Developing end-to-end tests with Selenium 4 and Java

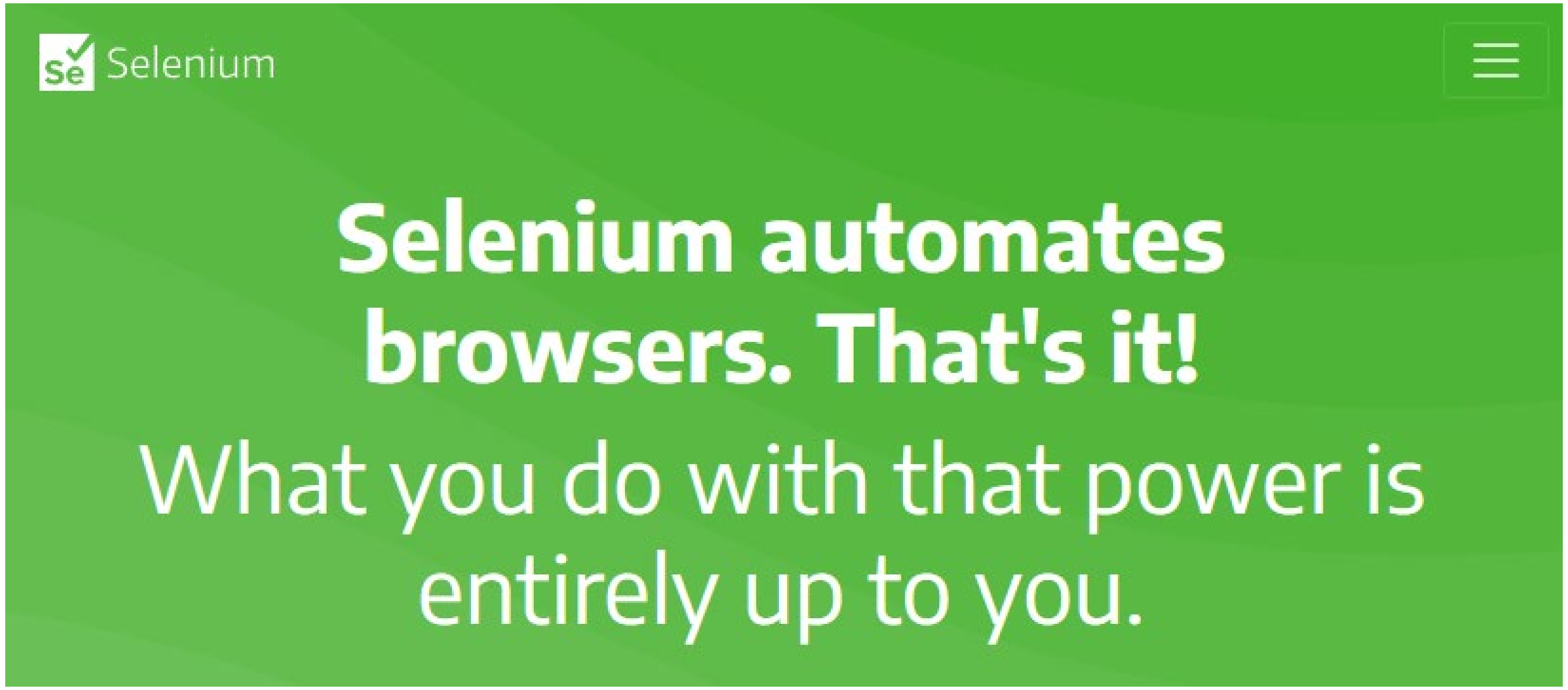
Testµ Conference 2022
August 24, 2022

Boni García

boni.garcia@uc3m.es https://bonigarcia.dev

@boni_gg https://github.com/bonigarcia

What is Selenium?

