**AUTOMATION TESTING**

**What is automation testing?**

Automation testing is a testing technique that performs testing using software testing tools to execute a testing suite. The automation testing software can enter test data under test, it compare actual result with expected result to generate detailed test report.

Software test automation requires considerable amount of money and resources. Successive development cycle will require repeatedly test the same test suite. In automation testing tool, it's possible to record and replay the test suite when it is required. No human intervention needed once test suite is automated. The goal of automation testing is to reduce the number of test cases run on manually. Test automation is the best way to increase effectiveness, test coverage and speed in software testing.

Automated software testing is important due to the following reasons:

* In manual testing workflows all fields and negative scenario consuming time and money.
* Multilingual sites are difficult to test manually.
* Test automation doesn’t require human intervention you can run automatic testing unattended. There is no need to check once you start testing it complete its process without human support.
* Automation testing increase the speed of execution.

**Which test cases to automate?**

* High Risk – Business Critical test cases
* Test cases that are repeatedly executed
* Test Cases that are difficult to perform manually
* Test Cases which are time-consuming

**Which test cases are not suitable for automation?**

* Test cases that are newly designed and not executed once manually.
* Test cases for which the requirements are frequently changing
* Test cases that can be executed on Ad-hoc basis.

**What are different automation tools?**

1. **testRigor**
2. **Ranorex**
3. **Kobiton**
4. **LambdaTest**
5. **Avo Assure**
6. **Subject7**
7. **Selenium**

**Selenium Introduction**

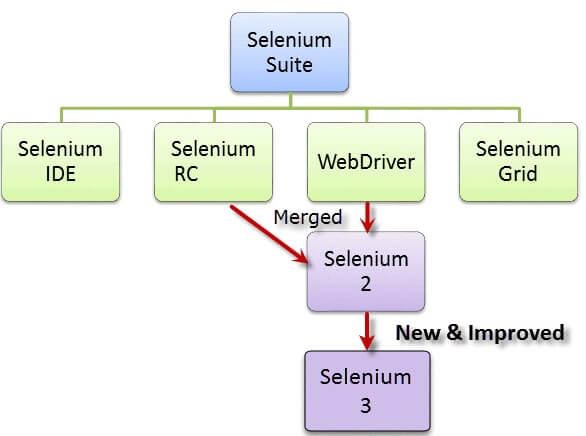
**What is selenium?**

Selenium is a free open-source automated testing framework which is used to automate different browser and web applications in different platforms. We can use multiple programming languages like c, c#, python, etc. Testing done with selenium testing tool which is known as selenium testing.

**Selenium tool suite:**

Selenium Software is not just a single tool but a suite of software, each piece catering to different Selenium QA testing needs of an organization. Here is the list of tools

* Selenium Integrated Development Environment (IDE)
* Selenium Remote Control (RC)
* WebDriver
* Selenium Grid



**What is selenium IDE?**

Selenium Integrated Development Environment (IDE) is a chrome and Firefox extension that can automate a browser through record and playback feature. Selenium IDE is the simplest framework in selenium suite easiest one to learn. It is a Chrome and Firefox plugin that you can install as easily as you can with other plugins. Selenium IDE should only be used as a prototyping tool.

**Pros and Cons of Selenium IDE:**

|  |  |
| --- | --- |
| Pros | Cons |
| Very easy to install and use. | Available only in Firefox and chrome |
| No programming experience is needed only knowledge of HTML and DOM is required. | Designed only to create prototypes of test. |
| Can exports test to formats usable in selenium Rc and WebDriver | No support for iteration and conditional operations. |
| Has built-in help and test result reporting module | Test execution is slow compare with selenium Rc and WebDriver. |
| Provide support for extension. |  |

Work flow of Selenium IDE:

Chrome Extension->Chrome Web Store->search on selenium Ide-> 1’st selenium Ide->Add to Chrome ->Add Extension->Enter web link->extension symbol->Selenium IDE->create new project->name for project->click Record->Stop->open Ide on task bar->Play.

