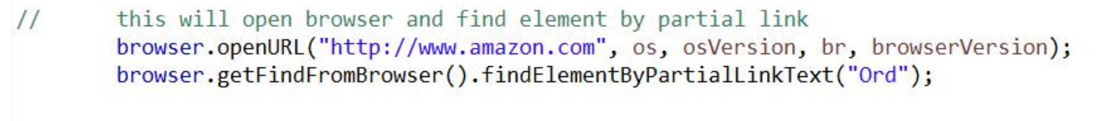
Purpose:

This method will find webElement by PartialLinkText with help of webElement object.

Returns:

Returns webElement object

Usage:



```
//      this will open browser and find element by partial link
        browser.openURL("http://www.amazon.com", os, osVersion, br, browserVersion);
        browser.getFindFromBrowser().findElementByPartialLinkText("Ord");
```

Method Name:

public WebElement findElementByName(String name)

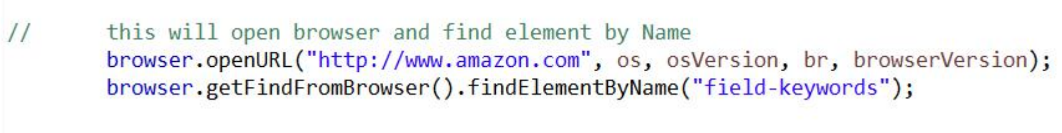
Purpose:

This method will find webElement by Name with help of webElement object.

Returns:

Returns webElement object

Usage:



```
//      this will open browser and find element by Name
        browser.openURL("http://www.amazon.com", os, osVersion, br, browserVersion);
        browser.getFindFromBrowser().findElementByName("field-keywords");
```

Method Name:

public WebElement findElementByTagName(String name)

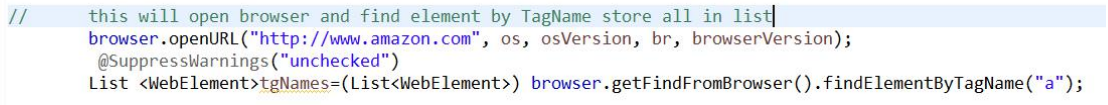
Purpose:

This method will find webElement by TagName with help of webElement object.

Returns:

Returns webElement object

Usage:



```
//      this will open browser and find element by TagName store all in list
        browser.openURL("http://www.amazon.com", os, osVersion, br, browserVersion);
         @SuppressWarnings("unchecked")
        List <WebElement>tgNames=(List<WebElement>) browser.getFindFromBrowser().findElementByTagName("a");
```

Method Name:

public WebElement findElementByXpath(String name)

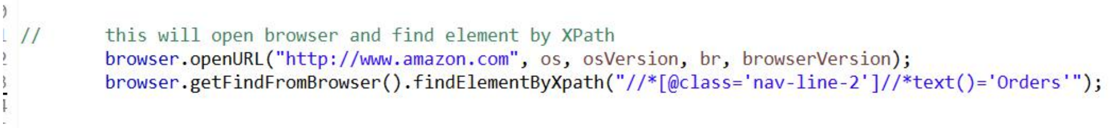
Purpose:

This method will find webElement by Xpath with help of webElement object.

Returns:

Returns webElement object

Usage:



```
//      this will open browser and find element by XPath
        browser.openURL("http://www.amazon.com", os, osVersion, br, browserVersion);
        browser.getFindFromBrowser().findElementByXpath("//*[@class='nav-line-2']//*text()='Orders'");
```

Method Name:

public List<WebElement> findElementsBy(By by)

Purpose:

This method gives list of WebElements by given 'By' instance

Returns:

Returns List<WebElement>

Method Name:

public List<WebElement> ~~findElementsByClassName~~(String className)

Purpose:

It is a Deprecated method.

This method gives list of WebElements by given ClassName instance

Returns:

Returns List<WebElement>

Method Name:

public List<WebElement> ~~findElementsByCssSelector~~(String cssSelector)

Purpose:

It is a Deprecated method.

This method gives list of WebElements by given CssSelector instance

Returns:

Returns List<WebElement>

Method Name:

public List<WebElement> ~~findElementsById~~(String id)

Purpose:

It is a Deprecated method.
This method gives list of WebElements by given ID instance

Returns:

Returns List<WebElement>

Method Name:

public List<WebElement> ~~findElementsByLinkText~~(String linkText)

Purpose:

It is a Deprecated method.
This method gives list of WebElements by given LinkText instance

Returns:

Returns List<WebElement>

Method Name:

public List<WebElement>
~~findElementsByName~~(String name)

Purpose:

It is a Deprecated method.
This method gives list of WebElements by given Name instance

Returns:

Returns List<WebElement>

Method Name:

public List<WebElement> ~~findElementsByPartialLinkText~~(String partialLinkText)

Purpose:

It is a Deprecated method.
This method gives list of WebElements by given PartialLinkText instance

Returns:

Returns List<WebElement>

Method Name:

public List<WebElement> ~~findElementsByTagName~~(String tagName)

Purpose:

It is a Deprecated method.
This method gives list of WebElements by given TagName instance

Returns:

Returns List<WebElement>

Method Name:

public List<WebElement> ~~findElementsByXpath~~(String xpath)

Purpose:

It is a Deprecated method.
This method gives list of WebElements by given TagName instance

Returns:

Returns List<WebElement>

Method Name:

public void setDriverForFind(WebDriver driver)

Purpose:

This method helps to set the driver to find class

Throws:

NullPointerException

Returns:

Returns nothing

..

Class: Key

Methods:

1.public boolean pressKey(LocatorType locator, String element, KeyType keyType)

2.public void setFinder(IFind findObj)

Method Name:

public boolean pressKey(LocatorType locator, String element, KeyType keyType)

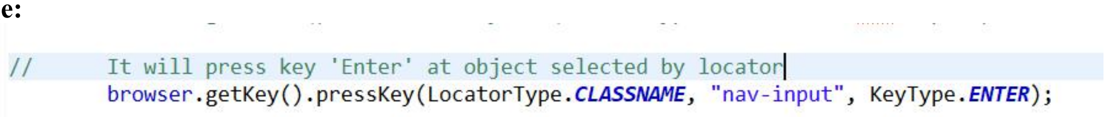
Purpose:

This method will find key by locatorType and perform pressKey operation with KeyType.