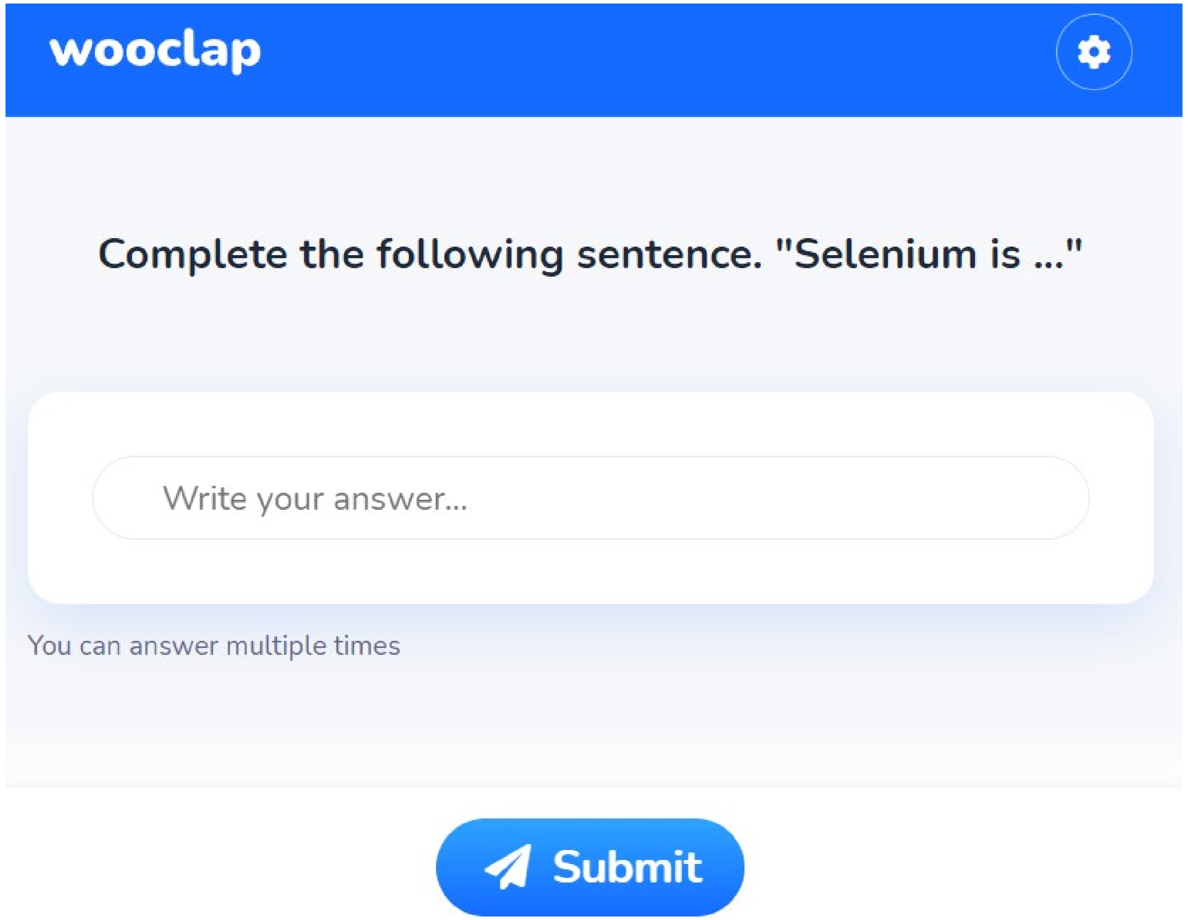

Selenium

Selenium automates browsers. That's it!

What you do with that power is entirely up to you.

https://www.selenium.dev/

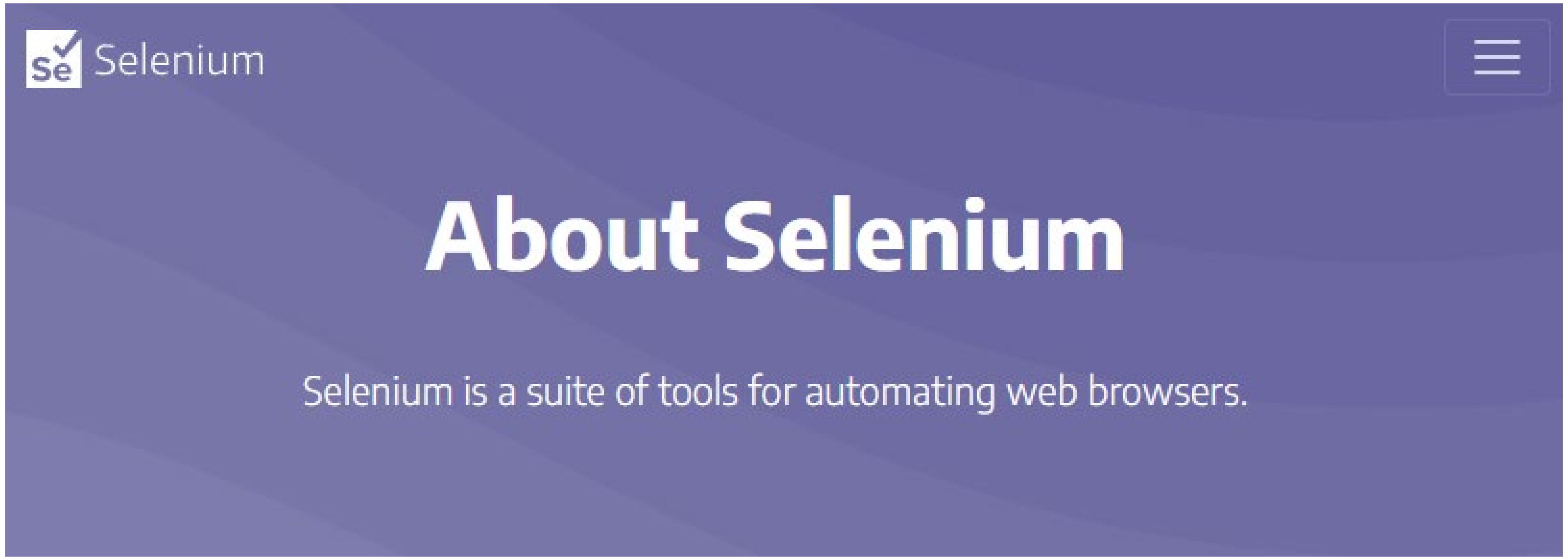
What is Selenium?



Complete the following sentence. "Selenium is ..."

https://app.wooclap.com/OJDKKR

What is Selenium?


