

# Architecture of Selenium Webdriver

## Table of content

- Selenium
- Selenium Architecture
- Video of Architecture of Selenium
- Things to remember with Selenium Webdriver
- Same Origin Policy
- Difference between Selenium 2.0 vs. Selenium 3.0
- When to automate?
- Kind of tests you can automate with selenium
- Is Selenium Future of Automation?
- Certification for Selenium Webdriver
- Mistakes we make while learning Selenium
- Advantages of Selenium Webdriver
- Dis-Advantages of Selenium Webdriver

## Selenium

*Selenium is a open-source web based automation tool. Selenium enables user to automate browsers*

Selenium supports only web automation, and it doesn't support automation of desktop applications

Why did they Name as Selenium? A couple of years back, we had only one automation giant, which is QTP ( currently known as UFT ). A company called Mercury owned QTP, they were charging hefty for the QTP product.

After years a program was developed by Jason Huggins, a programmer at ThoughtWorks (still we can find a couple of thought work packages in selenium webdriver if you have noticed), they wanted to name the program as something which gives opposite meaning to QTP, later they gave name Selenium.

Selenium is a Chemical mineral that cures the wounds caused by Mercury.

## Selenium Architecture

Let's understand what the components present in the selenium webdriver architecture are.

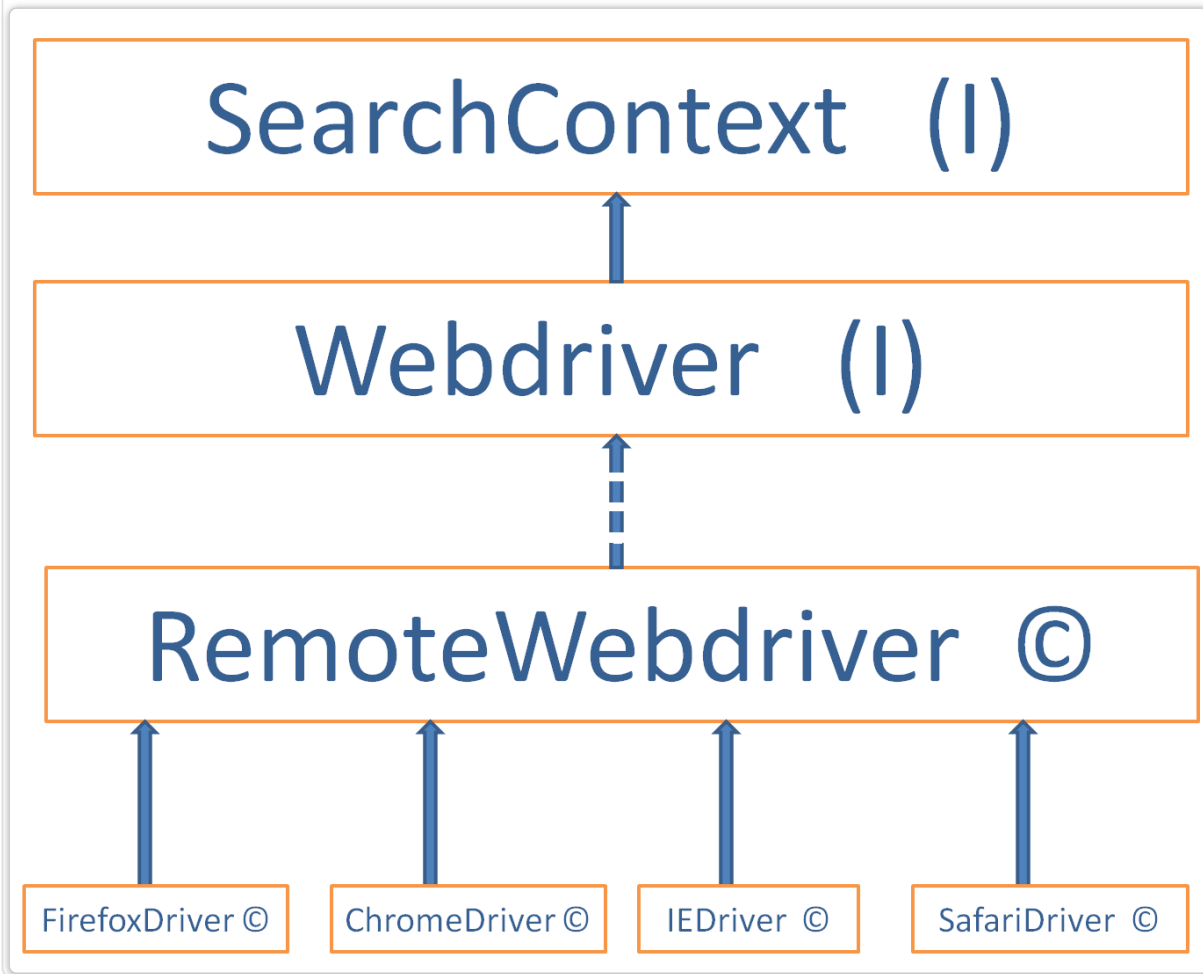
Search Context is the topmost interface in webdriver, Webdriver Interface extends the Search Context, Remote Webdriver class implements the Webdriver Interface, Browser classes (FirefoxDriver, ChromeDriver..) extends the Remote Webdriver Class.

