Browser Automation with Java

JavaCro'25
Rovinj, Croatia
October 13, 2025

Boni García
https://bonigarcia.dev/

Browser Automation

“ ***Browser automation*** *is the process of using software or scripts to control a web browser and perform tasks automatically, without manual human intervention*

Browser Automation – Use cases

- Test automation
- Web scrapping
- Repetitive tasks for web pages

Browser Automation – Tools

Se Selenium

cypress

Puppeteer

Java

Playwright

WEBDRIVER I/O

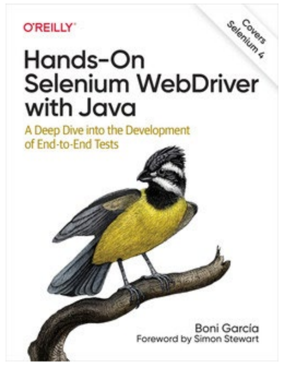
TestCafé®

About me

- Associate Professor at UC3M (Spain)
- Tech lead at the Selenium project
- Open-source maintainer
- Author, speaker

https://bonigarcia.dev/

Boni García
Mastering
Software Testing with JUnit 5
Comprehensive guide to develop high quality Java applications
Packt>


O'REILLY
Covers Selenium 4
Hands-On Selenium WebDriver with Java
A Deep Dive into the Development of End-to-End Tests
Boni García
Foreword by Simon Stewart

Selenium

What is Selenium?

“ *Selenium is a **browser automation library***

- Multilanguage
- Cross-browser
- Open-source and community-driven since 2004

https://selenium.dev/

Fork me on GitHub

Selenium Hello World

```java
public class HelloWorldSelenium {

    public static void main(String[] args) {
        // Open Chrome
        WebDriver driver = new ChromeDriver();

        // Navigate to web page
        String url = "https://bonigarcia.dev/selenium-webdriver-java/";
        driver.get(url);

        // Check page title
        String title = driver.getTitle();
        System.out.println(String.format("The title of %s is %s", url, title));

        // Close Chrome
        driver.quit();
    }

}
```

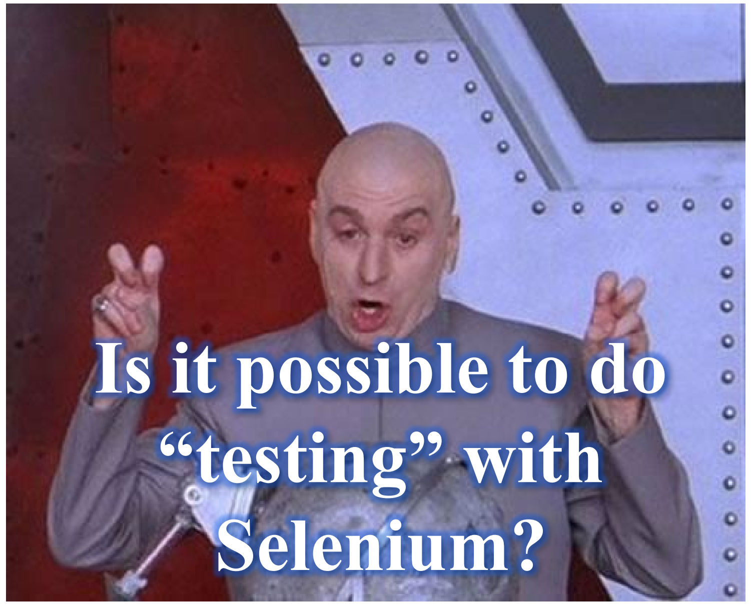
```xml
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.36.0</version>
</dependency>
```

```groovy
dependencies {
    implementation("org.seleniumhq.selenium:selenium-java:4.36.0")
}
```

What is NOT Selenium?

“ *Selenium is **not** a **testing framework***

- ✗ Test runner
- ✗ Assertions
- ✗ Reporting capabilities
- ✗ Integration with CI/CD
- ✗ Other testing features