Selenium

About Selenium

Selenium is a suite of tools for automating web browsers.

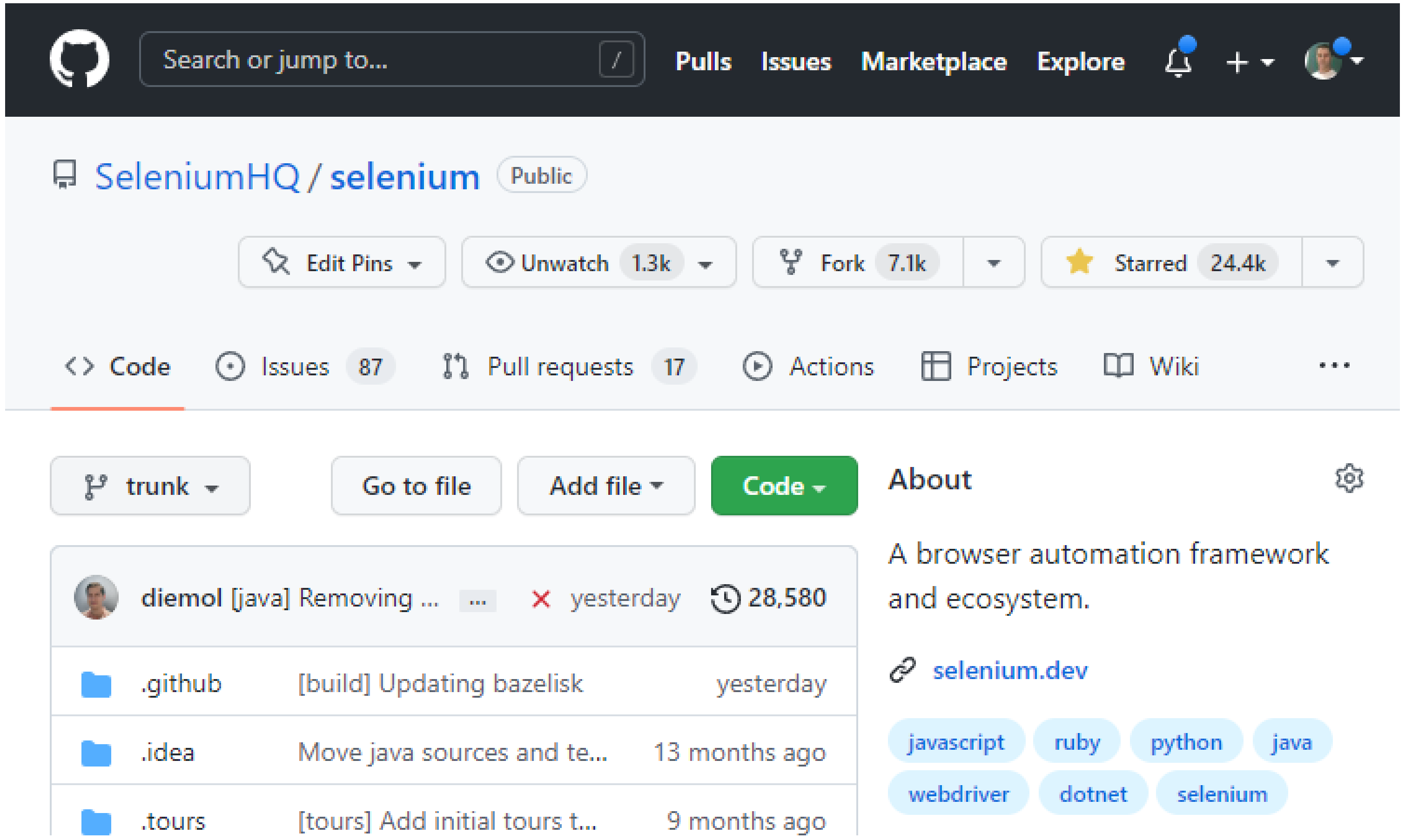
https://www.selenium.dev/about/

Selenium WebDriver

Selenium IDE

Selenium Grid

What is Selenium?



diemol [java] Removing ... ... yesterday 28,580

.github	[build] Updating bazelisk	yesterday
.idea	Move java sources and te...	13 months ago
.tours	[tours] Add initial tours t...	9 months ago