**What is Selenium Remote Control (RC)?**

Testers using Selenium Core had to install the whole application under test and the web server on their own local computers because of the restrictions imposed by the same origin policy to overcome this Selenium RC create a server that will act as an HTTP proxy to “trick” the browser into believing that Selenium Core and the web application being tested come from the same domain. This system became known as the Selenium Remote Control or Selenium 1. Selenium RC was the flagship testing framework of the whole Selenium project for a long time. This is the first automated web testing tool that allows users to use a programming language they prefer.

**Pros and Cons of Selenium RC:**

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Cross browser and cross platform. | Installation is more difficult than Selenium IDE. |
| Can performing looping and condition operations. | Must have programming knowledge. |
| Can support data driven testing. | Needs Selenium RC server to be running |
| Has matured and complete API | API contains repeated and confusing commands |
| Can readily support new browsers. | Browser interaction is less realistic |
| Faster execution than Selenium IDE | Inconsistent result and uses on JavaScript |
|  | Slower execution time than web Driver. |

**What is Selenium WebDriver?**

It was the first cross-platform testing framework that could control the browser from the OS level. WebDriver is better than Selenium IDE and RC. It implements more stable and modern approach on browser actions. It controls the browser by directly communicating on it.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Simpler installation than Rc | More complicated installation than IDE. |
| Communicating direct to the browser. | It requires more programming knowledge. |
| Browser interaction is more realistic. | Connot readily supports new browser. |
| No need for separate component such as RC server | No built-in mechanism for logging runtime messages and generate test report. |
| Faster execution timing than IDE and RC. |  |

**What is Selenium grid?**

Selenium grid is a tool used together with selenium RC and to run parallel test across different test machines and different browsers at same time. It was capable of capturing browser screenshots during significant stages, and also of sending out Selenium commands to different machines simultaneously.

**Features:**

* Enables simultaneous running of tests in multiple browsers and environments.
* Saves time enormously.
* Utilizes the hub-and-nodes concept. The hub acts as a central source of Selenium commands to each node connected to it.

## **Importing Packages:**

To get started, you need to import following two packages:

**org.openqa.selenium.\***– contains the WebDriver class needed to instantiate a new browser loaded with a specific driver

**org.openqa.selenium. WebDriver;** – contains the WebDriver class needed to instantiate a Firefox-specific driver onto the browser instantiated by the WebDriver class.

## **Instantiating objects and variables**

Normally, this is how a driver object is instantiated.

WebDriver driver=new ChromeDriver();

A ChromeDriver class with no parameters means that the default Chrome profile will be launched by our Java program. The default Chrome profile is similar to launching Chrome in safe mode (no extensions are loaded).

For convenience, we saved the Base URL and the expected title as variables.

## **Launching a Browser Session**

WebDriver’s **get()** method is used to launch a new browser session and directs it to the URL that you specify as its parameter.

driver.get(Base url);

Eg: driver.get("https://www.amazon.com/");

## **Get the Actual Page Title**

The WebDriver class has the **getTitle()** method that is always used to obtain the page title of the currently loaded page.

actualTitle=driver.getTitle();

Eg: String title=driver.getTitle();

## **Compare the Expected and Actual Values**

This portion of the code simply uses a basic Java if-else structure to compare the actual title with the expected one.

if(expectedTitle.equals(title))

{

System.out.println("Titles are equal");

}

else

{

System.out.println("Titles are not equal");

}

## **Terminating a Browser Session**

The “**close()**” method is used to close the browser window.

driver.close();

**Quite browser Session**

The “Quit” method is used to quit the browser window.

driver.quit();

**Get Commands:**

**get()**

* It automatically opens a new browser window and fetches the page that you specify inside its parentheses.
* It is the counterpart of Selenium IDE’s “open” command.
* The parameter must be a **String** object.

**getTitle()**

* Needs no parameters
* Fetches the title of the current page
* Leading and trailing white spaces are trimmed
* Returns a null string if the page has no title

**getPageSource()**

* Needs no parameters
* Returns the **source code of the page** as a String value

String pageSource=driver.getPageSource();

**getCurrentUrl()**

* Needs no parameters
* Fetches the string representing the **current URL** that the browser is looking at

String url=driver.getCurrentUrl();

**getText()**

* Fetches the **inner text** of the element that you specify