Ecosystem

WebDriverManager

Selenide
CONCISE UI TESTS IN JAVA

Serenity BDD

Galen Framework

Selenium-Jupiter

Selenium

JUnit

TestNG

AssertJ

cucumber

REST-assured

mockito

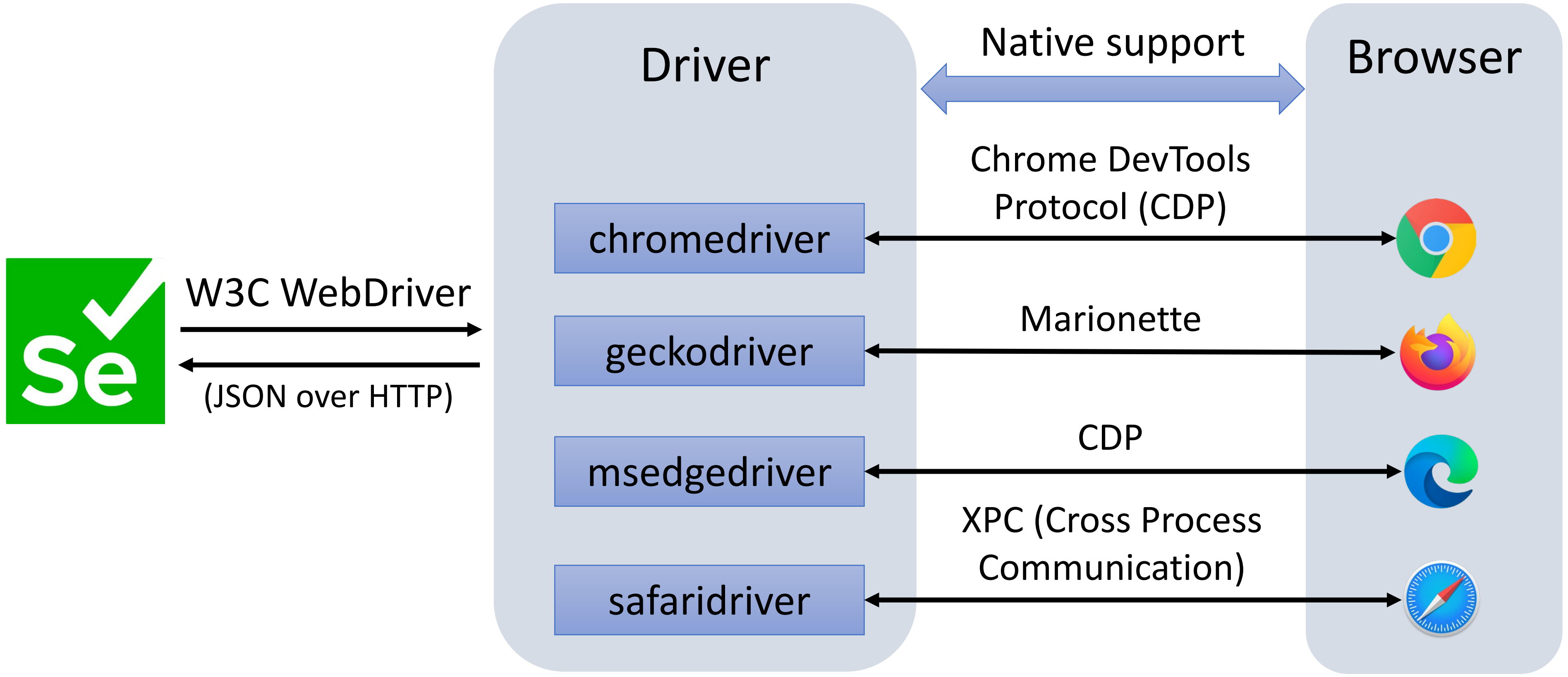
Datafaker

Allure Report

EXTENT REPORT

Java

Selenium Architecture



Automated Driver Management

WebDriverManager

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

https://bonigarcia.dev/webdrivermanager/

Automated Driver/Browser Management

Selenium Manager

“*Selenium Manager is the official driver manager of the Selenium project, and it is shipped out of the box with every Selenium release*

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

Fork me on GitHub

Selenium Hello World (E2E Test)

```java
class FirefoxBasicTest {

    WebDriver driver;

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

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

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

}
```

```xml
<dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.36.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <version>6.0.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.assertj</groupId>
    <artifactId>assertj-core</artifactId>
    <version>3.27.6</version>
    <scope>test</scope>
</dependency>
```

Automated Browser Management

- Selenium Manager automatically discovers, downloads, and caches the browsers driven with Selenium

	Test		
	✓	✓	✓ *
	✓	✓	✓
	✓	✓	✓

*Requires admin permissions

Fork me on GitHub

Automated Browser Management

```
class ChromeVersionTest {

    WebDriver driver;

    @BeforeEach
    void setup() {
        ChromeOptions options = new ChromeOptions();
        options.setBrowserVersion("beta");
        driver = new ChromeDriver(options);
    }

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

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

}
```

Specific browser versions (including `"beta"`, `"dev"`, or `"nightly"`) are supported

Fork me on GitHub

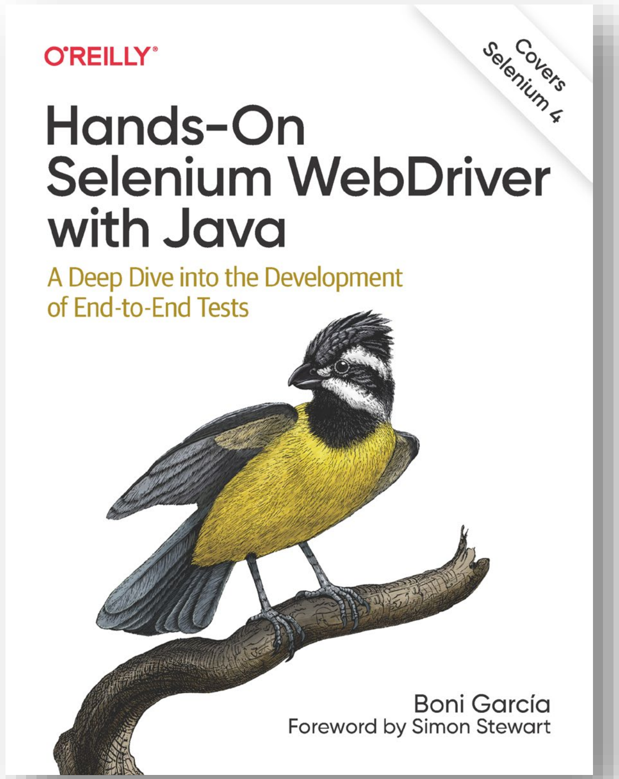
Selenium API

- Document Object Model (DOM) manipulation
- Impersonate user actions (keyboard, mouse)
- Waiting strategies
- Execute JavaScript
- Make screenshots
- Manage browser (e.g., headless, history, …)
- Chrome DevTools Protocol
- WebDriver BiDi
- …

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

https://github.com/bonigarcia/selenium-examples

https://github.com/bonigarcia/browser-automation-apis/


O'REILLY
Covers Selenium 4
Hands-On Selenium WebDriver with Java
A Deep Dive into the Development of End-to-End Tests
Boni García
Foreword by Simon Stewart

Selenium Locators

- **Locators** are used to identify and interact with web elements

Id | Name | Link text | Partial link text | Tag name | Class name | CSS selector | XPath

```
WebElement username = driver.findElement(By.id("username"));
```

```
WebElement textByName = driver.findElement(By.name("my-text"));
```

```
WebElement linkByText = driver.findElement(By.linkText("Return to index"));
```

```
WebElement linkByPartialText = driver.findElement(By.partialLinkText("index"));
```