Returns:

It returns true if key is pressed or false if it is not pressed.

Usage:



```
//      It will press key 'Enter' at object selected by locator
        browser.getKey().pressKey(LocatorType.CLASSNAME, "nav-input", KeyType.ENTER);
```

Method Name:

public void setFinder(IFind findObj)

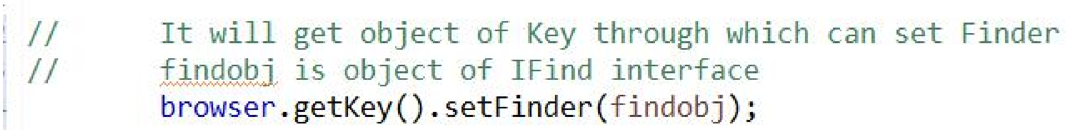
Purpose:

This method is used to set the finder object which is set from Browser class. findobj is object of IFind interface

Returns:

It returns true if key is pressed or false if it is not pressed.

Usages:



```
//      It will get object of Key through which can set Finder
//      findobj is object of IFind interface
        browser.getKey().setFinder(findobj);
```

..
Class:

MouseMove :

Methods :

1. clickAndHold(WebElement)(boolean)
2. dragAndDrop(WebElement, WebElement)(boolean)
3. dragAndDropBy(WebElement, int ,int)(boolean)
4. maouseHover(WebElement)(boolean)
5. releaseElement(WebElement)(boolean)
6. setDriverForMouseMove(WebDriver)(void)

Description :

A. **Method Name**:

Public boolean clickAndHold(WebElement element)

Purpose:
1. This method use to click the web element without releasing it or hold it.

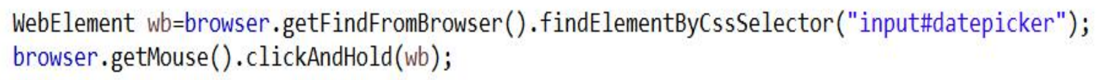
Parameters: This method have one parameter

- Element- It is WebElement type which having identification of web element.

Returns:
It returns true when element is selected and holds and false if not.

Exception : NullPointerException

Uses:



```
WebElement wb=browser.getFindFromBrowser().findElementByCssSelector("input#datepicker");
browser.getMouse().clickAndHold(wb);
```

B. **Method Name:**

public boolean dragAndDrop(WebElement from, WebElement to)

Purpose :
1. This method is perform click and hold source element then drag element and move it into another element.

Parameters: This method have two parameters:

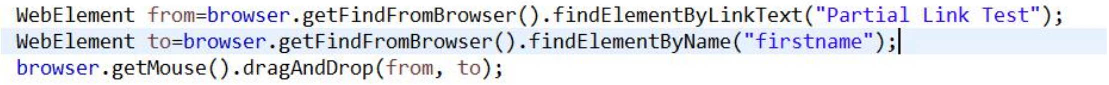
- From- It includes which element we want to drag.
- To- It includes where to drop the element.

Returns :

It returns true if drag and drop operation is successful otherwise returns false.

Exception : NullPointerException

Uses:



```
WebElement from=browser.getFindFromBrowser().findElementByLinkText("Partial Link Test");
WebElement to=browser.getFindFromBrowser().findElementByName("firstname");
browser.getMouse().dragAndDrop(from, to);
```

C. Method Name:

public boolean dragAndDropBy(WebElement element, int x ,int y)

Purpose:
1. This method performs click and hold the source element then move it to destination element by X and Y offset.
2. You will have to pass in the element that you want to move and then the x and y offset that you want to move it by.

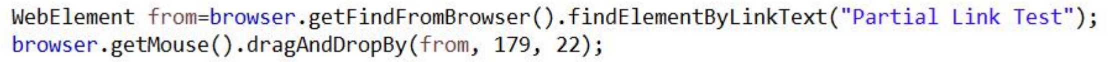
Parameters: This method having three Parameters:

- ➢ WebElement- It holds identification of web element.
- ➢ Offset X and Offset Y- X and Y are padding value of web element where we have to move the element.

Returns:
It returns true if drag and drop successful otherwise returns false.

Exception : NullPointerException

Uses:



```
WebElement from=browser.getFindFromBrowser().findElementByLinkText("Partial Link Test");
browser.getMouse().dragAndDropBy(from, 179, 22);
```

D. Method Name:

public boolean maouseHover(WebElement element)

Purpose:
1. This method is use to move the mouse to any element on web page.
2. It moves the mouse to the middle of the element.

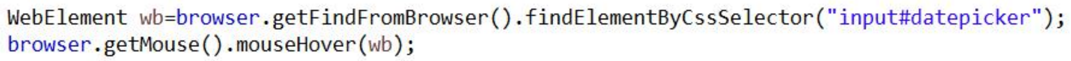
Parameters: This method having one parameter:

- ➢ WebElement- It is the identification of element on which we have to move the mouse

Returns:
It returns true if mouse move to element otherwise returns false.

Exception : NullPointerException

Uses:



```
WebElement wb=browser.getFindFromBrowser().findElementByCssSelector("input#datepicker");
browser.getMouse().mouseHover(wb);
```

E. Method Name:

public boolean releaseElement(WebElement)

Purpose:
1. This method release the mouse at the given elements location.

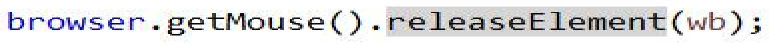
Parameters: This method having one parameter:

- WebElement- It contains identification of that element where we have to release the mouse button.

Returns:
It returns true if operation is successful otherwise returns false.

Exception : NullPointerException

Uses:



```
browser.getMouse().releaseElement(wb);
```

F. **Method Name:**

setDriverForMouseMove(WebDriver driver)

Purpose:
3. This method accepts reference variable of WebDriver Interface, which will initialize to FirefoxDriver or InternetExplorerDriver or the ChromeDriver class constructor.
4. This method is use to refer current instance variable.

Returns:
It will returns nothing.

..

Class:

Navigate :

Methods:

1. back()(boolean)

2. forword()(boolean)

3. refresh()(boolean)

4. setDriverForNavigate(WebDriver)

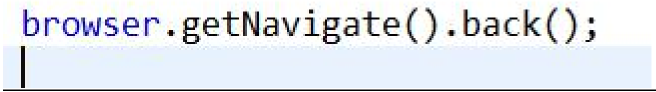
A. Method Name:

public boolean back()

Purpose:
1. This method is use to navigate the page to back.
2. It will navigate page in same window which is open, It will not open another browser or another window to show the back page.

Returns:
It returns true if back page will shown successfully otherwise it returns false.

Uses:



```
browser.getNavigate().back();
```

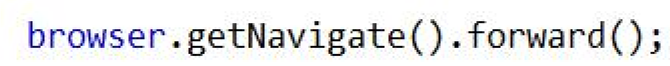
B. Method Name:

public boolean forword()

Purpose:
1. This method is used to navigate to the next page.
2. It will navigate to the next page in same window which is open, It will not open another browser or another window to show the next page.

Return:
It returns true if page successfully navigate to next page otherwise returns false.

Uses:



```
browser.getNavigate().forward();
```

C. Method Name:

public boolean refresh()

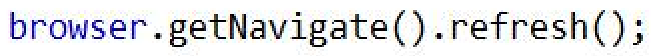
Purpose:

1. It will refresh the current page.

Return:

It will return true if page gets refresh otherwise returns false.

Uses:



```
browser.getNavigate().refresh();
```

D. Method Name:

setDriverForNavigate(WebDriver)

Purpose:

1. This method accepts reference variable of WebDriver Interface, which will initialize to FirefoxDriver or InternetExplorerDriver or the ChromeDriver class constructor.
2. This method is use to refer current instance variable.

Return:

It returns nothing.

..

Class: PageHelper

Methods:

1. public List<String> getElementsAttributeList(By by, String attributeValue)

2. public List<String> getElementsTextList(By by)

3. public void setDriverForPageHelper(WebDriver driver)

4.public void setFinder(IFind iFind)

Method Name:

public List<String> getElementsAttributeList(By by, String attributeValue)

Purpose:

This method will give list of WebElements Attribute by given 'By' instance

Returns:

Returns list of webElement

Method Name:

public List<String> getElementsTextList(By by)

Purpose:

This method will give list of WebElements Text by given 'By' instance.

Returns:

Returns list of webElement

Method Name:

public void setDriverForPageHelper(WebDriver driver)

Purpose:

This method is used to set WebDriver object for PageHelper

Returns:

Returns nothing

Method Name:

public void setFinder(IFind iFind)

Purpose:

This method is used to set Finder with help of object of IFind Interface.

Returns:

Returns nothing

..

Class: RunModeBrowserStack

Methods:

1. public RunModeBrowserStack(String enableLocal, String os,String osVersion,String browser, String browserVersion)

2. private String generateHubURL()

3. private String[] getCredentials()

4. public WebDriver getDriver()

5.public void setCapabilities(String enableLocal, String os,String osVersion, String browser, String browserVersion)

Method Name:

public RunModeBrowserStack(String enableLocal, String os, String osVersion,String browser, String browserVersion)

Purpose:

This method will run with capabilities like OS, OsVersion, browser, browser Version and run URL with help of RemoteWebDriver object if URL is correct it will run well and if not it will give exception.

Throws:

IllegalArgumentException

Method Name:

private String generateHubURL()

Purpose:

This method will take browser stack credentials and with help of credentials it will generate HubUrl.

Returns:

Returns HubUrl

Method Name:

private String[] getCredentials()

Purpose:

This method is used to get credentials and it will split credentials.

Throws:

NullPointerException

Returns:

Returns credentials

Method Name:

public WebDriver getDriver()

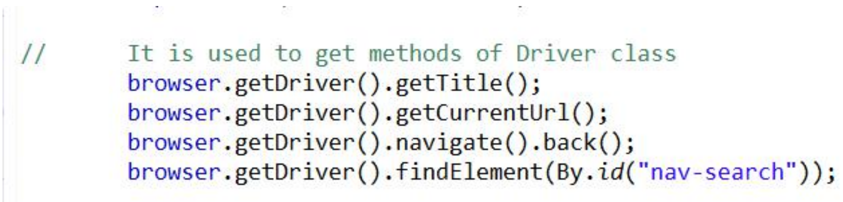
Purpose:

This method is to get the driver object from Browser class.

Returns:

Returns current driver object

Usage:



```
//      It is used to get methods of Driver class
        browser.getDriver().getTitle();
        browser.getDriver().getCurrentUrl();
        browser.getDriver().navigate().back();
        browser.getDriver().findElement(By.id("nav-search"));
```

Method Name:

public void setCapabilities(String enableLocal, String os, String osVersion, String browser, String browserVersion)

Purpose:

This method is used to set capabilities OS, OsVersion, browser, browser Version

Returns:

Returns nothing

..

Class: RunModeFactory

Methods:

1.public IDistributionManager getRunMode(String runMode, String os, String osVersion, String browser,
String browserVersion)

2.public IDistributionManager getRunMode(String runMode, String os, String osVersion, String browser, String browserVersion,DesiredCapabilities capabilities)

Method Name:

public IDistributionManager getRunMode(String runMode, String os, String osVersion, String browser,
String browserVersion)

Purpose:

This method is used get runMode as string and according to string given it will call different type of runMode methods.

Returns:

Returns object of runMode as per selected.

Method Name:

public IDistributionManager getRunMode(String runMode, String os, String osVersion, String browser, String browserVersion,DesiredCapabilities capabilities)

Purpose:

This method is used get runMode as string and according to string given it will call different type of runMode methods.Using this method we can set capabilities as per OS,OsVersion,browser,browserVersion

Returns:

Returns object of runMode as per selected with Desired Capabilities.

..

Class: RunModeGrid

Methods:

1.public IDistributionManager getRunMode(String runMode, String os, String osVersion, String browser,
String browserVersion)

2.public IDistributionManager getRunMode(String runMode, String os, String osVersion, String browser, String browserVersion,DesiredCapabilities capabilities)

3.private String generateHubURL()

4.public WebDriver getDriver()

5.public void setCapabilities(String enableLocal, String os, String osVersion, String browser, String
browserVersion)

6.private String getGridIP()

Method Name:

public IDistributionManager getRunMode(String runMode, String os, String osVersion, String browser,
String browserVersion)

Purpose:

This method is used get runMode as string and according to string given it will call different type of runMode methods.