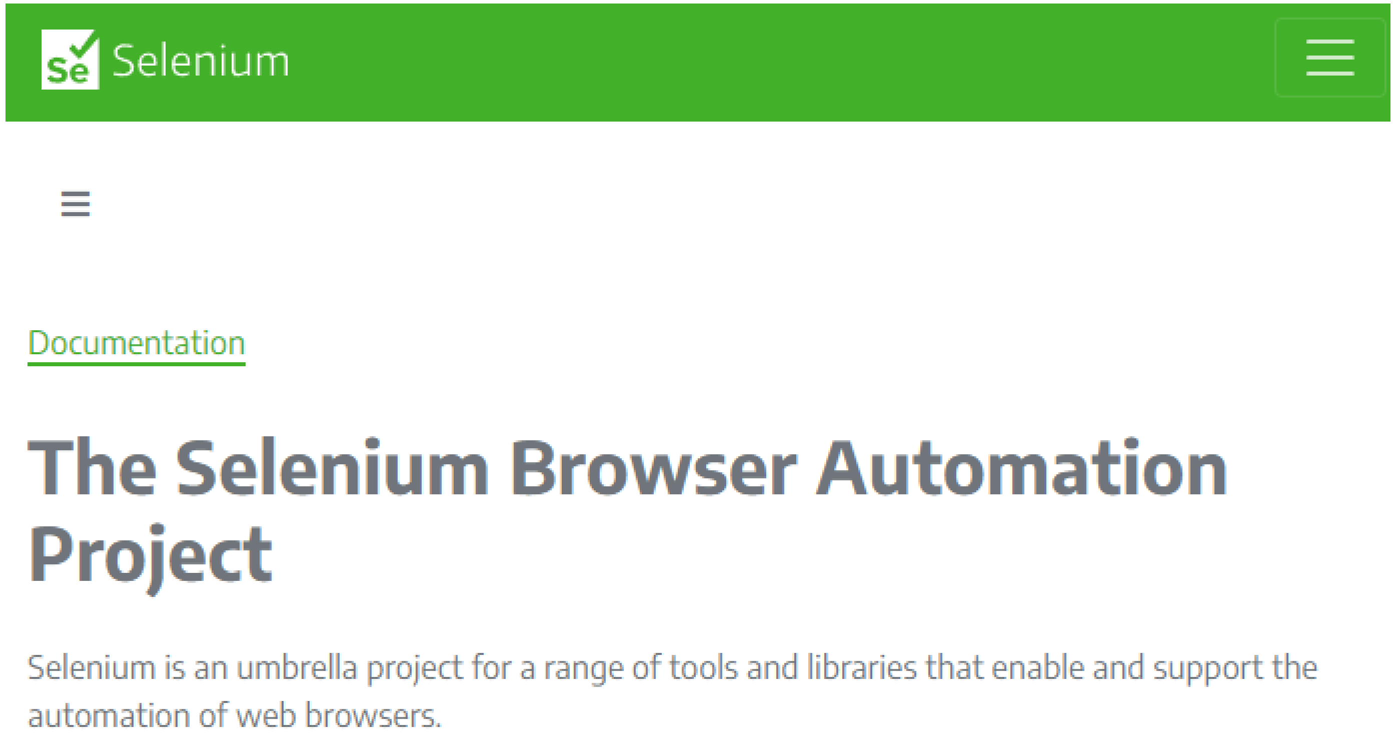
A browser automation framework and ecosystem.

selenium.dev

javascript ruby python java webdriver dotnet selenium

https://github.com/seleniumHQ/selenium/

What is Selenium?



The Selenium Browser Automation Project

Selenium is an umbrella project for a range of tools and libraries that enable and support the automation of web browsers.

https://www.selenium.dev/documentation/

What is Selenium?

- Selenium is an open source umbrella project that enables the automation of web browsers, and it is composed of three elements (or sub-projects):
  - Selenium WebDriver, a **library** for controlling browsers (e.g., Chrome, Firefox, Edge, Safari, or Opera) programmatically
  - Selenium IDE is a **tool** (concretely, a browser plugin) that implements the Record and Playback (R&P) automation technique
  - Selenium Grid a networked **infrastructure** that provides remote browsers accessible with the W3C WebDriver protocol

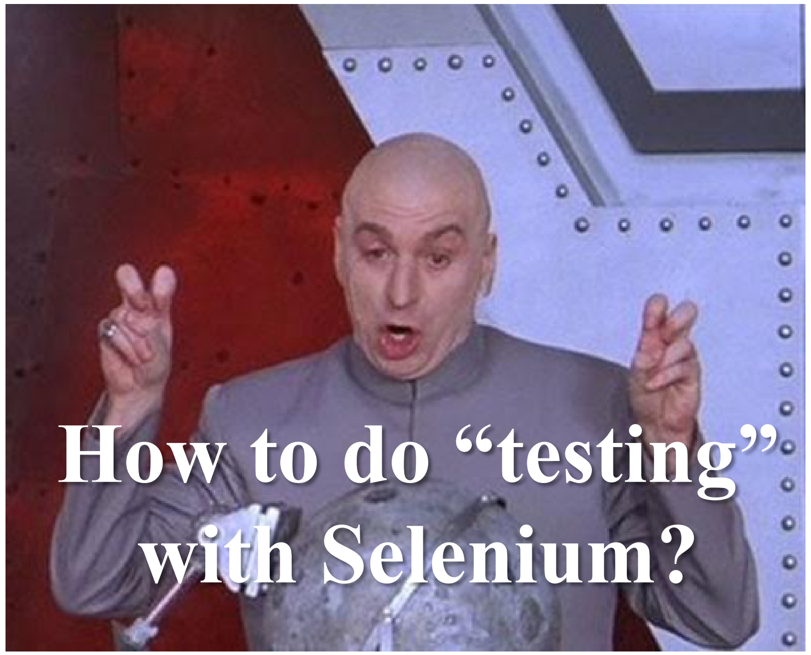
What is Selenium?

