How to handle Bootstrap dropdown:

For select dropdown,when we spy that particular dropdown,we have <select>tag in it.

Bit for bootstap dropdown,we don’t have a <select>tag in it.

So,we cannot use select class here to select the option in it.

Bootstrap dropdown will be a button and under that,there will be a list

*driver*.findElement(By.*id*("btndropdown")).click();

List<WebElement> lst=*driver*.findElements(By.*xpath*("//\*[@class='dropdown-menu']//li//label"));

System.***out***.println(lst.size());

**for** (**int** i=0;i<lst.size();i++)

{

String text=lst.get(i).getText();

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

if(text.equals("Bootstrap"))

{ lst.get(i).click();

break;}

1.first,click the bootstrap button dropdown

2.then,get all the items in a list using customized x path

Here,hierarchy is <ul class="dropdown-menu">

<li>

<label>

So,first write the xpath for ul and then next parent li and then next parent label .

3.then,get the text of all the elements in the list and then checks the condition.

How to use selenium with sikuli?

1.used to automate desktop application

2.used to automate image processing

3.automate flash objects-youtube video.for.eg.to pause the video and play the video.

4.automation is done using images .

Steps:

1.download sikuli

Latest version -sikuliX

Either download and add jar file or create a maven proj and add sikuli and selenium dependency

Limitations:

It cannot open any browser or url

So,we need selenium webdriver to use siluli,so we can integrate selenium test cases with sikuli.

e.g.

*driver*.get("https://www.youtube.com/watch?v=CQ9iQhBRnlU");

//impoet screen class to sikuli.script

Screen s=**new** Screen();

//import pattern class to pattern.script

Pattern pauseimg=**new** Pattern("pause.png");

s.wait(pauseimg,2000);

sometimes,video will not be paused in a single click,so use s.click double times.

s.click();

s.click();

here,in youtube🡪capture the screenshot🡪prntscr(keyboard)

open microsoft paint🡪paste

crop the pause button🡪save in png format’

save in the proj folder in eclipse under target folder

Headless browser testing using chrome:

🡪chrome browser will not be launched and test cases will be executed,so the execution will be very fast.

🡪esp.used with amazon web services and linux machines where we don’t have a GUI

🡪also used with Jenkins

e.g.

ChromeOptions options=**new** ChromeOptions();

options.addArguments("window-size=1400,800");

//headless argument shd be passed

options.addArguments("headless");

//options ref variable shd be passed to chrome driver

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

*driver*.get("https://www.google.ca/");

String title= *driver*.getTitle();

Note:

1.window size is given mandatory,because if not given it may not work with few applications that are not compatible.

2.used only if chrome version is greater than 60

What happens?

Chrome will be launched but we cannot view the page.

It directly intearacts with HTML DOM element

Selenium webdriver architecture:

Eclipse client

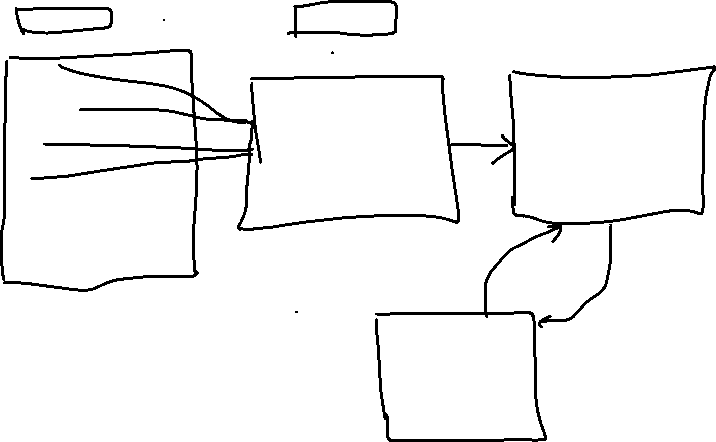
Ruby Rest

C#,dotnet

Java JSON Wire protocol Browser

PHP Over HTTP drivers

Chrome,IE



HTTP server,HTTP over

Real browsers

Chrome,safari,

IE,opera,FF

Rest->Representational stateful APIs

We can send the request ,and get the response over the internet thru Rest APIs.

Flow:

Request : --------------------------------------🡪 JSON file will be created---------------🡪servers /Browser drivers---



🡪chrome browser

e.g. Key,value pair chrome driver,FF driver

webdriver driver =new chromedriver;



driver.get(); chrome browser



complete details about API s and architecture:

google.com🡪json wire protocol selenium🡪<https://github.com/SeleniumHQ/selenium/wiki/JsonWireProtocol>