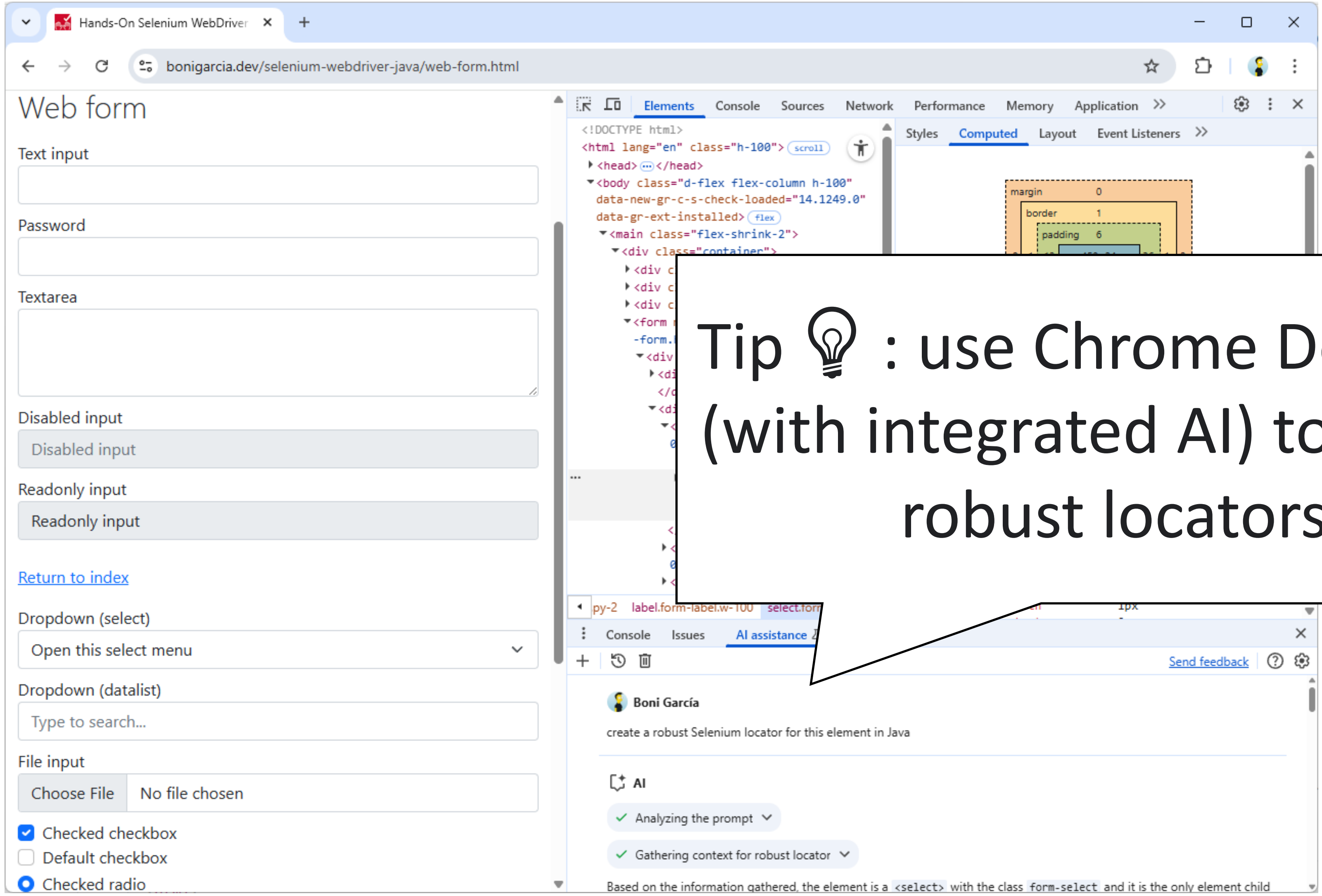
```
WebElement textarea = driver.findElement(By.tagName("textarea"));
```

```
WebElement alert = driver.findElement(By.className("alert"));
```

```
WebElement hidden = driver.findElement(By.cssSelector("input[type=hidden]"));
```

```
WebElement radio = driver.findElement(By.xpath("//*[@type='radio' and @checked]"));
```

Selenium Locators



Web form

Text input

Password

Textarea

Disabled input

Disabled input

Readonly input

Readonly input

Return to index

Dropdown (select)

Open this select menu

Dropdown (datalist)

Type to search...

File input

Choose File No file chosen

Checked checkbox

Default checkbox

Checked radio

```
<!DOCTYPE html>
<html lang="en" class="h-100"> scroll
  <head> ⋯ </head>
  <body class="d-flex flex-column h-100"
  data-new-gr-c-s-check-loaded="14.1249.0"
  data-gr-ext-installed> flex
    <main class="flex-shrink-2">
      <div class="container">
```

py-2 label.form-label.w-100 select.form

Boni García

create a robust Selenium locator for this element in Java

AI

✓ Analyzing the prompt

✓ Gathering context for robust locator

Based on the information gathered, the element is a `<select>` with the class `form-select` and it is the only element child

Tip 💡 : use Chrome DevTools (with integrated AI) to create robust locators

Selenium Waiting Strategies

- **Waiting strategies** are used to handle synchronization between the Selenium script and the actual response speed of the web element

Implicit Wait

Explicit Wait

Fluent Wait

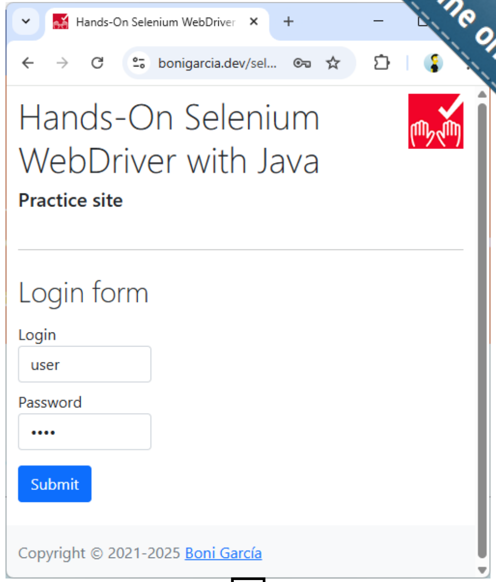
```
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
WebElement landscape = wait.until(ExpectedConditions
        .presenceOfElementLocated(By.id("landscape")));
```

```
Wait<WebDriver> wait = new FluentWait<>(driver)
        .withTimeout(Duration.ofSeconds(10))
        .pollingEvery(Duration.ofSeconds(1))
        .ignoring(NoSuchElementException.class);
WebElement landscape = wait.until(ExpectedConditions
        .presenceOfElementLocated(By.id("landscape")));
```