Returns:

Returns object of runMode as per selected.

Method Name:

public IDistributionManager getRunMode(String runMode, String os, String osVersion, String browser, String browserVersion,DesiredCapabilities capabilities)

Purpose:

This method is used get runMode as string and according to string given it will call different type of runMode methods.Using this method we can set capabilities as per OS,OsVersion,browser,browserVersion

Returns:

Returns object of runMode as per selected with Desired Capabilities.

Method Name:

private String generateHubURL()

Purpose:

This method will take browser stack credentials and with help of credentials it will generate HubUrl.

Returns:

Returns HubUrl

Method Name:

public WebDriver getDriver()

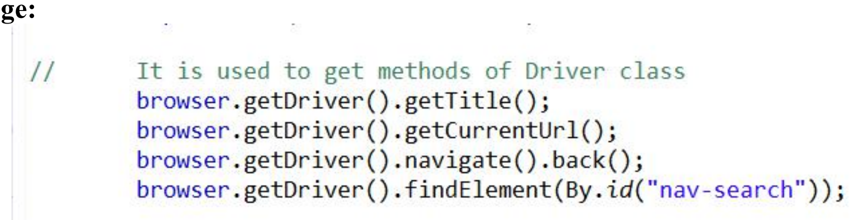
Purpose:

This method is to get the driver object from Browser class.

Returns:

Returns current driver object

Usage:



```
//      It is used to get methods of Driver class
        browser.getDriver().getTitle();
        browser.getDriver().getCurrentUrl();
        browser.getDriver().navigate().back();
        browser.getDriver().findElement(By.id("nav-search"));
```

Method Name:

public void setCapabilities(String enableLocal, String os, String osVersion, String browser, String browserVersion)

Purpose:

This method is used to set capabilities OS, OsVersion, browser, browser Version

Returns:

Returns nothing

Method Name:

private String getGridIP()

Purpose:

This method is to get the GridIP

Throws:

NullPointerException

Returns:

Returns current driver object

..
Class:

RunModeLocal :

Method Names:

1. RunModeLocal(String)
2. RunModeLocal(String, DesiredCapabilities)
3. getDriver() (WebDriver)
4. setChromeDriver() (WebDriver)
5. setChromeDriver(DesireCapabilities) (WebDriver)
6. setFirefoxDriver() (WebDriver)
7. setFirefoxDriver(DesireCapabilities) (WebDriver)

Description:

A. Mathod Name:

Public RunModeLocal(String browser)

Purpose:
1. It is the constructor of RunModeLocal class, having one parameter string.
2. It is use to initialized reference variable of WebDriver interface to any browser class like ChromeDriver(), FirefoxDriver()

Parameters: This constructor having one parameter:

- Browser- It is string , in this variable we can wright any browser name like chrome, firefox, safari etc

Exception: NullPointerException

B. Mathod Name:

Public RunModeLocal(String browser, DesiredCapabilities capabilities)

Purpose:
1. It is the constructor of RunModeLocal class, having one parameter string.
2. It is use to initialized reference variable of WebDriver interface to any browser class like ChromeDriver(), FirefoxDriver()
3. We can configure driver instance like FirefoxDriver, Chromedriver, InternetExplorerDriver by using desired capabilities.

Parameters:

- Browser- It is name of browser.
- Capabilities- It is capabilities like FirefoxDriver, Chromedriver, InternetExplorerDriver .

Exception: NullPointerException

C. Mathod Name:

Public WebDriver getDriver()

Purpose:
1. It will get current reference variable of WebDriver Interface.

D. Mathod Name:

Private WebDriver setChromeDriver()

Purpose:
1. It will sets the chrome webdriver and give a chromedriver.exe path to it for various operating system.

E. Mathod Name:

Private WebDriver setChromeDriver(DesireCapabilities)

Purpose:
1. Desired capability can also be used to configure the driver instance of selenium WebDriver.
2. We can configure driver instance like Chromedriver by using desired capabilities.

F. Mathod Name:

Private WebDriver setFirefoxDriver()

Purpose:
1. It will sets the firefox webdriver and give a geckodriver.exe path to it for various operating system.

G. Mathod Name:

Private WebDriver setFirefoxDriver(DesireCapabilities)

Purpose:
1. It will sets the firefox webdriver.
2. We can configure driver instance like FirefoxDriver by using desired capabilities.

..

Class:

RunModeSauceLabs:

Method Names:

1. **RunModeSauceLabs(String,String,String,String)**
2. **generateHubURL() (String)**
3. **getCredentials() (String[])**
4. **getDriver() (WebDriver)**
5. **setBrowserCapability(String) (void)**
6. **setCapabilities(String,String,String,String) (void)**

Description:

A. Method name:

Public RunModeSauceLabs(String os,String osVersion,String browser,String browserVersion)

Purpose:
1. It is the constructor of RunModeSauceLabs class, which accepts OS, version of OS, Browser name, and Version of browser.

Parameters: This Constructor having four parameters:

- OS- This the name of Operating system.
- OsVersion- Version of Operating system.
- Browser- Name of Browser.
- BrowserVersion- Version of Browser.

Exception: illegalArgumentException

B. Method name:

Private String generateHubURL()

Purpose:
1. It create hub url by using username and access key.

Returns:
It returns URL.

C. Method name:

Private String[] getCredentials()

Purpose:

1. It will get the property of credential.

Exception: NullPionterException

D. Method name:

Public WebDriver getDriver()

Purpose:

2. It will get current reference variable of WebDriver Interface.

E. Method name:

Private void setBrowserCapability(String browser)

Purpose:

1. It will set or change the capabilities of give browser.

Parameters: This method having one parameter:

- Browser- provide name of Browser.

F. Method name:

Public void setCapabilities(String os,String osVersion,String browse,String browserVersion)

Purpose:

1. This method change or set the capabilities of browser.
2. Desired capability can also be used to configure the driver instance of selenium WebDriver.
3. We can configure driver instance like FirefoxDriver, Chromedriver, InternetExplorerDriver by using desired capabilities.

Parameters: This method having four parameters:

- OS- This the name of Operating system.
- OsVersion- Version of Operating system.
- Browser- Name of Browser.
- BrowserVersion- Version of Browser.