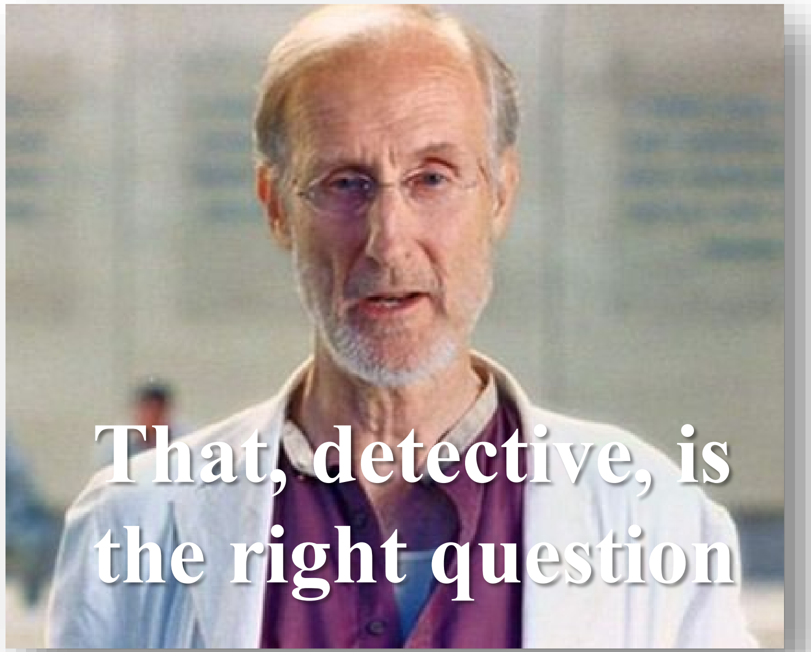
- Selenium WebDriver is the heart of the Selenium project and it is often known as simply Selenium

“ *Selenium is a browser automation library*

What is NOT Selenium?

- Selenium is NOT a testing framework ❌
- Selenium is NOT a testing library ❌





Testing with Selenium

“*Software **testing** (or simply testing) consists of the dynamic evaluation of a piece of software, called System Under Test (SUT), through a finite set of test cases (or simply tests), giving a verdict about it*

“*In **automated testing**, we use specific software tools to develop tests and control their execution against the SUT*

Testing with Selenium

“***End-to-end*** *(E2E) testing is a type of testing in which the SUT is evaluated as a whole through its User Interface (UI)*

Selenium (WebDriver) is key for **end-to-end automated testing for web applications**… but it is not the only ingredient we need