Selenium Waiting Strategies

```java
class LoginSeleniumTest {

    // Fixture

    @Test
    void test() throws Exception {
        // Open system under test (SUT)
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/login-form.html");

        // Log in
        driver.findElement(By.id("username")).sendKeys("user");
        driver.findElement(By.id("password")).sendKeys("user");
        driver.findElement(By.cssSelector("button[type='submit']")).click();

        // Assert expected text
        WebElement successElement = driver.findElement(By.id("success"));  // FIXME: flaky
        assertThat(successElement.getText()).contains("Login successful");

        // Take screenshot
        File screenshot = ((TakesScreenshot) driver).getScreenshotAs(FILE);
        Path destination = Paths.get("login-selenium.png");
        Files.move(screenshot.toPath(), destination, REPLACE_EXISTING);
    }

}
```



Hands-On Selenium WebDriver with Java

Practice site

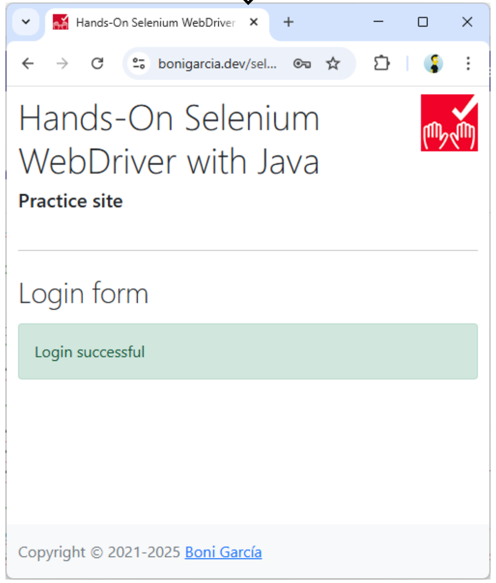
Login form

Login: user

Password: ····

Submit

Copyright © 2021-2025 Boni García



Hands-On Selenium WebDriver with Java

Practice site

Login form

Login successful

Copyright © 2021-2025 Boni García

Fork me on GitHub

Selenium Waiting Strategies

```java
class SlowLoginSeleniumTest {

    // Fixture

    @Test
    void test() throws Exception {
        // Open system under test (SUT)
        driver.get("https://bonigarcia.dev/selenium-webdriver-java/login-slow.html");

        // Log in
        driver.findElement(By.id("username")).sendKeys("user");
        driver.findElement(By.id("password")).sendKeys("user");
        driver.findElement(By.cssSelector("button[type='submit']")).click();

        // Assert expected text
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        WebElement successElement = wait.until(ExpectedConditions.presenceOfElementLocated(By.id("success")));
        assertThat(successElement.getText()).contains("Login successful");

        // Take screenshot
        File screenshot = ((TakesScreenshot) driver).getScreenshotAs(FILE);
        Path destination = Paths.get("slow-login-selenium.png");
        Files.move(screenshot.toPath(), destination, REPLACE_EXISTING);
    }

}
```

Explicit wait

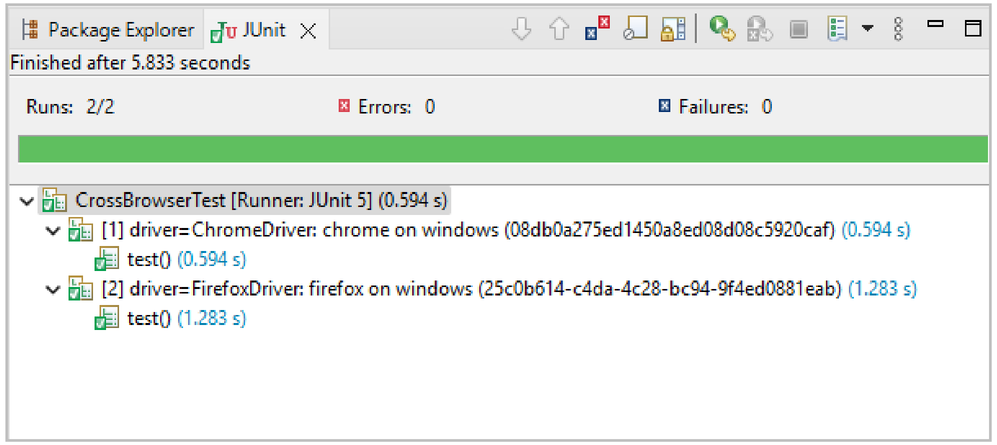
Fork me on GitHub

Ecosystem – Cross-Browser Testing

```java
class CrossBrowserTest extends CrossBrowserParent {

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

}
```

```java
public class CrossBrowserProvider implements ArgumentsProvider {

    @Override
    public Stream<? extends Arguments> provideArguments(
            ExtensionContext context) {
        ChromeDriver chrome = new ChromeDriver();
        FirefoxDriver firefox = new FirefoxDriver();

        return Stream.of(Arguments.of(chrome), Arguments.of(firefox));
    }

}
```

```java
@ParameterizedClass
@ArgumentsSource(CrossBrowserProvider.class)
class CrossBrowserParent {

    @Parameter
    WebDriver driver;

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

}
```



```
Finished after 5.833 seconds
Runs: 2/2        Errors: 0        Failures: 0

CrossBrowserTest [Runner: JUnit 5] (0.594 s)
  [1] driver=ChromeDriver: chrome on windows (08db0a275ed1450a8ed08d08c5920caf) (0.594 s)
      test() (0.594 s)
  [2] driver=FirefoxDriver: firefox on windows (25c0b614-c4da-4c28-bc94-9f4ed0881eab) (1.283 s)
      test() (1.283 s)
```

