WIKIPEDIA

Selenium (software)

Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including C#, Groovy, Java, Perl, PHP, Python, Ruby and Scala. The tests can then run against most modern web browsers. Selenium runs on Windows, Linux, and macOS. It is open-source software released under the Apache License 2.0.

Selenium

Stable release	3.141.59 / November 19, 2018[1]
Repository	github.com/SeleniumHQ/selenium (https://github.com/SeleniumHQ/selenium)
Written in	Java
Operating system	Cross-platform
Type	Software testing framework for web applications
License	Apache License 2.0
Website	selenium.dev/ (https://selenium.dev/)

Contents

History

Selenium was originally developed by Jason Huggins in 2004 as an internal tool at ThoughtWorks. Huggins was later joined by other programmers and testers at ThoughtWorks, before Paul Hammant joined the team and steered the development of the second mode of operation that would later become "Selenium Remote Control" (RC). The tool was open sourced that year.

In 2005 Dan Fabulich and Nelson Sproul (with help from Pat Lightbody) made an offer to accept a series of patches that would transform Selenium-RC into what it became best known for. In the same meeting, the steering of Selenium as a project would continue as a committee, with Huggins and Hammant being the ThoughtWorks representatives.

In 2007, Huggins joined Google. Together with others like Jennifer Bevan, he continued with the development and stabilization of Selenium RC. At the same time, Simon Stewart at ThoughtWorks developed a superior browser automation tool called WebDriver. In 2009, after a meeting between the developers at the Google Test Automation Conference, it was decided to merge the two projects, and call the new project Selenium WebDriver, or Selenium 2.0.[2]

In 2008, Philippe Hanrigou (then at ThoughtWorks) made "Selenium Grid", which provides a hub allowing the running of multiple Selenium tests concurrently on any number of local or remote systems, thus minimizing test execution time. Grid offered, as open source, a similar capability to the internal/private Google cloud for Selenium RC. Pat Lightbody had already made a private cloud for "HostedQA" which he went on to sell to Gomez, Inc.

The name Selenium comes from a joke made by Huggins in an email, mocking a competitor named Mercury, saying that you can cure mercury poisoning by taking selenium supplements. The others that received the email took the name and ran with it.[3]

Components

Selenium is composed of several components with each taking on a specific role in aiding the development of web application test automation.[4]

Selenium IDE

Selenium IDE is a complete integrated development environment (IDE) for Selenium tests. It is implemented as a Firefox Add-On and as a Chrome Extension. It allows for recording, editing and debugging of functional tests. It was previously known as Selenium Recorder. Selenium-IDE was originally created by Shinya Kasatani and donated to the Selenium project in 2006. Selenium IDE was previously little-maintained.[5] Selenium IDE began being actively maintained in 2018.[6][7][8][9]

Scripts may be automatically recorded and edited manually providing autocompletion support and the ability to move commands around quickly. Scripts are recorded in *Selenese*, a special test scripting language for Selenium. Selenese provides commands for performing actions in a browser (click a link, select an option) and for retrieving data from the resulting pages.

The 2.x version of the Selenium IDE for Firefox stopped working[10] after the Firefox 55 upgrade and has been replaced by Selenium IDE 3.x.[11]

In addition to the official Selenium IDE project, two alternative Selenium IDE browser extensions are actively maintained:[12] Kantu (Open-Source GPL license) and Katalon Recorder (Closed Source).

Selenium client API

As an alternative to writing tests in Selenese, tests can also be written in various programming languages. These tests then communicate with Selenium by calling methods in the Selenium Client API. Selenium currently provides client APIs for Java, C#, Ruby, JavaScript, R and Python.

With Selenium 2, a new Client API was introduced (with *WebDriver* as its central component). However, the old API (using class *Selenium*) is still supported.

Selenium WebDriver

Selenium WebDriver is the successor to Selenium RC. Selenium WebDriver accepts commands (sent in Selenese, or via a Client API) and sends them to a browser. This is implemented through a browser-specific browser driver, which sends commands to a browser and retrieves results. Most browser drivers actually launch and access a browser application (such as Firefox, Google Chrome, Internet Explorer, Safari, or Microsoft Edge); there is also an HtmlUnit browser driver, which simulates a browser using the headless browser HtmlUnit.

Unlike in Selenium 1, where the Selenium server was necessary to run tests, Selenium WebDriver does not need a special server to execute tests. Instead, the WebDriver directly starts a browser instance and controls it. However, Selenium Grid can be used with WebDriver to execute tests on remote systems (see below). Where possible, WebDriver uses native operating system level functionality rather than browser-based JavaScript commands to drive the browser. This bypasses problems with subtle differences between native and JavaScript commands, including security restrictions.[13]

In practice, this means that the Selenium 2.0 API has significantly fewer calls than does the Selenium 1.0 API. Where Selenium 1.0 attempted to provide a rich interface for many different browser operations, Selenium 2.0 aims to provide a basic set of building blocks from which developers can create their own Domain Specific Language. One such DSL already exists: the Watir project in the Ruby language has a rich history of good design. Watir-webdriver implements the Watir API as a wrapper for Selenium-Webdriver in Ruby. Watir-webdriver is created entirely automatically, based on the WebDriver specification and the HTML specification.

As of early 2012, Simon Stewart (inventor of WebDriver), who was then with Google and now with Facebook, and David Burns of Mozilla were negotiating with the W3C to make WebDriver an internet standard. In July 2012, the working draft was released and the recommendation followed in June 2018.[14] Selenium-Webdriver (Selenium 2.0) is fully implemented and supported in Python, Ruby, Java and C#.