..
Class:

Scroll:

Methods:

1. public boolean down(int)

2. public void setDriverForScroll(WebDriver)

3. public boolean toView(LocatorType)

4. public boolean up(int)

A. Method Name:

public boolean down(int scrollDownRange)

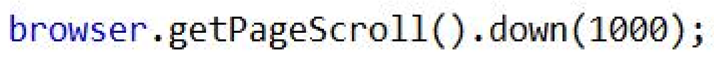
Purpose:
1. It will scrolls the screen to down direction in given range.

Parameters: This method having one parameter:

- scrollDownRange- It it the integer value provided as a range.

Returns:
It will returns true if successful otherwise returns false.

Uses:



```
browser.getPageScroll().down(1000);
```

B. Method Name:

public void setDriverForScroll(WebDriver driver)

Purpose:
1. This method accepts reference variable of WebDriver Interface, which will initialize to FirefoxDriver or InternetExplorerDriver or the ChromeDriver class constructor.
2. This method is use to refer current instance variable.

Returns:
It returns nothing.

C. Method Name:

public boolean toView(LocatorType locator, String element)

Purpose:

1. This method show the specific web element.

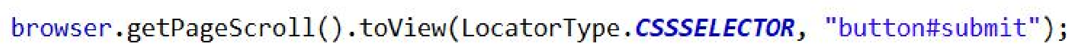
Parameters: This method having two parameter:

- Locator- It is identifier of web element.
- Element- It is value for locator.

Returns:

It will returns true if element found otherwise returns false.

Uses:



```
browser.getPageScroll().toView(LocatorType.CSSSELECTOR, "button#submit");
```

D. Method Name:

public boolean up(int scrollUpRange)

Purpose:

1. This method scrolls the screen at upward direction in given range.

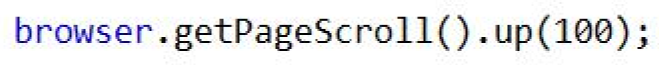
Parameters: This method having one parameter:

- scrollUpRange- It is integer value for range.

Returns:

It will returns true when screen gets scroll otherwise return false.

Uses:



```
browser.getPageScroll().up(100);
```

..

Class: TextField

Methods:

1.public boolean enterTextField(LocatorType locator, String locatorValue, String textToEnter)

2. public boolean getTextField(LocatorType locator, String locatorValue, String textToEnter)

3.public void setFinder(IFind iFind)

Method Name:

public boolean enterTextField(LocatorType locator, String locatorValue, String textToEnter)

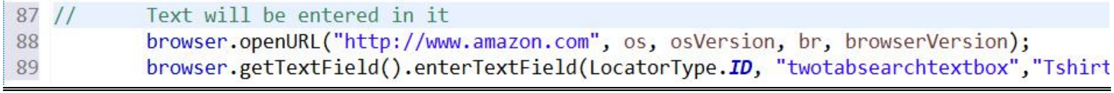
Purpose:

This method is used to enter the characters or text in the text fields

Returns:

It returns true if the text is entered in the text field or false if not

Usage:



```
//      Text will be entered in it
        browser.openURL("http://www.amazon.com", os, osVersion, br, browserVersion);
        browser.getTextField().enterTextField(LocatorType.ID, "twotabsearchtextbox","Tshirt
```

Method Name:

public boolean getTextField(LocatorType locator, String locatorValue, String textToEnter)

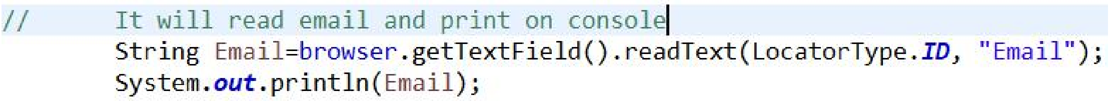
Purpose:

This method is used to read and get the characters or text from the text fields

Returns:

returns text or characters

Usage:



```
//      It will read email and print on console
        String Email=browser.getTextField().readText(LocatorType.ID, "Email");
        System.out.println(Email);
```

Method Name:

public void setFinder(IFind iFind)

Purpose:

This method is used to set Finder with help of object of IFind Interface.

Returns:

Returns nothing

..
Class:

Verify:

Methods:

1. verifyIsDisplayed(LocatorType, Boolean, String)(boolean)
2. verifyIsEnabled(LocatorType, String, Boolean, String)(boolean)
3. verifyIsSelected(LocatorType, String, Boolean, String)(boolean)
4. verifyPageTitle(String, String, String)(boolean)
5. verifyPageUrl(String, String, String)(boolean)
6. verifyText(LocatorType, String, String, String)(boolean)
7. setFind(IFind)(void)
8. getElement(LocatorType, String)(WebElement)

A. Method Name:

verifyIsDisplayed(LocatorType locator, Boolean expected, String errorMessage)

Purpose:
1. This method capable to check all kind of elements are present or not.

Parameters: This method having three parameters:

- Loactor- It is identifier of element.
- Expected- Expected boolean value after execution.
- errorMessage- If expected boolean value is not returns then string for error message.

Returns:
It returns true if element displayed otherwise false.

Throws : illegalArgumentException

B. Method Name:

public boolean verifyIsEnabled(LocatorType locator, String value, Boolean expected, String errorMessage)

Purpose:
1. This method used to verify if web element is enabled or not within the web page.
2. It is primarily used with buttons.

Parameters: This method having four parameters

- Locator- It is identifier of web element.
- Value- It is value for locator.
- Expected- It is expected boolean value after execution.
- errorMessage- If unexpected result arrive then string error message.

Returns:
It returns true if element is enabled otherwise returns false.

Throws : illegalArgumentException

C. Method Name:

public boolean verifyIsSelected(LocatorType locator, String value, Boolean expected, String errorMessage)

Purpose:
1. This method is used to verify if web element is selected or not.

Parameters: This method having four parameters

- Locator- It is identifier of web element.
- Value- It is value for locator.
- Expected- It is expected boolean value after execution.
- errorMessage- If unexpected result arrive then string error message.

Returns:
It returns true if web element is selected otherwise returns false.

Throws : illegalArgumentException

D. Method Name:

public boolean verifyPageTitle(String actual , String expected, String errorMessage)

Purpose:
1. This method used to verify current page title.