Dotted Lines represents a class implementing an interface

Solid Lines represents class implementing class or Interface implementing the interface

(I)- Interface

(c)- class



Video of Architecture of Selenium

Subscribe to my youtube channel :

Things to remember with Selenium Webdriver

*Selenium WebDriver imitates the real world user behavior as closely as possible. Since real world user is not able to interact with any invisible element or elements not rendered yet, neither does WebDriver.*

So we need to always make sure that the web element is in the right status before we try to interact with it.

## Same Origin Policy

Same Origin Policy, also called Single Host Origin Policy, is a security measure used in Web browsers programming languages such as JavaScript and Ajax to protect the confidentiality and integrity of Webpage information.

Same Origin Policy prevents a web site's scripts from accessing and interacting with scripts used on other sites.

Same Origin Policy With Selenium Core: Selenium Core is nothing but Javascript commands, It would set itself up as a server (locally), and open a browser pointed to the Selenium server running locally. So the browser is now talking to the Selenium Server running locally.

This is a problem though because the browser is getting a script from Selenium(<http://localhost:9000/selenium>), which tells it what it wants to fetch resources from <https://websitetotest.com>. The browser says, "hey, this script came from localhost, and now it's requesting a resource from some outside website. This violated the same-origin-policy.

Same Origin Policy With Selenium RC : Selenium RC tricked the browser by proxy, and the browser considered that both webpage and server are from the same host, so the browser lets the Selenium RC to run.

Same Origin Policy With Selenium Webdriver : Webdriver provides native OS-level support in controlling the browser automation, so the browser feels like there is only a webpage. (with RC and Core it was Webpage and server)

Human Way of Same-origin Policy :

Selenium Core: Consider you are the webpage, your manager/wife could be the server which tells you to do this, do that. At some point, your body realizes that no need to follow them, as your mind does not generate these, this was the problem with Selenium Core.

RC example : Someone hypnotized you, and they are just asking you do to some job, and you are doing them this is selenium RC, hypnotization is nothing but the proxy.

Webdriver : You have been told what to do when something occurs from your childhood. Now you will do the same thing even after becoming an adult. But you will never realize that you were doing something that someone taught you long back. The only

thing that you feel is, this the way of doing.

*Java*

*C#*

*Ruby*

*Python*

*Javascript*

## Supported Browsers:

*Mozilla Firefox*

*Google Chrome*

*Opera*

*Microsoft Edge*

*Safari*

*Opera*

*etc...*

## Difference between Selenium 2.0 vs. Selenium 3.0

The most significant difference between selenium 3.0 and selenium 2.x is, In Selenium 3, all the major browser vendors are providing their own WebDriver implementations (Apple, Google, Microsoft, and Mozilla) instead of being developed by Selenium project (Selenium 2.x) and because they know their browsers better than anyone else, their WebDriver implementations can be tightly coupled to the browser, leading to a better testing experience for you.

Unfortunately, most of the [methods in Actions class](#) are not supported or being implemented now. Compared to Selenium 3 version Selenium 2 provides better operations

Most Unfortunate thing is, all the browsers are updated now, so selenium 2.x is not supported by them which forces users are forced to use Selenium 3.0 version

Mozilla has made changes to Firefox that mean that from Firefox 48 user must use their geckodriver to use firefox browser, regardless of whether you're using Selenium 2 or 3.

Selenium 3 is more mobile automation focused, maybe in future Selenium replaces Appium or Appium replaces Selenium.

HtmlUnitDriver is now packaged separately, means for headless testing we need to add the htmlunitdriver

Selenium (3.8.0) doesn't support for PhantomJS, and it's recommended to use **headless Firefox or Chrome instead**

## When to automate?

Below are a few scenarios where we go for automation:

*Frequent regression testing, whenever a new feature introduced to the application we perform functional testing at the same time we also required to perform regression testing to ensure that new didn't affect the existing functionality negatively*

*Repeated test case Execution is required, we may need to test a few functionalities again and again for a set of data like login details*

*User Acceptance Tests, we write our code on a test environment, but we actually have to run the same code on UAT as well to check for bugs quickly*

*Faster Feedback to the developers, when developers deployed something tester must be able to give the details of the deployment effect on basic functionalities of the application*

*Reduce the Human Effort, When we test a functionality manually it takes a lot of time but automating reduces the effort*

*Test the same application on multiple environments, like test env and Live env.*

### Custom Locators in Selenium

## Kind of tests you can automate with selenium

The type of testing selenium supports are basically Functional Testing and Regression Testing of Web Applications  
*Functionality Testing which includes Smoke Testing, Regression Testing, and Sanity Testing*  
*Database Testing using drivers like JDBC can also be performed as it is a part of Functional Testing*  
*Browser Compatibility Testing (Cross-browser and Different versions of the same browser)*  
*UI Testing*  
*API Testing*

## Is Selenium Future of Automation?

I believe that instead of looking into a Selenium automation career, you should aim at something more significant.

