**Desired Capabilities**

* Basically, the DesiredCapabilities help to set properties for the WebDriver opened browser.
* DesiredCapabilities are used like setting the browser configuration your with key-value pair.
* So we generally set DesiredCapabilities for conveying our WebDriver opened browser for specific setting we will be needing to run our test.
* The Desired Capabilities class provides a setCapabilityMethod () to set the different capabilities in a browser like accepting SSL certificates, enabling JavaScript, querying the browser location is allowed or not etc.

**Common Methods for all the browser to change the setting**

1. **Handle the SSL certificate:**

* CapabilityType.ACCEPT\_SSL\_CERTS

For enabling or a disabling a browser session to accept the SSL certificates by default.

1. **Application Cache Enabled:**

* CapabilityType.SUPPORTS\_APPLICATION\_CACHE

For checking if the browser instance is capable of interacting with application cache.

1. **CSS Selecters Enabled:**

* CapabilityType.SUPPORTS\_FINDING\_BY\_CSS

For checking if the use of CSS selector for locating web elements is enabled or not.

1. **javascriptEnabled:**

* CapabilityType.SUPPORTS\_JAVASCRIPT

For checking if javascript execution is enabled or not in the browser instance.

**How to maximize chrome browser**

import org.openqa.selenium.chrome.ChromeOptions;

ChromeOptions chOptions = new ChromeOptions();

chOptions.addArguments(“start-maximized”);

**How to Handle SSL Certificate in Chrome Browser:**

//Specifying desired capabilities for Chrome browser

DesiredCapabilities acceptSSLCertificate = DesiredCapabilities.chrome();

//Setting capability to accept SSL certificates

acceptSSLCertificate.setCapability(**CapabilityType.ACCEPT\_SSL\_CERTS**, true);

//Binding the capabilities to a new instance of chrome browser

WebDriver driver = new ChromeDriver(acceptSSLCertificate);

**How to Handle SSL Certificate in IE:**

// we can handle SSL certificate in IE by using java Script:

driver.get("javascript:document.getElementById('overridelink').click();");

**How to change the IE setting at the time of Execution:**

DesiredCapabilities cap = new DesiredCapabilities();

cap.setCapability("ignoreProtectedModeSettings", true); cap.setCapability("ignoreZoomSetting", true);

cap.setCapability("disable-popup-blocking", true);

cap.setCapability("nativeEvents", false);

cap.setCapability("unexpectedAlertBehaviour", "accept");

cap.setCapability (InternetExplorerDriver.INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS,true);

cap.setCapability(InternetExplorerDriver.IGNORE\_ZOOM\_SETTING, true);

cap.setCapability(CapabilityType.ACCEPT\_INSECURE\_CERTS, true);

cap.setCapability(CapabilityType.BROWSER\_NAME, "chrome");

String c1= CapabilityType.BROWSER\_VERSION;

System.out.println(c1);

CSS SELECTOR IN SELENIUM WEBDRIVER

**Using Id**

**Syntax - #id**

**Ex:**

<button id="submitButton1" type="button" class="btn">Submit</button>

CSS Locator - **#submitButton1**

Description - '#submitButton1' will select the element with id 'submitButton1'.

### Using class

**Syntax - .class**

Ex:

<button id="submitButton1" type="button" class="btn">Submit</button>

CSS Locator - **.btn**

Description - '.btn' will select all the elements with class 'btn'.

### Using tag

**Syntax - tagName**

Ex:

<input id="fname" type="text" name="firstName" class="textbox">

CSS Locator – input

Description - 'input' will select all the input type elements.

### Using attributes and their value

**Syntax - [attributeName='attributeValue']**

Ex:

<input id="fname" type="text" name="firstName" class="textbox">

CSS Locator - [name='firstName']

Description - [name='firstName'] will select the elements with name attribute having value 'firstName'.

### Using tags and id

**CSS Selector Rule - tag#id**

Ex:

<input id="fname" type="text" name="firstName" class="textbox">

CSS Locator - input#fname

Description - input#fname will select the 'input' element with id 'fname'.

### Using tags and class

**Syntax - tag.class**

Ex:

<input id="fname" type="text" name="firstName" class="textbox">

CSS Locator - input.textbox

Description - input.textbox will select the 'input' element with id 'textbox'.

### Using tags and attributes

**Syntax - tag[attributeName='attributeValue']**

**Ex:**

<input id="fname" type="text" name="firstName" class="textbox">

CSS Locator - input[name='firstName']

Description - input[name='firstName'] will select the 'input' element with 'name' attribute having value 'firstName'.

### Locating Child Elements (direct child only)

**Syntax: - parentLocator>childLocator**

Ex:

<div id="buttonDiv" class="small">

<button id="submitButton1" type="button" class="btn">Submit</button>

</div>

CSS Locator - div#buttonDiv>button

Description - 'div#buttonDiv>button' will first go to div element with id 'buttonDiv' and then select its child element - 'button'.

### Locating elements inside other elements (child or subchild)

**Syntax - locator1 locator2**

EX:

<div id="buttonDiv" class="small">

<button id="submitButton1" type="button" class="btn">Submit</button>

</div>

CSS Locator - div#buttonDiv button

nav[role='navigation'] a[title='NEW IN']

Description - 'div#buttonDiv button' will first go to div element with id 'buttonDiv' and then select 'button' element inside it (which may be its child or sub child).

### Nth child

**Syntax - ParentCSS: nth-child(n)**

**Ex:**

<ul id="testingTypes">

<li>Automation Testing</li>

<li>Performance Testing</li>

<li>Manual Testing</li>

</ul>

CSS Locator - #testingTypes li:nth-child(2)

Description - '#testingTypes li:nth-child(2)' will select the element with id 'testingType' and then locate the 2nd child of type li i.e. 'Performance Testing' list item.

### Locating Siblings

**CSS Selector Rule - locator1+locator2**

Ex:

<ul id="testingTypes">

<li id="automation">Automation Testing</li>

<li>Performance Testing</li>

<li>Manual Testing</li>

</ul

CSS Locator - li#automation + li

Description - 'li#automation + li' will first go to li element with id 'automation' and then select its adjacent li i.e. 'Performance Testing' list item.

nav[role='navigation'] li + li>a[id='topNav\_NEW IN']

### ^ - Starts with

**CSS Selector Rule - [attribute^=attributeValue]**

**Ex:**

<button id="user1\_btn\_263" type="button" class="btn">Submit</button>

button[id^='user1']

CSS Locator - id^="user1"

Description - 'id^="user1"' will select the element whose id starts with "user1" value

### $ - Ends with

**CSS Selector Rule - [attribute$=attributeValue]**

EX:

<button id="user1\_btn\_263" type="button" class="btn">Submit</button>

CSS Locator - id$="btn\_263"

Description - 'id$="btn\_263"' will select the element whose id ends with "btn\_263" value

### \* - Contains

**CSS Selector Rule - [attribute\*=attributeValue]**

<button id="user1\_btn\_263" type="button" class="btn">Submit</button>

input[id\*='String']

CSS Locator - id\*="btn"

Description - 'id\*="btn"' will select the element whose id contains with "btn" value

**MAVEN**

*Maven* is a powerful build tool for Java software projects. Actually, you can build software projects using other languages too, but Maven is developed in Java so it is mainly used for Java project.