Selenium Remote Control

Selenium Remote Control (RC) is a server, written in Java, that accepts commands for the browser via HTTP. RC makes it possible to write automated tests for a web application in any programming language, which allows for better integration of Selenium in existing unit test frameworks. To make writing tests easier, Selenium project currently provides client drivers for PHP, Python, Ruby, .NET, Perl and Java. The Java driver can also be used with JavaScript (via the Rhino engine). An instance of selenium RC server is needed to launch html test case - which means that the port should be different for each parallel run. However, for Java/PHP test case only one Selenium RC instance needs to be running continuously.[15]

Selenium Remote Control was a refactoring of Driven Selenium or Selenium B designed by Paul Hammant, credited with Jason as co-creator of Selenium. The original version directly launched a process for the browser in question, from the test language of Java, .Net, Python or Ruby. The wire protocol (called 'Selenese' in its day) was reimplemented in each language port. After the refactor by Dan Fabulich and Nelson Sproul (with help from Pat Lightbody) there was an intermediate daemon process between the driving test script and the browser. The benefits included the ability to drive remote browsers and the reduced need to port every line of code to an increasingly growing set of languages. *Selenium Remote Control* completely took over from the Driven Selenium code-line in 2006. The browser pattern for 'Driven'/'B' and 'RC' was response/request, which subsequently became known as Comet.

With the release of Selenium 2, Selenium RC has been officially deprecated in favor of Selenium WebDriver.

Selenium Grid

Selenium Grid is a server that allows tests to use web browser instances running on remote machines. With Selenium Grid, one server acts as the hub. Tests contact the hub to obtain access to browser instances. The hub has a list of servers that provide access to browser instances (WebDriver nodes), and lets tests use these instances. Selenium Grid allows running tests in parallel on multiple machines and to manage different browser versions and browser configurations centrally (instead of in each individual test).

The ability to run tests on remote browser instances is useful to spread the load of testing across several machines and to run tests in browsers running on different platforms or operating systems. The latter is particularly useful in cases where not all browsers to be used for testing can run on the same platform.

See also

- Acceptance testing
- Capybara (software)
- Given-When-Then
- List of web testing tools
- MediaWiki Selenium
- Regression testing
- Robot Framework

References

1. "Releases - SeleniumHQ/selenium" (https://github.com/SeleniumHQ/selenium/releases). Retrieved September 28, 2018 – via GitHub.
2. "The Selenium Project" (https://thenewcircle.com/static/bookshelf/selenium_tutorial/project.html). NewCircle. Retrieved June 29, 2014.
3. Krill, Paul (April 6, 2011). "Open source Selenium web app test suite to support iPhone and Android" (http://news.techworld.com/applications/3272444/open-source-selenium-web-app-test-suite-to-support-iphone-and-android/). *InfoWorld*. Retrieved May 9, 2012. "Selenium was so named because Huggins, dissatisfied with testing tools on the market, was seeking a name that would position the product as an alternative to Mercury Interactive QuickTest Professional commercial testing software. The name, Selenium, was selected because selenium mineral supplements serve as a cure for mercury poisoning, Huggins explained."
4. Moizuddin, Khaja (June 7, 2018). "Components of the Selenium Automation Tool" (https://dzone.com/articles/components-of-selenium-automation-tool). *dzone.com*. DevOps Zone. Retrieved July 10, 2019.
5. Evans, Jim. "Selenium Users - Selenium IDE seems dated and lacks features" (https://groups.google.com/d/msg/selenium-users/6eHTTtriMZk/_5DkomDtBwAJ). *groups.google.com*. Retrieved February 7, 2016.
6. "It's back! Selenium IDE Reborn with Dave Haeffner" (https://testingpodcast.com/230-its-back-selenium-ide-reborn-with-dave-haeffner/). *testingpodcast.com*.
7. "Selenium IDE Is Dead, Long Live Selenium IDE!" (https://seleniumhq.wordpress.com/2018/08/06/selenium-ide-tng/). *Selenium IDE Official Blog*. August 6, 2018. Retrieved November 27, 2019 – via seleniumhq.wordpress.com.
8. Colantonio, Joe (November 27, 2018). "Stunning return of Selenium IDE" (https://testguild.com/selenium-ide/). *testguild.com*. Retrieved November 27, 2019.
9. "List of new Selenium IDE features" (https://applitools.com/blog/why-selenium-ide-2019). *applitools.com*.
10. "Firefox 55 and Selenium IDE" (https://seleniumhq.wordpress.com/2017/08/09/firefox-55-and-selenium-ide/). *Official Selenium Blog*. August 9, 2017. Retrieved August 21, 2017.
11. "Selenium IDE Download Site" (https://www.seleniumhq.org/projects/ide/). *seleniumhq.org*. Retrieved May 10, 2018.

12. "Selenium IDE rises like a phoenix from the ashes" (https://a9t9.com/blog/selenium-ide-2018/). *Automation Technology Blog*. Retrieved April 8, 2019.
13. "The Architecture of Open Source Applications: Selenium WebDriver" (http://www.aosabook.org/en/selenium.html). *aosabook.org*. Retrieved February 7, 2016.
14. Smith, Michael. "WebDriver motors on to W3C Recommendation" (https://www.w3.org/blog/2018/06/webdriver-recommendation/). *W3C Blog*. Retrieved September 10, 2018.
15. "Selenium Remote-Control" (https://www.seleniumhq.org/projects/remote-control/). *seleniumhq.org*. Retrieved November 13, 2019.

External links

- Official website (https://selenium.dev/) ✎

Retrieved from "https://en.wikipedia.org/w/index.php?title=Selenium_(software)&oldid=961486701"

This page was last edited on 8 June 2020, at 19:19 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.