Selenium automation is basically programming with an automation library, so instead of becoming a specialist in a library, why not do that for the language?

In selenium terms, Apache POI is a library, if you become very expert in that apache poi, do you think, you know everything, NO.

Similarly, Selenium is also a Library/Framework in the language(selenium framework is there in different languages).

Focusing on only selenium may help in shorter-term (6-8 years), but for the long term, benefits learn the language and learning a library will not be difficult if you know the language

What is the name of someone who specializes in using a programming language? Ah, yes, a programmer.

So why not aiming at being a programmer since then you have a lot of options?

You could do test automation but not only with Selenium WebDriver but any other automation library or tool.

Why bet your career on using one library? No one can tell if Selenium will be here in 5 or 10 years because no one has the ability to look into the future.

Maybe Selenium Project Manager(selenium's official dev team) will decide at some point that he does not want to manage the Selenium project and then the project's quality goes way down.

With low quality, fewer and fewer companies will use it, and not many will need Selenium automation experts.

Many companies will not have in the future dedicated staff for test automation; Developers will do some of it, dev-ops will have a slice of the automation pie as well, maybe some testers will join in.

What a good programmer combines the selenium with some open-source Robotic Process automation tool, then every company will remove the test programmer as they anybody can drag and drop elements in RPA.

I would suggest you not only to learn selenium but also put more focus on language and analytical skills for solving the problem.

## Certification for Selenium Webdriver

1. Selenium does not offer any certification.
2. Selenium was introduced for free; they support it for free.
3. Some folks in the technology field who have studied Selenium are selling it out, but which is not considered as a certification.
4. If you really want to do the certification, then do the certification on the Programming Language, which you use for selenium.

Handle dropdowns in selenium

## Mistakes we make while learning Selenium

1. Becoming a Selenium automation expert in 4 months, I took almost 6 Years to become such a level and still learning. So stop becoming expert in 4 months, because you cannot do so.
2. Learning selenium for certification, because there is no such thing called certifications for selenium.

3. Starting selenium scripting without even learning the Programming language, learn programming language first
4. Believing a learning website as ideal for learning everything about selenium, I own this site, but I still suggest my reader not to stick to this site as this site is a replica of my views or learning on selenium.
5. Believing Selenium IDE is basic to Selenium Webdriver, Selenium IDE aka SIDE has no influence on Selenium Webdriver.
6. Not asking for help, I have seen people who are scared that If I ask this question, then my senior or SME may identify me as a guy do not have knowledge. Know this, knowledge is nothing but knowing the subject, but if you do not read or ask anyone, how you will know anything. Ask the expert/experienced guy.
7. Only preparing for interviews, when you don't know the subject you cannot survive in a company even though you cracked an interview

Capture Screenshot in selenium

## Advantages of Selenium Webdriver

Selenium webdriver is pure open source, freeware, and portable tool

Selenium webdriver supports many operating systems like Windows, Macintosh, Linux, Unix, etc. (platform independent)

Selenium can be integrated with continuous integration tools like Jenkins or Bamboo

Selenium Webdriver support third-party tools like AutoIt, Sikuli, Apache POI, with the help of the language

It supports all popular web browsers

It supports parallel test execution

It uses fewer Hardware resources as compare to other popular tools like ...QTP/UFT

It can be utilized for regression, user interface, functional and UAT testing.

Supports to **take a screenshot of execution**

## Dis-Advantages of Selenium Webdriver

*Since it is an open-source tool, No reliable Technical support (Official Users Group, Chat room in seleniumhq.org)*

*It doesn't support Desktop Applications/Windows bases applications*

*No Other tool integration for Test Management*

*Difficult to use as compare to other tools like QTP/UFT*

*Image verification is not possible*

*Selenium partiality for Dialog boxes.*

*Selenium doesn't support built-in add-ins assistance.*

*Additional Tools Required for Generating Reports.*

*New features may not work properly(with new browser driver executable like the gecko, chrome functionalities like dragging mouse movement).*

*Good tests require good coders. Yeah, I'm sure there are plenty of good developers automating web testing through Selenium. But for everyone's benefit, it's optimal to have your best developers write product features and leave automation testing to lesser skilled people.*

*Selenium test suites take too long to run. Write enough Selenium tests and your suite will take days to finish. No framework out there can make a browser run faster, and Selenium has pretty cool support for the grid that can run parallel tests on multiple computers. But very few people have actually set it up, and again, it requires quite a bit of skill to keep that beast running.*

[Learn about Parameterized Constructor in Selenium and Java](#)

## Recommended Readings

[Install Eclipse for Selenium Webdriver](#)

[Install Selenium jars with Eclipse](#)

[Open Browser in Selenium Webdriver | Webdriver driver = new FirefoxDriver\(\)](#)

[Close and Quit Browser in Selenium Webdriver](#)

[HashMap in Java](#)

[Arraylist in Selenium](#)

[Collections in Selenium](#)

[Wrapper Classes in Java](#)