Parameters: This method having three parameters:

- Actual- It is actual web page title.
- Expected- It is what page title we expects.
- errorMessage- If actual and expected don't match then error message prints.

Returns:
It returns true if actual and expected title is similar otherwise returns false.

Throws : illegalArgumentException

E. Method Name:

public boolean verifyPageUrl(String actual, String expected, String errorMessage)

Purpose:
1. This method used to verify current URL of web page.

Parameters: This method having three parameters:

- Actual- It is actual web page URL.
- Expected- It is what page URL we expects.
- errorMessage- If actual and expected don't match then error message prints.

Returns:
It returns true if actual and expected gets match otherwise returns false.

Throws : illegalArgumentException

F. Method Name:

public boolean verifyText(LocatorType loactor, String value, String expected, String errorMessage)

Purpose:
1. This method used to verify text of web element.

Parameters: This method having four parameters

- Locator- It is identifier of web element.
- Value- It is value for locator.
- Expected- It is expected text value after execution.
- errorMessage- If unexpected result arrive then string error message.

Returns:
It returns true if expected text found in web element otherwise returns false.

Throws : illegalArgumentException

G. Method Name:

Public void setFind(IFind)

Purpose:
5. This method accepts the reference variable of IFind interface.
6. This method is use to refer instance variable.

Returns:
It will returns Nothing.

H. Method Name:

Private WebElement getElement(LocatorType locator, String value)

Purpose:
1. It is private method use to get the web element by using locator.

Parameters: This method having two parameters:

- Locator- It is Identifier of web element .
- Value- It is the value for locator.

Returns:
It returns current reference variable of WebElement Interface.

…
Class:

WaitElement:

Methods:

1. HardPause(long)(boolean)

2. implicitWait(long)(boolean)

3. pageLoadTimeout(long)(boolean)

4. safeWait(long)(boolean)

5. setDriverForWait(WebDriver)(void)

6. setScriptTimeout(long)(boolean)

7. waitFor(LocatorType , String)(boolean)

8. waitForElementPresence(LocatorType, String, long)(void)

9. waitForElementPresence(LocatorType, String , long, long)(void)

10. waitForElementToBeClickable(LocatorType, String, long)(void)

11. waitForElementToBeClickable(LocatorType, String, long, long)(void)

12. waitForElementVisibility(LocatorType, String, long)(void)

13. waitForElementVisibility(LocatorType, String, long, long)(void)

Description:

A. Method Name:

Public boolean HardPause(long milliseconds)

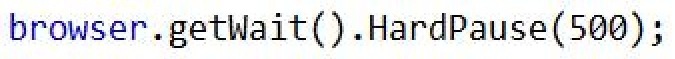
Purpose:
1. This method will stops the next line to execute until given time.
2. It will compulsory pause the execution for given time.

Parameters: This method have one parameter:

- Milliseconds: It is long value which is take as a milliseconds(time unit).

Returns:
It will return true after completing wait time otherwise returns false.

Uses:



```
browser.getWait().HardPause(500);
```

B. Method Name:

public boolean implicitWait(long seconds)

Purpose:

1. This method wait until specific element gets found on web page.
2. Some elements may be not immediately loaded on web page, this wait method helps us to pause the next line execution till element gets available.
3. This wait method will applied after every statement.

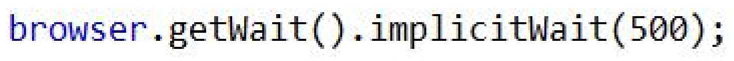
Parameters:This method having one parameter

- Seconds- This method take wait time in seconds(time unit).

Returns:
It returns true when element gets available otherwise returns false.

Exception:
1. NullPointerException - If driver is not found.

Uses:



```
browser.getWait().implicitWait(500);
```

C. Method Name:

public boolean pageLoadTimeout(long seconds)

Purpose:
1. This method sets the time limit to load the page.
2. If page gets load in given time span then only further scripts gets executed otherwise execution gets stop.

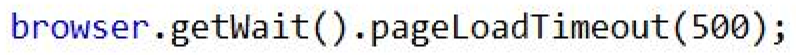
Parameters: This method having one parameter:

- Seconds- This parameter takes long value which is in seconds(time unit).

Returns:
It returns true if page gets load in given time span otherwise returns false.

Exception:
1. TimeoutException- If element is not found in given time.
2. NullPointerException - If driver is not found.

Uses:



```
browser.getWait().pageLoadTimeout(500);
```

D. Method Name:

public boolean safeWait(long milliseconds)

Purpose:

1. This method just make sure that web page gets load properly and after that next scripts will get execute.
2. This method will wait for given time span.

Parameters: This method having one parameter:

- Milliseconds- It will take the long value as a milliseconds(time unit).

Returns:

It will returns true after safe wait completed otherwise returns false.

E. Method Name:

public void setDriverForWait(WebDriver driver)

Purpose:

5. This method accepts reference variable of WebDriver Interface, which will initialize to FirefoxDriver or InternetExplorerDriver or the ChromeDriver class constructor.
6. This method is use to refer current instance variable.

Returns:

It returns nothing.

F. Method Name:

public boolean setScriptTimeout(long seconds)

Purpose:

1. This method sets time to execute test scripts within it.
2. It can sets the time span to the script so driver can not interrupt the execution of test script.

Parameters: This method having one parameter:

- Seconds- This will take long value as a seconds(time unit).

Returns:

It will returns true after time completion otherwise returns false.

Exception:

1. TimeoutException- If element is not found in given time.
2. NullPointerException - If driver is not found.

Uses:

```
browser.getWait().setScriptTimeout(5000);
```

G. Method Name:

public boolean waitFor(LocatorType , String)

Purpose:

1. This method is wait until given element gets visible.

2. Some times web elements are not available immediately , so need to wait until element gets available.

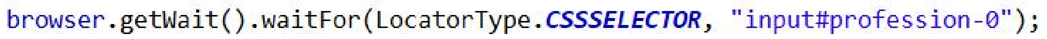
Parameters: This method having two parameters:

- LocatorType- This is the identifier of element.
- Value- This is the value for locator.

Returns:
It will returns true when element gets available otherwise returns false.

Exception:
1. TimeoutException- If element is not found in given time.
2. NullPointerException - If driver is not found.
3. NoSuchElementException- If given element is not present.