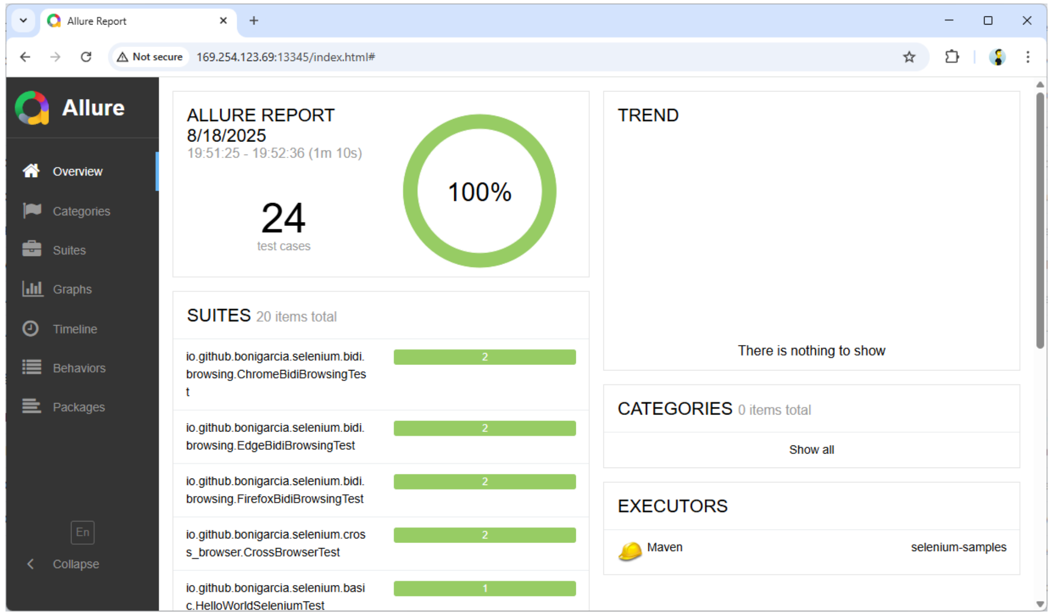
https://junit.org/

Fork me on GitHub

Ecosystem – Reporting

```
<dependencies>
   <dependency>
      <groupId>io.qameta.allure</groupId>
      <artifactId>allure-junit5</artifactId>
      <version>2.29.1</version>
      <scope>test</scope>
    </dependency>
</dependencies>
<build>
   <plugins>
      <plugin>
         <groupId>org.apache.maven.plugins</groupId>
         <artifactId>maven-surefire-plugin</artifactId>
         <version>3.5.3</version>
         <configuration>
            <properties>
               <property>
                  <name>listener</name>
                  <value>io.qameta.allure.junit5.AllureJunit5</value>
               </property>
            </properties>
          </configuration>
       </plugin>
       <plugin>
          <groupId>io.qameta.allure</groupId>
          <artifactId>allure-maven</artifactId>
          <version>2.15.2</version>
       </plugin>
    </plugins>
</build>
```

```
mvn test
mvn allure:report
mvn allure:serve
```



Allure Report

Not secure 169.254.123.69:13345/index.html#

Allure

Overview
Categories
Suites
Graphs
Timeline
Behaviors
Packages
Collapse

ALLURE REPORT 8/18/2025
19:51:25 - 19:52:36 (1m 10s)

24 test cases

100%

TREND

There is nothing to show

SUITES 20 items total

io.github.bonigarcia.selenium.bidi.browsing.ChromeBidiBrowsingTest	2
io.github.bonigarcia.selenium.bidi.browsing.EdgeBidiBrowsingTest	2
io.github.bonigarcia.selenium.bidi.browsing.FirefoxBidiBrowsingTest	2
io.github.bonigarcia.selenium.cross_browser.CrossBrowserTest	2
io.github.bonigarcia.selenium.basic.HelloWorldSeleniumTest	1

CATEGORIES 0 items total

Show all

EXECUTORS

Maven selenium-samples

Allure Report

https://allurereport.org/

Fork me on GitHub

Ecosystem – Reporting

```
class ReportingJupiterTest {

    WebDriver driver;
    static ExtentReports reports;

    @BeforeAll
    static void setupClass() {
        reports = new ExtentReports();
        ExtentSparkReporter htmlReporter = new ExtentSparkReporter("extentReport.html");
        reports.attachReporter(htmlReporter);
    }

    @BeforeEach
    void setup(TestInfo testInfo) {
        reports.createTest(testInfo.getDisplayName());
        driver = new ChromeDriver();
    }

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

    @AfterAll
    static void teardownClass() {
        reports.flush();
    }

    // Tests

}
```

EXTENT REPORT

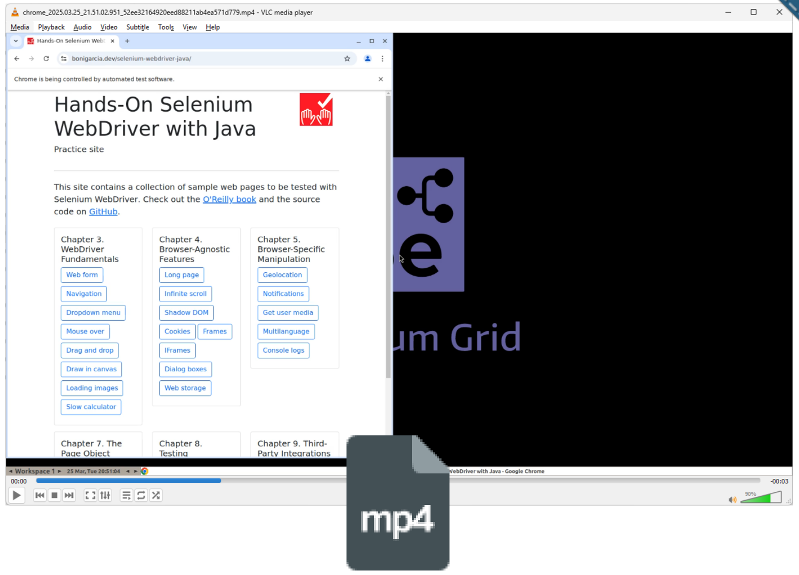
https://extentreports.com/

Fork me on GitHub

Ecosystem – Video Recording

```java
class DockerChromeRecordingTest {

    WebDriver driver;
    WebDriverManager wdm;

    @BeforeEach
    void setupTest() {
        wdm = WebDriverManager.chromedriver().browserInDocker().enableRecording();
        driver = wdm.create();
    }

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

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

}
```

Docker Selenium
Docker images for Selenium Grid


chrome_2025.03.25_21.51.02.951_52ee32164920eed88211ab4ea571d779.mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
Hands-On Selenium WebD
bonigarcia.dev/selenium-webdriver-java/
Chrome is being controlled by automated test software.
um Grid
Workspace 1 25 Mar, Tue 20:51:04
WebDriver with Java - Google Chrome
00:00
-00:03
90%
mp4

Hands-On Selenium WebDriver with Java

Practice site

This site contains a collection of sample web pages to be tested with Selenium WebDriver. Check out the O'Reilly book and the source code on GitHub.

