1.SELENIUM OVERIVEW

Introduction:

1. Selenium **is an open source tool and software testing tool for testing web application.**
2. It has capabilities to operate the across different browsers.
3. It has capabilities to operate the multiple operating system.

It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works.

# Features: -

1. Open source (free of cost)
2. Support for multiple browsers
3. Support for all top languages
4. Parallel execution.

# Limitations:

1. selenium does not support for desktop applications.
2. Selenium does not support for SOAP/REST Api (Rest Assure).
3. It is not possible automate the images (Sikuli).
4. It is not flexible for the reports

## 2.SELENIUM COMPONENTS:

Component is nothing but the versions of the selenium.

1. Selenium IDE
2. Selenium RC
3. Selenium WebDriver
4. Selenium grid

# Selenium IDE:

Selenium ide is a Firefox plugin

You can record actions of their test cases

# Selenium RC:

Selenium RC stands for selenium remote control. (selenium 1.0 version)

Selenium RC supports for program languages like java, c#, etc…

# Selenium GRID:

Selenium grid is a tool that can be used for remote and parallel execution.

Selenium grid can be configured with both RC,Web drivers versions.

Using Selenium Grid will help in reducing the execution time drastically.

# Selenium Web Driver:

Selenium WebDriver [Selenium 2.0 Version] is the successor to Selenium RC [Selenium 1.0] Version.

Selenium WebDriver will send commands directly to the browser and retrieves results.

Selenium WebDriver has Robust and powerful methods to can be used easily.

Web driver Methods:

WebDriver is an interface.

Web driver has 10 methods

1.get ()🡪 used to launch URL and it wait for the until all the elements of web page are loaded.

2. navigate ()🡪this method also used to launch the URL and it has the back and forward methods to navigate across the web pages.

3. findElement ()🡪 this method is used to find the web element in the web page.

4.findElements ()🡪 this is same as like the findElement () but its return type list.

5.getCurrentURL ()🡪this method is used to get the current URL of the web driver instance

6. getTitle ()🡪 this method is used to get title of the current web driver instance.

7. getWindowHandle ()🡪is used to get the runtime generated page id of web driver instance.

8. getWindowHandles ()🡪 This method is used to get all the window handles of the diver instance.

9.close ()🡪is used to close the current window of the driver instance.

10. quite ()🡪 quit method will close all the windows of the driver instance.

|  |
| --- |
| **package** org.openqa.selenium;  **import** org.openqa.selenium.logging.LoggingPreferences;  **import** org.openqa.selenium.logging.Logs;  **import** java.net.URL;  **import** java.util.List;  **import** java.util.Set;  **import** java.util.concurrent.TimeUnit;  **public** **interface** WebDriver **extends** SearchContext {    **void** get(String url);  String getCurrentUrl();  String getTitle();  List<WebElement> findElements(By by);  WebElement findElement(By by);  String getPageSource();  Set<String> getWindowHandles();  String getWindowHandle();    TargetLocator switchTo();  Navigation navigate();    Options manage();  **interface** Options {  **void** addCookie(Cookie cookie);  **void** deleteCookieNamed(String name);  **void** deleteCookie(Cookie cookie);  **void** deleteAllCookies();  Set<Cookie> getCookies();  Cookie getCookieNamed(String name);  Timeouts timeouts();  ImeHandler ime();  Window window(); |
| }  **interface** Timeouts {    Timeouts pageLoadTimeout(**long** time, TimeUnit unit);  }    **interface** TargetLocator {    WebDriver frame(**int** index);  WebDriver frame(String nameOrId);  WebDriver frame(WebElement frameElement);  WebDriver parentFrame();  WebDriver window(String nameOrHandle);  WebDriver defaultContent();  WebElement activeElement();  Alert alert();  }  **interface** Navigation {  **void** back();  **void** forward();  **void** to(String url);  **void** to(URL url);  **void** refresh();  }  **interface** ImeHandler {  List<String> getAvailableEngines();  String getActiveEngine();  **boolean** isActivated();  **void** deactivate();  **void** activateEngine(String engine);  }} | | | | |
| **//get method example**  **package** com.praneeth;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** App {  **public** **static** **void** main(String[] args) {  System.*setProperty*("webdriver.chrome.driver", "D:\\chrome\\chromedriver\_win32 (2)\\chromedriver.exe");  // create an instance for web driver by using ChromeDriver class  WebDriver driver = **new** ChromeDriver();  System.***out***.println("driver:" + driver);  // open the google url by using get method  driver.get("https://www.google.com");  }  } | | |

//using navigation methods

**interface** Navigation {

**void** back();

**void** forward();

**void** to(String url);

**void** to(URL url);

**void** refresh();

}

|  |
| --- |
| **package** com.praneeth;  **import** org.openqa.selenium.WebDriver;  **import** org.openqa.selenium.chrome.ChromeDriver;  **public** **class** App {  **public** **static** **void** main(String[] args) {  System.*setProperty*("webdriver.chrome.driver", "D:\\chrome\\chromedriver\_win32 (2)\\chromedriver.exe");  // create an instance for web driver by using ChromeDriver class  WebDriver driver = **new** ChromeDriver();  System.***out***.println("driver:" + driver);  // open the google url by using get method  driver.get("https://www.google.com");  // navigate to facebook application  driver.navigate().to("https://facebook.com");  // navigate back to google web page  driver.navigate().back();  // navigate forward to facebook  driver.navigate().forward();  }  } |

# findElement():

WebElement findElement();

This method is used to find a web element in the current document of driver instance based

on the node properties.

the following are the sub methods that findElment supports

* sendKeys – to enter text
* click – to perform click operation
* clear – to clear the existing contents
* isDisplayed – to verify the existence – true/false
* isEnabled – to verify if the field is enabled – true/false
* isSelected – if the checkbox is selected – true/false
* getAttribute – to get the attribute value/property value.

Practical: