**SELENIUM WEB DRIVER**

I**nterface**:

A Java interface defines a set of methods but does not implement them. A class that implements the interface agrees to implement all of the methods defined in the interface.

**Web driver is the Interface**

**All Known Implementing Classes:**

[AndroidDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/android/AndroidDriver.html), [AndroidWebDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/android/library/AndroidWebDriver.html), [ChromeDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/chrome/ChromeDriver.html), [EventFiringWebDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/support/events/EventFiringWebDriver.html), [FirefoxDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/firefox/FirefoxDriver.html), [HtmlUnitDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/htmlunit/HtmlUnitDriver.html), [InternetExplorerDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/ie/InternetExplorerDriver.html),[IPhoneDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/iphone/IPhoneDriver.html), [IPhoneSimulatorDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/iphone/IPhoneSimulatorDriver.html), [RemoteWebDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/remote/RemoteWebDriver.html), [SafariDriver](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/safari/SafariDriver.html)

**i) To create driver object for chrome**

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

WebDriver driver = **new** FirefoxDriver();

WebDriver driver = **new**  [EdgeDriver](http://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/edge/EdgeDriver.html)Driver();

WebDriver driver = **new**  [**InternetExplorer**](http://seleniumhq.github.io/selenium/docs/api/java/org/openqa/selenium/ie/InternetExplorerDriver.html) **D**river();

* Import package and chrome driver.

We cannot invoke browsers in third-party APIs directly bcz of security issue, so we have to invoke browser by download “**.exe file”.** And set location. By using set property.

**System. Set property (“webdriver.chrome.driver”, c:\\chromedriver.exe”);**

**ii) To get url**

driver.get(“www.google.com”)

**iii) To get title**

driver.getTitle();

**iv) To get page source**

driver.getpagesource();

**v) To go back and front**

drive.navigate().back();

drive.navigate().forward();

**vi) To close windows or quit the browser**

driver.close();

driver.quit();

**Locators techniques and tools for web objects**

**i) Locators:**

* Id
* Name
* Class name
* Lin text
* Xpath
* Css
* Evey object may not have ID, className or name- Xpath and CSS Preferred
* Alpha numeric id may vary on every refresh- check
* Confirm the link object with anchor "a" tag
* Classes should not have spaces- Compound classes cannot be accepted
* Multipl values - Selenium identifies the first one- Scans from top left
* Double quotes inside double quotes are not accepted
* Xpath/CSS can be defined in n number of ways
* Rightclick copy on blue highlighted html code to generate xpath
* Firepath depreciated from firefox-
* when xpath starts with html-Not reliable- Switch browser to get another one
* There is no direct way to get CSS in chrome. You will find it in tool bar
* Degrade browser to less firefox 55 to ge Firepath
* $("") - for css , $x("") or xpath

**ii) To find element by locator:**

driver.findelement(by.xpath(“ ”)).sendkeys( “ ”).

**How to validate Xpath and CSS accuracy From Browser:**

**For Xpath:**

* Get Xpath From Browser
* Go to console in browser
* And enter $x {“Xpath”}

**For CSS:**

* Get Css From Browser
* Go to console in browser
* And enter $ {“Css”}

**Customized css and xpath generation technique:**

**Xpath Syntax:**

<input type="text" name="username" maxlength="50" value="" tabindex="12" id="username" class="form-control" autocomplete="off" autocapitalize="off" autocorrect="off" aria-describedby="helpBlock">

In this :

Input = html tag

**name="username"**

attribute = name,

value = “username”

**autocapitalize="off"**

attribute = autocapitalize

value: “off”

**Syntax:**

**//tagname[@attribute=”value”]**

**//\*[@attribute=”Value”] to get xpath from whole page**

**Ex:**

<input type="text" name="username" maxlength="50" value="" tabindex="12" id="username" class="form-control" autocomplete="off" autocapitalize="off" autocorrect="off" aria-describedby="helpBlock">

**Xpath: //input[@type="text"]**

**CSS Syntax:**

**tagname[attribute=”value”]**

**or**

**[attribute=”value”] to get CSS from whole page**

**Or**

**TagName#id**

**Or**

**Tagname.classname**

**How to write regular expression for css and xpath:**

When we dnt have many attributes, and when values are keep changing,

Xpath:

// tagname[contains(@atributes,’value’)]

If Value is very big we can use any unique part from value.

Css:

Tagname[attribute\*=’value’]

**Parent Child Relationship:**

**Xpath:**

**//parenttagname[@attribute=’value’]/ first child tagname/ add more child tagname[position of child ex: 1, 2,3 ]/tagname of child/tagname of link.**

**Difference b/w Absolute and relative Xpath:**

**Relative Xpath:**

**Java Data Base Connection**

To validate data from database for test suite we are going to use JDBC connection.

We Need

* DB Credentials
* JDBC Jar (https://mvnrepository.com/artifact/mysql/mysql-connector-java/5.1.42\_)

**CODE:**

**public** **class** JDBConncetion {

**public** **static** **void** main(String[] args) **throws** SQLException {

String host ="localhost";

String port ="3306";

Connection con = DriverManager.*getConnection*("jdbc:mysql://" + host +":"+ port + "/employeeportal","root","root");

Statement a = con.createStatement();

ResultSet rs =a.executeQuery("select \* from Employeeinfo where id = 56784");

**while**(rs.next()) // It will move your result from baseindex to first index

{

System.***out***.println(rs.getString("name"));

System.***out***.println (rs.getString("dept"));

System.***out***.println(rs.getInt("age"));

//DriverManager.getConnection("jdbc:mysql://"+localhost+":"+3306+"/employeeportal", "root", "root")

}

}

}