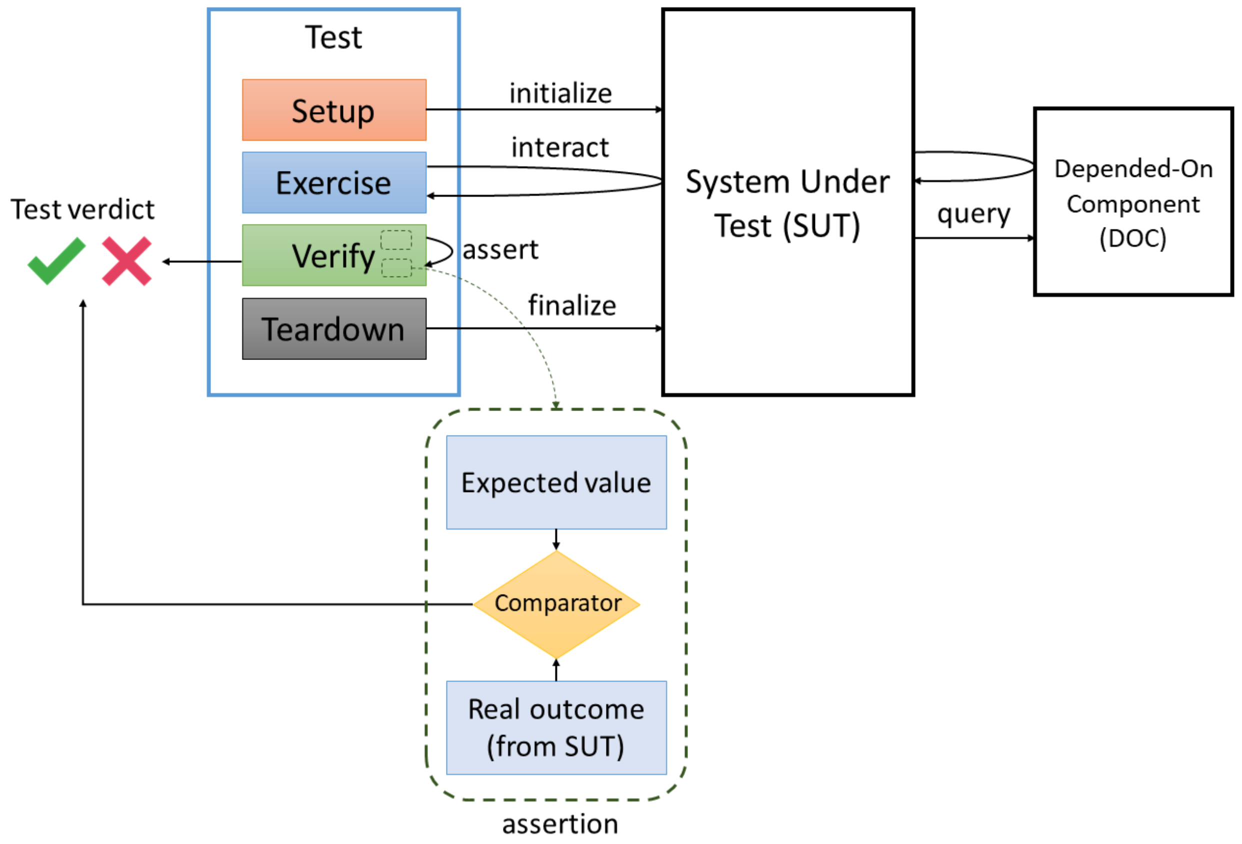
Testing with Selenium

- Selenium provides a cross-browser Application Programming Interface (API) in several programming languages

First we need to select the binding language

Testing with Selenium – Testing framework

- Since Selenium is not a testing library, we need an actual (unit) **testing framework**



JUnit

JUnit 5

TestNG

Testing with Selenium – Assertions library

- Testing frameworks (such as JUnit or TestNG) already provide specific classes for creating assertions
- In addition, there are specific **assertion libraries** which provide extra benefits:
  - Improve the test code readability by providing a rich set of fluent assertions
  - Provide enhanced error messages to help testers understand the cause of a failure

AssertJ Hamcrest Truth

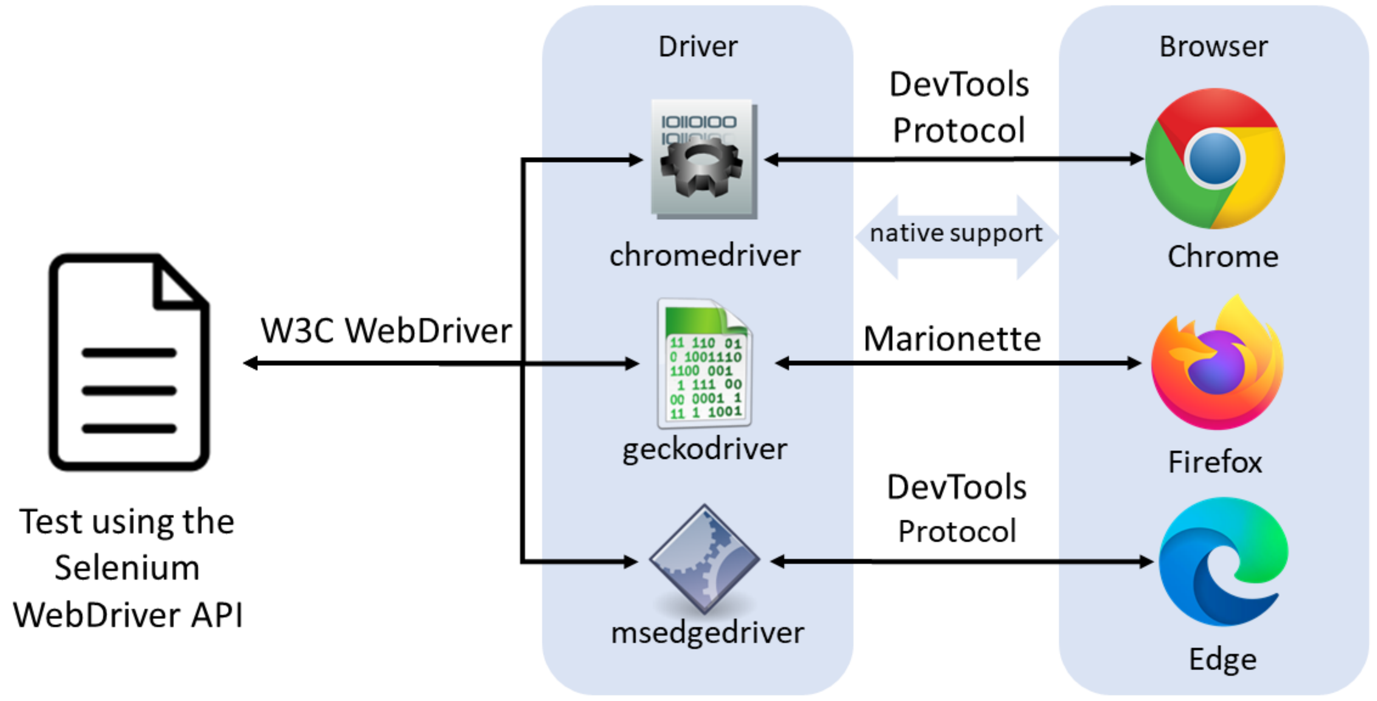
Testing with Selenium – Browsers

- Of course, we need one or more browsers to be driven with Selenium:

- There are different alternatives to provide these browsers:
  - Local
  - Remote
  - Cloud

Testing with Selenium – Drivers

- Selenium WebDriver uses the native support implemented by each browser to carry out the automation process
- For this reason, it is required a component called **driver** between the test using the Selenium WebDriver API and the browser



