SELENIUM

**I. Review:**

- An umbrella project (a number of sub-projects that conform to a standard) for a range of tools and libraries that enable and support the automation of the web browsers.

- A portable framework for testing web applications.

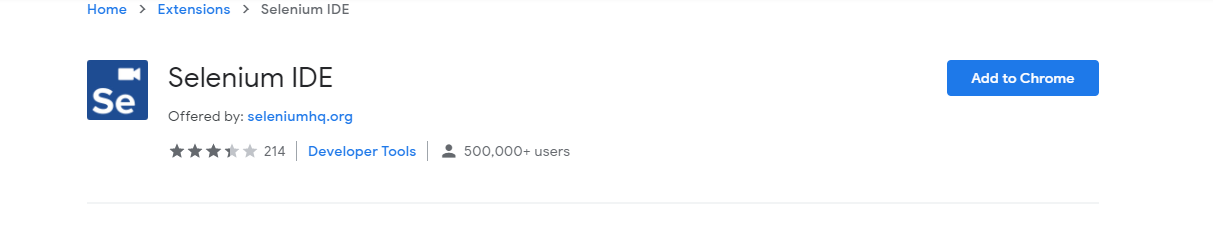
- It provides extensions to emulate user interaction with browsers.

- Automate web browser interaction (click, scroll, close tab, click check box,..).

(*Resource:* [*The Selenium Browser Automation Project :: Documentation for Selenium*](https://www.selenium.dev/documentation/en/)*)*

**II. Components:**

- Selenium Integrated Development Environment (IDE):



- Selenium Remote Control (RC) (Selenium 1): a system has a server that would act as an HTTP proxy to “trick” the browser into ingesting Selenium Core and that the tested web application came from the same domain name.

- Web Driver (Core of Selenium):

+ Communicating with the browser through a driver.

+ A compact object-oriented API.

+ Terminology:

* API(Application programming interface): The set of “commands” used to manipulate Web Driver (Rest API,...)
* Library: A code module which contains the APIs and the code necessary to implement them. Each language has specific binding, such as .jars file for Java, .dll files for .NET,…
* Driver: Controlling the actual browser. Drivers are generally executable modules that run on the system with the browser itself, not on the system executing the test suite (Although those may be the same system).
* Framework: An additional library used as a support for Web Driver suites (Junit, Nunit,..) and is responsible for running and executing Web Driver.

- Selenium Grid: allows the execution of Web Driver scripts on remote machines (virtual or real).

(*Resource:* [*Bài 1: Giới thiệu về Selenium (viblo.asia)*](https://viblo.asia/p/bai-1-gioi-thieu-ve-selenium-aWj538VwK6m)*,* [*The Selenium Browser Automation Project :: Documentation for Selenium*](https://www.selenium.dev/documentation/en/)*)*

**III. Tools and support:**

1. **Support multi-language:**

\*Standard:

- Usability: a programming language should be high-level and have the ability to manage all the memory management tasks.

- Simplicity: programming language syntax nomenclature (danh pháp) should be easy to understand.

- Test framework support: need to have to manage all the test cases and prioritize the testing flow.

- IDE (Integrated development environment) Support: to write and manage test cases.

- Industry Adoption: have a better community for support and updated packages and frameworks.

- Recommend: Python, Java, Ruby, C#, PHP,...

**2. Tools:**

**-** The surrounding infrastructure provided under the Selenium umbrella gives you the tools to put together your [grid of browsers](https://www.selenium.dev/documentation/en/grid/) so tests can be run on different browsers and multiple operating systems across a range of machines.

(*Resource:* [*Selecting a Programming Language for Selenium Automation Testing - DZone DevOps*](https://dzone.com/articles/how-to-select-programming-language-for-selenium-au) *).*

IV. Advantages and Disadvantages:

1. Advantages:
2. Supporting multi-languages: see III
3. Supporting multi-browser with Web Driver.
4. Write one time, run everywhere.
5. It is easy to understand and to implement.
6. It can take evidence when running test cases.
7. Disadvantages:
8. It is neccessary to handle time-out (depent on Internet).
9. It is neccessary to control test cases which might lead to overload of ram.

*(Resource:* [*Selenium là gì? Một số kinh nghiệm làm việc với Selenium | TopDev*](https://topdev.vn/blog/selenium-la-gi-mot-so-kinh-nghiem-lam-viec-voi-selenium/)*)*