Chapter 3. WebDriver Fundamentals: Web form, Navigation, Dropdown menu, Mouse over, Drag and drop, Draw in canvas, Loading images, Slow calculator

Chapter 4. Browser-Agnostic Features: Long page, Infinite scroll, Shadow DOM, Cookies, Frames, IFrames, Dialog boxes, Web storage

Chapter 5. Browser-Specific Manipulation: Geolocation, Notifications, Get user media, Multilanguage, Console logs

Chapter 7. The Page Object

Chapter 8. Testing

Chapter 9. Third-Party Integrations

```xml
<dependency>
   <groupId>io.github.bonigarcia</groupId>
   <artifactId>webdrivermanager</artifactId>
   <version>6.3.2</version>
   <scope>test</scope>
</dependency>
```

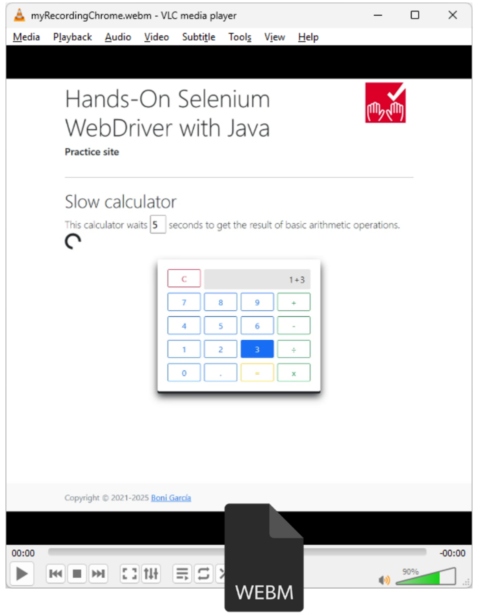
WebDriverManager

https://bonigarcia.dev/webdrivermanager/

Fork me on GitHub

Ecosystem – Video Recording

```java
class RecordEdgeTest {

    WebDriver driver;
    File targetFolder;
    WebDriverManager wdm;

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

    @Test
    void test() {
        driver.get(
                "https://bonigarcia.dev/selenium-webdriver-java/slow-calculator.html");
        wdm.startRecording();
        // test logic
        wdm.stopRecording();
    }

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

}
```


myRecordingChrome.webm - VLC media player
Media Playback Audio Video Subtitle Tools View Help
00:00
-00:00
90%
WEBM

Hands-On Selenium WebDriver with Java

Practice site

Slow calculator

This calculator waits 5 seconds to get the result of basic arithmetic operations.

C
1+3
7 8 9 +
4 5 6 -
1 2 3 ÷
0 . = x

Copyright © 2021-2025 Boni García

WebDriverManager

https://bonigarcia.dev/webdrivermanager/

Playwright

What is Playwright?

“ *Playwright is an **end-to-end testing framework****

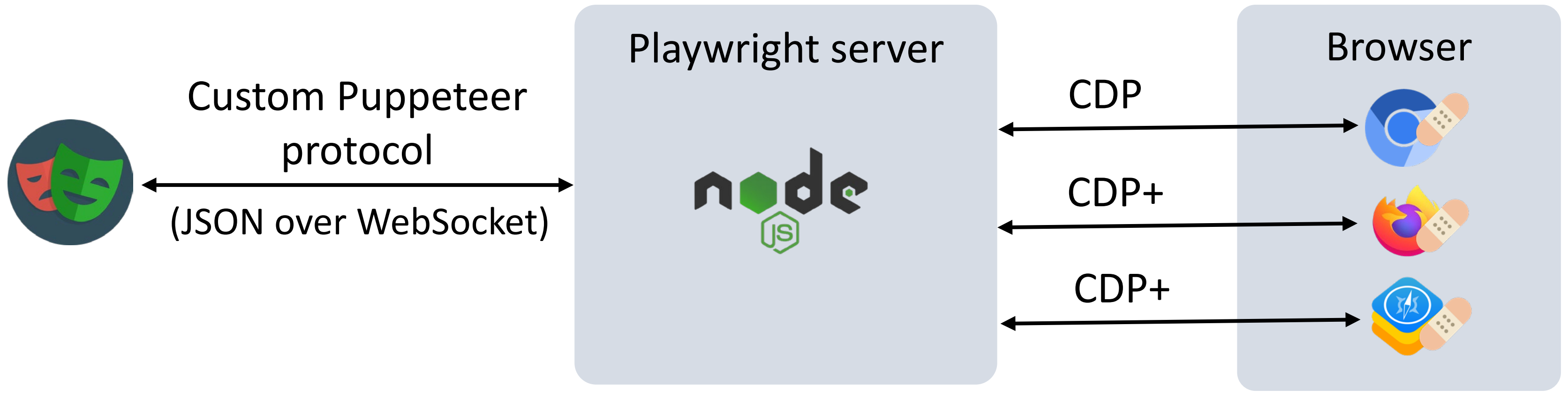
- Multilanguage
- Cross-browser
- Open-source, maintained by Microsoft since 2020

Playwright

https://playwright.dev/

Playwright Architecture

- Playwright maintains patched releases of Chromium, Firefox, and WebKit
- Playwright uses an extended version of CDP to implement to control uniformly across these browsers



Playwright Test Runner

- The Playwright Test runner (`@playwright/test`) is a full-featured testing framework bundled with Playwright in Node.js, providing:
  - Test runner and assertions (similar to JUnit/TestNG)
  - Built-in fixtures (browser/page/context lifecycle)
  - Parallel test execution across multiple browsers/devices
  - Retries mechanism
  - HTML reporting
  - Video capture on failures
  - API testing (built-in request fixture)
  - Visual comparisons (`expect(page).toHaveScreenshot()`)
  - Component testing (for React/Vue/Svelte/Angular)

What is Playwright?