Testing with Selenium – Drivers

WebDriverManager

Automated driver management and other helper features for Selenium WebDriver in Java

https://bonigarcia.dev/webdrivermanager/

Selenium-Jupiter

JUnit 5 extension for Selenium WebDriver

https://bonigarcia.dev/selenium-jupiter/

Testing with Selenium – Build tools

- **Build tools** are utilities used to automate the creation of software applications from its source code
- These tools ease project management in terms of dependencies management, compilation, packaging, test execution, or deployment

Maven™

<APACHE ANT>

Bazel

Testing with Selenium – Build tools

```xml
<dependencies>
    <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>${selenium.version}</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>${junit5.version}</version>
        <scope>test</scope>
     </dependency>
    <dependency>
        <groupId>org.assertj</groupId>
        <artifactId>assertj-core</artifactId>
        <version>${assertj.version}</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>io.github.bonigarcia</groupId>
        <artifactId>webdrivermanager</artifactId>
        <version>${wdm.version}</version>
        <scope>test</scope>
    </dependency>
<dependencies>
```

```groovy
dependencies {
    testImplementation("org.seleniumhq.selenium:selenium-java:${seleniumVersion}")
    testImplementation("org.junit.jupiter:junit-jupiter:${junit5Version}")
    testImplementation("org.assertj:assertj-core:${assertjVersion}")
    testImplementation("io.github.bonigarcia:webdrivermanager:${wdmVersion}")
}
```

Maven™

Testing with Selenium – IDE

- An **IDE** (Integrated Development Environment) provide an excellent experience for development because they have a full-fledged environment (for coding, running, debugging, autocompletion, etc.)

eclipse

IntelliJ IDEA

Visual Studio

Apache NetBeans IDE

Testing with Selenium – Build server

- **Continuous Integration** (CI) is a software development practice where members of a software project build, test, and integrate their work continuously
- We need to use a server-side infrastructure called a **build server** to implement a CI pipeline

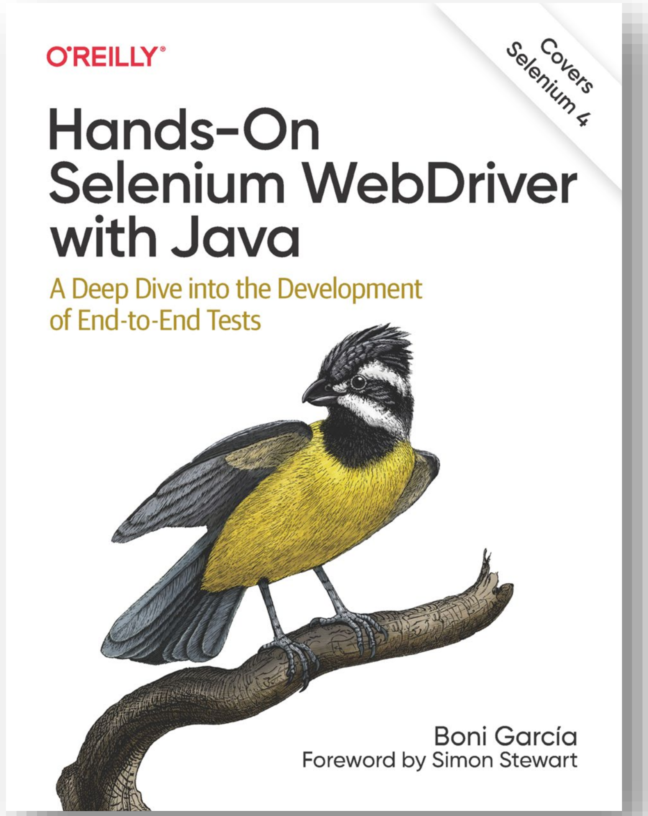
GitHub Actions

CI/CD

Jenkins

Bamboo

Testing with Selenium – Putting all together



O'REILLY

Hands-On Selenium WebDriver with Java

A Deep Dive into the Development of End-to-End Tests

Covers Selenium 4

Boni García

Foreword by Simon Stewart

https://github.com/bonigarcia/selenium-webdriver-java

Testing with Selenium – Putting all together

```java
class HelloWorldChromeJupiterTest {

    WebDriver driver;

    @BeforeAll
    static void setupClass() {
        WebDriverManager.chromedriver().setup();
    }

    @BeforeEach
    void setup() {
        driver = new ChromeDriver();
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    public void test() {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

```java
@ExtendWith(SeleniumJupiter.class)
class HelloWorldChromeSelJupTest {

    @Test
    void test(ChromeDriver driver) {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

Testing with Selenium – Advanced features

- Browser finder

```java
class HelloWorldSafariJupiterTest {

    WebDriver driver;

    @BeforeAll
    static void setupClass() {
        Optional<Path> browserPath = WebDriverManager.safaridriver()
                .getBrowserPath();
        assumeThat(browserPath).isPresent();
    }

    @BeforeEach
    void setupTest() {
        driver = new SafariDriver();
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void test() {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

```java
@EnabledIfBrowserAvailable(SAFARI)
@ExtendWith(SeleniumJupiter.class)
class HelloWorldSafariSelJupTest {

    @Test
    void test(SafariDriver driver) {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