Uses:



```
browser.getWait().waitFor(LocatorType.CSSSELECTOR, "input#profession-0");
```

H. Method Name:

public void waitForElementPresence(LocatorType locator, String value, long timeOutInSeconds)

Purpose:
1. It will wait until given element gets available on page.
2. We can provide time span to load that web element.

Parameters: This method having three parameters:

- locatorType- It is identifier of element.
- Value- It is value for locator.
- timeOutInSeconds- It is time span in seconds(time unit)

Returns:
It returns true when element gets available in give time span otherwise returns false.
Exception:
1. TimeoutException- If element is not found in given time.
2. NullPointerException - If driver is not found.

I. Method Name:

public void waitForElementPresence(LocatorType locator, String value, long timeOutInSeconds, long pollingInSeconds)

Purpose:
1. It will wait until given web element gets available on page.
2. It added polling facility, which checks the availability of element after every few seconds which we provided.

Parameters: This method having four parameters:

- Locator- It is the identifier of element.
- Value - It is value for locator.
- timeOutInseconds- It is the total time to load the web element.
- pollingInseconds- It is time interval, after that interval it will check for element availability.

Returns:
It returns nothing.

Exception:

1. TimeoutException- If element is not found in given time.
2. NullPointerException - If driver is not found.
3. NoSuchElementException- If given element is not present.

J. Method Name:

public void waitForElementToBeClickable(LocatorType locator, String Value, long timeOutInSeconds)

Purpose:
1. It will wait until given element gets available for clicking purpose (buttons, links, clickable images).

Parameters:This method having three parameters:

- Locator- It is identifier of element.
- Value- It is value for locator.
- timeOutInSeconds- It is wait time in seconds.

Returns:
It will return nothing.

Exception:

1. TimeoutException- If element is not found in given time.
2. NullPointerException - If driver is not found.

K. Method Name:

public void waitForElementToBeClickable(LocatorType, String, long, long)

Purpose:
1. It will wait untile given element is available on page for clicking purpose(buttons, links, clickable images).
2. This method also provide time to check availability of element at every interval time.

Parameters: This method having four parameters:

- Locator- It is identifier of element.
- Value- It is value for locator.
- TimeOutInSeconds- It is total wait time.
- pollingInSeconds- It is interval time to check the availability of element.

Returns:
It will return nothing.

Exception:

1. TimeoutException- If element is not found in given time.
2. NullPointerException - If driver is not found.
3. NoSuchElementException- If given element is not present.

L. Method Name:

public void waitForElementVisibility(LocatorType, String, long)

Purpose:

1. It will wait until element gets visible on screen(images, links).

Parameters:This method having three parameters:

- Locator- It is identifier of element.
- Value- It is value for locator.
- TimeOutInSeconds- It is total wait time to get page load.

Returns:'

It will returns nothing.

Exception:

1. TimeoutException- If element is not found in given time.
2. NullPointerException - If driver is not found.

M. Method Name:

public void waitForElementVisibility(LocatorType, String, long, long)

Purpose:

1. It will wait until element gets visible on screen(images, links).
2. This method provides one more time interval to check element visibility at every interval.

Parameters: This method having four parameters:

- Locator- It is identifier of element.
- Value- It is value for locator.
- timeOutInSeconds- It is total time to load the page in seconds.
- pollingInseconds- It is interval time to check visibility of element after interval time.

Returns:

It will return nothing.

Exception:

1. TimeoutException- If element is not found in given time.
2. NullPointerException - If driver is not found.
3. NoSuchElementException- If given element is not present.

..
Class:

WebTable:

Method names:

1. columnCount(LocatorType,String) (int)
2. dataFromColumnByIndex(LocatorType, String, int) (List<String>)
3. dataFromRow(LocatorType, String, int) (List<String>)
4. dataFromTable(LocatorType, string) (List<Object>)
5. dataFromTableCell(LocatorType,String, int,int) (String)
6. rowCount(LocatorType, String) (int)
7. setDriverForWebTable(WebDriver) (void)
8. setFinder(IFind) (void)
9. tableElementFinder(LocatorType, String) (WebElement)

Description:

A. Method Name

Public int columnCount(LocatorType,String)

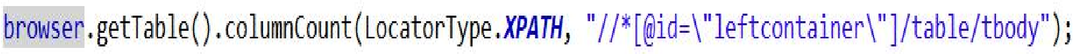
Purpose:
1. It will give us count of columns present in web page.

Parameters: This method having two parameters:
- Locator- It is identifier of web element.
- Value- It is value for locator.

Returns:
It returns count of columns.

Exceptions: noSuchElementException

Uses:



```
browser.getTable().columnCount(LocatorType.XPATH, "//*[@id=\"leftcontainer\"]/table/tbody");
```

B. Method Name

Public List<String> dataFromColumnByIndex(LocatorType, String, int)

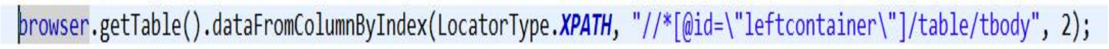
Purpose:
1. This method gives us list of values inside column, which is base on there index.

Parameters: This method having three parameters:

- Locator- It is identifier for web element.
- Value- It is value for locator,
- Index- It is index of column data.

Returns:
It returns list of data included in column.

Uses:



```
browser.getTable().dataFromColumnByIndex(LocatorType.XPATH, "//*[@id=\"leftcontainer\"]/table/tbody", 2);
```

C. Method Name

Public List<String> dataFromRow(LocatorType, String, int)

Purpose:
1. This method gives us list of data included in row.

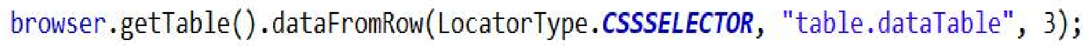
Parameters: This method having three parameters:

- Locator- It is identifier for web element.
- Value- It is value for locator.
- Index- It is index number of row.

Returns:
It returns list of data included in row.

Exceptions: NoSuchElementException

Uses:



```
browser.getTable().dataFromRow(LocatorType.CSSSELECTOR, "table.dataTable", 3);
```

D. Method Name

Public List<Object> dataFromTable(LocatorType, string)

Purpose:
1. It will give us list of data included in table.