“ *Playwright is an **end-to-end testing framework** for **Node.js***

“ *Playwright is a **browser automation library** for **Java/Python/.NET***

Fork me on GitHub

Playwright Hello World

```java
public class HelloWorldPlaywright {

    public static void main(String[] args) {
        try (Playwright playwright = Playwright.create()) {
            // Open Chromium
            Browser browser = playwright.chromium().launch();
            Page page = browser.newPage();

            // Navigate to web page
            String url = "https://bonigarcia.dev/selenium-webdriver-java/";
            page.navigate(url);

            // Check page title
            String title = page.title();
            System.out.println(
                    String.format("The title of %s is %s", url, title));
        } // Close Chromium
    }

}
```

```xml
<dependency>
    <groupId>com.microsoft.playwright</groupId>
     <artifactId>playwright</artifactId>
     <version>1.55.0</version>
</dependency>
```

```groovy
dependencies {
   implementation("com.microsoft.playwright:playwright:1.55.0")
}
```

Fork me on GitHub

Playwright Hello World (E2E Test)

```java
class HelloWorldPlaywrightTest {

    Browser browser;
    Page page;

    @BeforeEach
    void setup() {
        browser = Playwright.create().chromium().launch();
        page = browser.newContext().newPage();
    }

    @Test
    void test() {
        // Open system under test (SUT)
        page.navigate("https://bonigarcia.dev/selenium-webdriver-java/");

        // Assert web page title
        String title = page.title();
        assertThat(title).contains("Selenium WebDriver");
    }

    @AfterEach
    void teardown() {
        browser.close();
    }

}
```

```xml
<dependency>
    <groupId>com.microsoft.playwright</groupId>
    <artifactId>playwright</artifactId>
    <version>1.55.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter</artifactId>
    <version>6.0.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.assertj</groupId>
    <artifactId>assertj-core</artifactId>
    <version>3.27.6</version>
    <scope>test</scope>
</dependency>
```

Maven™

```
dependencies {
    testImplementation("com.microsoft.playwright:playwright:1.55.0")
    testImplementation("org.junit.jupiter:junit-jupiter:6.0.0")
    testImplementation("org.assertj:assertj-core:3.27.6")
}
```

Gradle

Fork me on GitHub

Playwright Features (in Java)

- DOM manipulation
- Impersonate user actions (keyboard, mouse)
- Browser management (screenshots, JavaScript, cookies, …)
- Auto-waiting
- Trace viewer
- Video recording
- Network interception
- Some testing features (accessibility, API testing, locator assertions)
- …

https://playwright.dev/java/docs/intro

https://github.com/bonigarcia/selenium-examples

Playwright Locators

- Playwright supports multiple selector strategies:

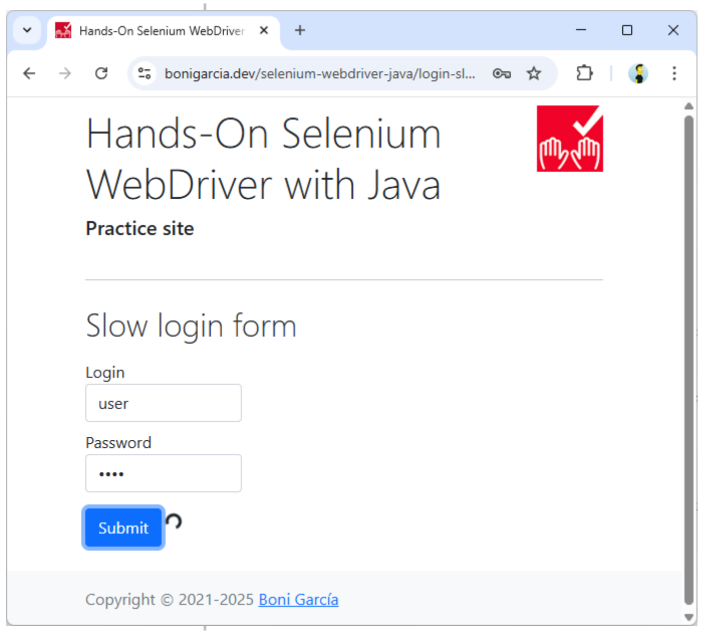
Text | Role (ARIA) | Label | Placeholder | Alt Text | Title | Test Id | CSS selector | XPath

```java
Locator submitBtn = page.getByRole(AriaRole.BUTTON, new Page.GetByRoleOptions().setName("Submit"));
Locator loginLink = page.getByText("Login");
Locator emailInput = page.getByLabel("Email address");
Locator searchBox = page.getByPlaceholder("Search...");
Locator logo = page.getByAltText("Company Logo");
Locator tooltipIcon = page.getByTitle("More info");
Locator cart = page.getByTestId("shopping-cart");
Locator userInput = page.locator("#username");
Locator item = page.locator("//div[@class='item'][1]");
```

Fork me on GitHub

Playwright Auto-Waiting

```java
class SlowLoginPlaywrightTest {

    // Fixture

    @Test
    void test() {
        // Open system under test (SUT)
        page.navigate(
                "https://bonigarcia.dev/selenium-webdriver-java/login-slow.html");

        // Log in
        page.fill("#username", "user");
        page.fill("#password", "user");
        page.click("button[type='submit']");

        // Assert expected text
        String successText = page.textContent("#success");
        assertThat(successText).contains("Login successful");

        // Take screenshot
        page.screenshot(new Page.ScreenshotOptions()
                .setPath(Paths.get("slow-login-playwright.png")));
    }

}
```


Hands-On Selenium WebDriver
bonigarcia.dev/selenium-webdriver-java/login-sl...