Testing with Selenium – Advanced features

- Browsers in Docker

```java
class DockerChromeJupiterTest {

    WebDriver driver;

    WebDriverManager wdm = WebDriverManager.chromedriver().browserInDocker();

    @BeforeEach
    void setupTest() {
        assumeThat(isDockerAvailable()).isTrue();
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        wdm.quit();
    }

    @Test
    void testDockerChrome() {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

```java
@EnabledIfDockerAvailable
@ExtendWith(SeleniumJupiter.class)
class DockerChromeSelJupTest {

    @Test
    void testDockerChrome(@DockerBrowser(type = CHROME) WebDriver driver) {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

Testing with Selenium – Advanced features

- Browsers in Docker

```
class DockerChromeVncJupiterTest {

    WebDriver driver;

    WebDriverManager wdm = WebDriverManager.chromedriver().browserInDocker()
            .enableVnc();

    @BeforeEach
    void setupTest() {
        assumeThat(isDockerAvailable()).isTrue();
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        wdm.quit();
    }

    @Test
    void testDockerChromeVnc () throws Exception {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

```
@EnabledIfDockerAvailable
class DockerChromeVncSelJupTest {

    @RegisterExtension
    static SeleniumJupiter seleniumJupiter = new SeleniumJupiter();

    @Test
    void testDockerChromeVnc(
            @DockerBrowser(type = CHROME, vnc = true) WebDriver driver) {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

Testing with Selenium – Advanced features

- Browsers in Docker

```java
class DockerChromeBetaJupiterTest {

    WebDriver driver;

    WebDriverManager wdm = WebDriverManager.chromedriver().browserInDocker()
            .browserVersion("beta");

    @BeforeEach
    void setupTest() {
        assumeThat(isDockerAvailable()).isTrue();
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        wdm.quit();
    }

    @Test
    void testDockerChromeBeta() {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

```java
@EnabledIfDockerAvailable
@ExtendWith(SeleniumJupiter.class)
class DockerChromeBetaSelJupTest {

    @Test
    void testDockerChromeBeta(
            @DockerBrowser(type = CHROME, version = "beta") WebDriver driver) {
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/");
        assertThat(driver.getTitle()).contains("Selenium WebDriver");
    }

}
```

Testing with Selenium – Advanced features

- Monitoring
  - WebDriverManager provides seamless integration with BrowserWatcher

BrowserWatcher

“ *Browser extension for console monitoring, tab recording, Content Security Policy (CSP) disabling, and JavaScript/CSS injection*

https://bonigarcia.dev/browserwatcher/

Testing with Selenium – Advanced features

- Monitoring

```java
class GatherLogsFirefoxJupiterTest {

    WebDriverManager wdm = WebDriverManager.firefoxdriver().watch();
    WebDriver driver;

    @BeforeEach
    void setup() {
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void testGatherLogsFirefox() {
        driver.get(
                "https://bonigarcia.dev/selenium-webdriver-java/console-logs.html");
        List<Map<String, Object>> logMessages = wdm.getLogs();
        assertThat(logMessages).hasSize(5);
    }

}
```

```java
class GatherLogsFirefoxSelJupTest {

    @RegisterExtension
    static SeleniumJupiter seleniumJupiter = new SeleniumJupiter();

    @Test
    void testGatherLogsFirefox(@Watch FirefoxDriver driver) {
        driver.get(
                "https://bonigarcia.dev/selenium-webdriver-java/console-logs.html");
        List<Map<String, Object>> logMessages = seleniumJupiter.getLogs();
        assertThat(logMessages).hasSize(5);
    }

}
```

Testing with Selenium – Advanced features

- Monitoring

```java
class RecordEdgeJupiterTest {

    WebDriver driver;
    WebDriverManager wdm = WebDriverManager.edgedriver().watch();

    @BeforeEach
    void setup() {
        driver = wdm.create();
    }

    @AfterEach
    void teardown() {
        driver.quit();
    }

    @Test
    void test() throws InterruptedException {
        driver.get(
                "https://bonigarcia.dev/selenium-w

        wdm.startRecording(REC_FILENAME);

        // test logic

        wdm.stopRecording();
    }

}
```

```java
class RecordEdgeSelJupTest {


    @RegisterExtension
    static SeleniumJupiter seleniumJupiter = new SeleniumJupiter();

    @Test
    void testRecordEdge(@Watch EdgeDriver driver) throws InterruptedException {
        driver.get(
                "https://bonigarcia.dev/selenium-webdriver-java/slow-calculator.html");

        seleniumJupiter.startRecording(REC_FILENAME);

        // test logic

        seleniumJupiter.stopRecording();

    }

}
```

Takeaways

- Selenium WebDriver (often called simply Selenium) is a browser automation library, not a testing library
- To carry out end-to-end tests with Selenium, we should also use other testing frameworks and tools
- When using Java, we can use JUnit/TestNG (unit testing framework), Maven/Gradle (build tool), and AssertJ (assertions library), among others
- WebDriverManager and Selenium-Jupiter reduce the complexity of developing Selenium WebDriver tests by providing automated driver management and other features

Developing end-to-end tests with Selenium 4 and Java

Thank you very much!
Q&A

Boni García

boni.garcia@uc3m.es https://bonigarcia.dev

@boni_gg https://github.com/bonigarcia