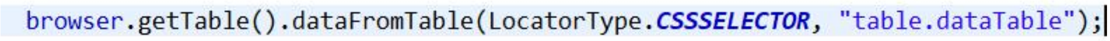
Parameters: This method having two parameters:

- Locator- It is identifier for web element.
- Value- It is value for locator.

Returns:
It returns list of data included in table.

Exceptions: noSuchElementException

Uses:



```
browser.getTable().dataFromTable(LocatorType.CSSSELECTOR, "table.dataTable");
```

E. Method Name

Public String dataFromTableCell(LocatorType,String, int,int)

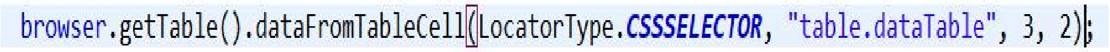
Purpose:
1. This method gives us specific cell in table.

Parameters: This parameter having four parameters:

- Locator- It is Identifier of web element.
- Element- It is value for locator.
- rowIndex- It is index of row.
- columnIndex- It is index of column.

Returns:
It returns that value which in present at given row index and column index.

Uses:



```
browser.getTable().dataFromTableCell(LocatorType.CSSSELECTOR, "table.dataTable", 3, 2);
```

F. Method Name

Public int rowCount(LocatorType, String)

Purpose:
1. This method provides count of row.

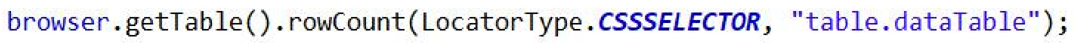
Parameters: This method having two parameters:

- Locator- It is identifier of web element.
- element- It is value for locator.

Returns:
It returns number of row count.

Exceptions: noSuchElementException

Uses:



```
browser.getTable().rowCount(LocatorType.CSSSELECTOR, "table.dataTable");
```

G. Method Name

Public void setDriverForWebTable(WebDriver)

Purpose:
7. This method accepts reference variable of WebDriver Interface, which will initialize to FirefoxDriver or InternetExplorerDriver or the ChromeDriver class constructor.
8. This method is use to refer current instance variable.

Returns:
It will returns nothing.

H. Method Name

Public void setFinder(IFind)

Purpose:

7. This method accepts the reference variable of IFind interface.
8. This method is use to refer instance variable.

Returns:
It will returns Nothing.

I. Method Name

Private WebElement tableElementFinder(LocatorType, String)

Purpose:

1. This method finds the elements into the table.

Parameters: This method having two parameters:

- Locator- It is identifier of web element.
- Element- It is value for locator.

Returns:
It returns reference variable of WebElement interface.

..

Class: WindowHandlerImpl

Methods:

1. public boolean closeAllOtherWindows(String
   openWindowHandle)

2. public boolean closeWindow(String windowNameOrHandle)

3. public String getMainWindowHandle()

4. public Set<String> getWindowHandles()

5.public String getWindowTitle(String
windowNameOrHandle)

6.public WindowHandler.FrameHandler handleFrame()

7. public WindowHandler.PopUpHandler handlePopup()

8. public void setDriver(BrowserType browserType)

9.public boolean switchToWindow(String
windowNameOrHandle)

10.public boolean waitForNewWindowAndSwitchToIt(int
noOfSecsToWait)

Method Name:

public boolean closeAllOtherWindows(String
openWindowHandle)

Purpose:

This method will close all other Windows other than open window

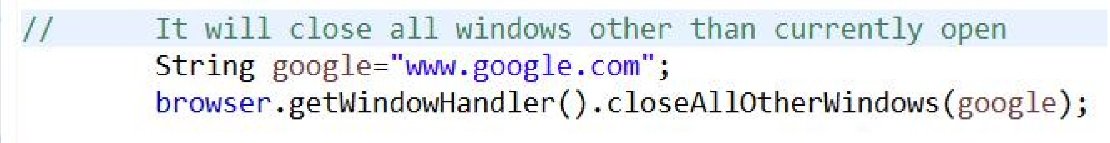
Throws:

NoSuchWindowException

Returns:

Returns true or false for all windows operations

Usage:



```
//      It will close all windows other than currently open
        String google="www.google.com";
        browser.getWindowHandler().closeAllOtherWindows(google);
```

Method Name:

public boolean closeWindow(String windowNameOrHandle)

Purpose:

This method will close Window with given name

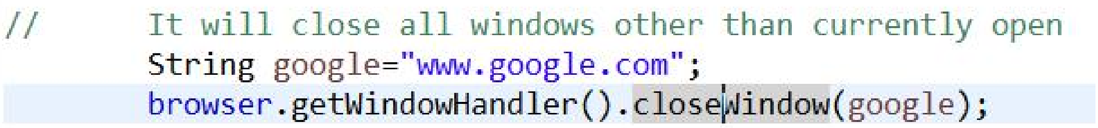
Throws:

NoSuchWindowException

Returns:

Returns true or false for close Window operations

Usage:



```
//      It will close all windows other than currently open
        String google="www.google.com";
        browser.getWindowHandler().closeWindow(google);
```

Method Name:

public String getMainWindowHandle()

Purpose:

This method is used to Get Main window title

Returns:

Returns object of getMainWindowHandler

Method Name:

public Set<String> getWindowHandles()

Purpose:

This method is used to Get list of Main window title

Returns:

Returns list of getMainWindowHandler

Method Name:

public String getWindowTitle(String windowNameOrHandle)

Purpose:

This method is used to Get Title for given window name.

Returns:

Returns title

Method Name:

public WindowHandler.FrameHandler handleFrame()

Purpose:

This method is used to Get object of FrameHandler Interface.

Returns:

Returns FrameHandler object

Method Name:

public WindowHandler.PopUpHandler handlePopup()

Purpose:

This method is used to Get object of PopUpHandlerHandler Interface.

Returns:

Returns PopUpHandlerHandler object

Method Name:

public void setDriver(BrowserType browserType)

Purpose:

This method will set the path for driver

Returns:

Returns nothing

Usage:

Used to set Driver.

Method Name:

public boolean switchToWindow(String

windowNameOrHandle)

Purpose:

This method will switches to window with given window name.

Returns:

Returns true or false for switch window operations

Method Name:

public boolean waitForNewWindowAndSwitchToIt(int noOfSecsToWait)

Purpose:

This method will Waits for new window for the given time in seconds and then switches to it

Returns:

Returns true or false for new window and switch operations
