

What Tools Does Selenium Testing Include and What Do They Do?

TechRounder PDF Edition

Live article:

<https://www.techrounder.com/development/what-tools-does-selenium-testing-include-and-what-do-they-do/>

By Vipin PG | Published April 6, 2022 | Updated March 8, 2026 | Format: Article | 4 min read

In brief

Selenium is an open-source automated testing framework used to validate different web applications in various browsers and platforms. Disparate selenium tools are used to automate the tests across browser applications.

Selenium is an open-source automated testing framework used to validate different web applications in various browsers and platforms. Disparate selenium tools are used to automate the tests across browser applications. The testing process is carried out to validate the responsiveness of these applications.

To avoid operating bugs in the system, having three levels of automated testing enables the specialists to test each level separately and prevent the system. The unit tests examine each function; the API tests check whether the APIs work following the pre-set plans or not, and the Selenium tests check the behavior of real users. Integrating Selenium testing into your process will finalize the testing process and support operating systems.

Programming languages like Java, C#, Python, and Ruby are used for scripting tests. In addition, other tests are performed to communicate with Selenium via the Selenium Client API. In some cases, companies use selenium alternative applications also for testing. However, here we are listing different tools used for selenium testing and how they help the testers.

Selenium IDE

The Selenium Integrated Development Environment is a tool used by the test case developer to develop selenium test cases. It's designed to be used by even new developers to generate automated test cases for their web applications. You do not require any particular setup to start the selenium IDE. Just ensure that you have added the extension to the specific browser.

The Selenium IDE provides an easy-to-use GUI for recording your interactions with the website. You can edit your test cases and test suites later. In addition, you can convert different test cases into various programming languages with the development environment.

Several features of the selenium IDE are under the toolbar used to control the test case execution. Speed Control helps control the speed of the test cases, while Run All allows the performance of the entire Test Suite. The Run feature runs the currently selected test. Finally, the Resume lets the developer pause and resume a particular test case.

The benefits of the Selenium IDE include allowing the developer to automatically record the test cases based on the interactions with the browser. Then, as the developer, you can set breakpoints to debug a particular test case. There is also great flexibility in executing the rest of the test cases.

Selenium WebDriver

The Selenium WebDriver permits you to execute cross-browser tests. It automates web-based application testing to validate that it performs as expected. This tool allows you to choose the programming language to create the test scripts. Tools like Sikuli and Autolt can overcome the handling of the window components during selenium testing.

Selenium WebDriver generally consists of four layers: Selenium Client library, JSON wire protocol, browsers, and Drivers. In addition, the Selenium Client library or the language bindings supports multiple libraries like Python, Java, and Ruby.

These bindings are developed to ensure compatibility for the various languages used in the testing process. All the supported language bindings can be downloaded from the official site of Selenium.

JSON (JavaScript Object Notation) wire protocol provides a transport mechanism to transfer data between the client and the server that's on the web. It supports various data structures like arrays and objects, doing reading and writing data from the JavaScript Object Notation easier. It serves as the Representational State Transfer API that supports exchanging information between HTTP servers.

Each driver is provided with a specific browser without revealing the internal operations of the browser working. That's how the browser servers interact with the particular browser through a secure connection. Browser drivers are specific to the languages used for the test case automation.

When dealing with the browser driver, you can download one of your choices per the language you are using on the official suite. The Selenium WebDriver supports multiple operating systems and supports modern browsers. It completes the execution of the test scripts faster when compared to the other tools.

Selenium GRID

Selenium GRID is a tool that makes it easy to run the tests on multiple machines parallel. You must route the commands to a remote web browser with one server acting as the Hub.

The Hub then routes the test commands in the JavaScript Object Notation to the multiple registered Grid nodes. A process allows the simultaneous execution of the tests on various machines while managing different browsers centrally. Selenium GRID supports single tests on multiple machines and browsers, making cross-browser testing easy.

The Selenium GRID consists of two main elements - Hub and Node. The Hub is the server that receives and accepts requests from WebDriver clients. The instructions received are executed remotely on the various parallel nodes.

A node refers to a remote device consisting of a native OS and a remote WebDriver. It executes all requests from the Hub in the form of JavaScript Object Notation using the WebDriver.

Selenium GRID helps run tests on many browsers and their different versions, operating systems, and devices. It's also used to reduce the time the test suite takes for complete execution. Consequently, it improves the overall turnaround time of the test results and has easy maintenance due to the virtual infrastructure.

The procedure for getting started with Selenium GRID browser testing is effortless. First, download the Selenium Server Standalone package and install it. The jar file contains Hub, WebDriver, and the legacy Remote Controller needed to run the GRID.

Then start the Hub with the appropriate commands. Start the Nodes by opening the command prompt and navigating to the Selenium Server Standalone jar file's directory. Configure the nodes and then start running the tests.

References

1. functionize.com - selenium-testing - <https://www.functionize.com/selenium-testing>
2. testrigor.com - alternative / selenium - <https://testrigor.com/alternative/selenium/>
3. blog.axway.com - amplify-products / api-management - <https://blog.axway.com/amplify-products/api-management/what-is-rest-api>
4. eecs.oregonstate.edu - ecampus-video / CS290 - <https://eecs.oregonstate.edu/ecampus-video/CS290/core-content/javascript-basics/javascript-objects.html>