Hands-On Selenium WebDriver with Java

Practice site

Slow login form

Login: user

Password: ••••

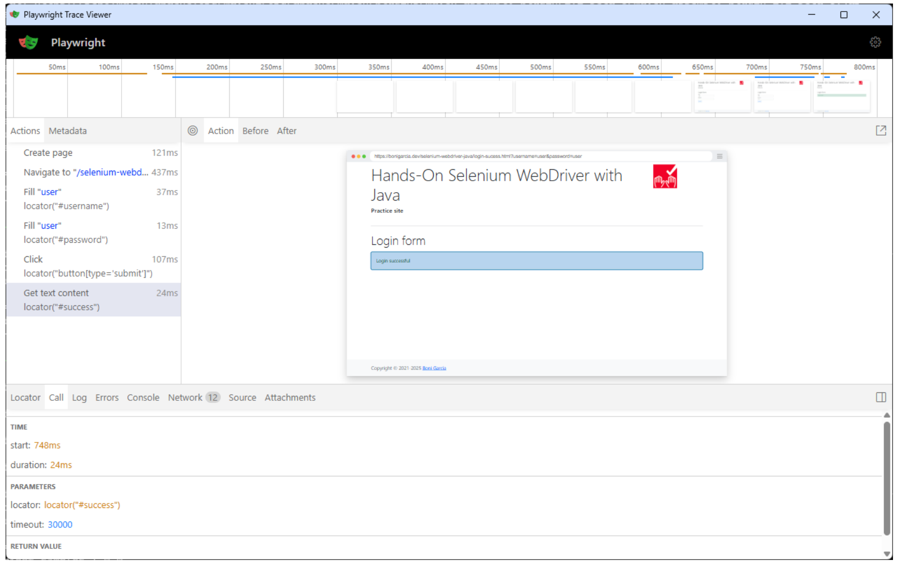
Copyright © 2021-2025 Boni García

Fork me on GitHub

Playwright Trace Viewer

```java
class TraceLoginPlaywrightTest {

    Browser browser;
    BrowserContext context;
    Page page;

    @BeforeEach
    void setup() {
        browser = Playwright.create().chromium()
                .launch(new BrowserType.LaunchOptions().setHeadless(false));
        context = browser.newContext();

        // Start tracing
        context.tracing().start(new Tracing.StartOptions().setScreenshots(true)
                .setSnapshots(true).setSources(true));

        page = context.newPage();
    }

    @Test
    void test() {
        // ...
    }

    @AfterEach
    void teardown() {
        context.tracing().stop(new Tracing.StopOptions()
                .setPath(Paths.get("login-traces.zip")));
        browser.close();
    }

}
```

```
mvn exec:java -e
  -D exec.mainClass=com.microsoft.playwright.CLI
  -D exec.args="show-trace login-traces.zip"
```



```
Create page                              121ms
Navigate to "/selenium-webd...           437ms
Fill "user"                               37ms
locator("#username")
Fill "user"                               13ms
locator("#password")
Click                                    107ms
locator("button[type='submit']")
Get text content                          24ms
locator("#success")
```

https://bonigarcia.dev/selenium-webdriver-java/login-sucess.html?username=user&password=user

Hands-On Selenium WebDriver with Java

Practice site

Login form

Login successful

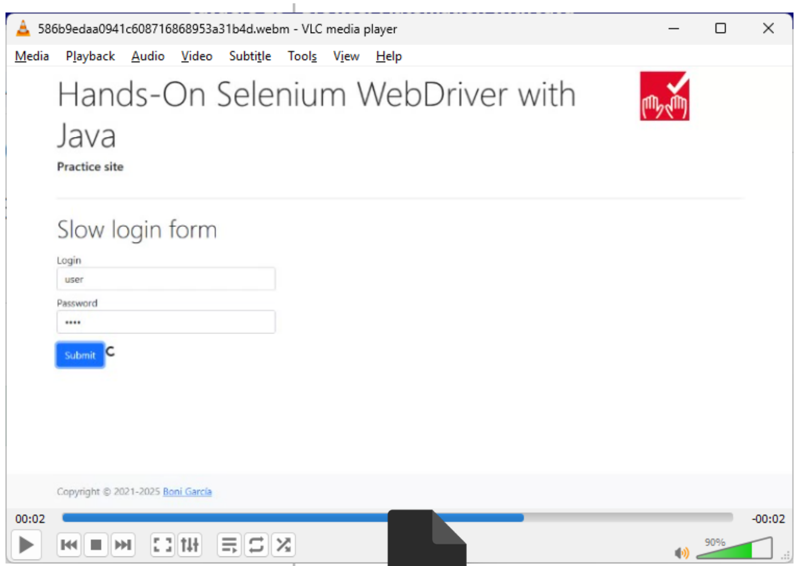
Copyright © 2021-2025 Boni Garcia

```
TIME
start: 748ms
duration: 24ms
PARAMETERS
locator: locator("#success")
timeout: 30000
RETURN VALUE
```

Fork me on GitHub

Playwright Video Recording

```java
class RecordingSlowLoginPlaywrightTest {

    Browser browser;
    Page page;

    @BeforeEach
    void setup() {
        browser = Playwright.create().chromium()
                .launch(new BrowserType.LaunchOptions().setHeadless(false));
        Browser.NewContextOptions options = new Browser.NewContextOptions()
                .setRecordVideoDir(Paths.get("."));
        page = browser.newContext(options).newPage();
    }

    @Test
    void test() {
        // ...
    }

    @AfterEach
    void teardown() {
        browser.close();
    }

}
```


586b9edaa0941c608716868953a31b4d.webm - VLC media player
Media Playback Audio Video Subtitle Tools View Help
00:02
-00:02
90%

Hands-On Selenium WebDriver with Java

Practice site

Slow login form

Login: user

Password: ••••

Submit

Copyright © 2021-2025 Boni García

WEBM

Fork me on GitHub

Ecosystem – Cross-Browser Testing

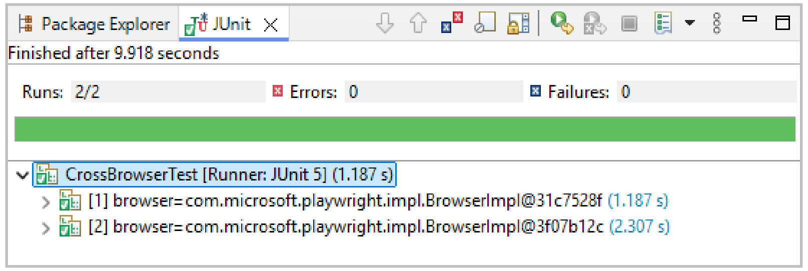
```java
class CrossBrowserTest extends CrossBrowserParent {

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

}
```

```java
public class CrossBrowserProvider implements ArgumentsProvider {

    @Override
    public Stream<? extends Arguments> provideArguments(
            ExtensionContext context) {
        Playwright playwright = Playwright.create();
        Browser chromium = playwright.chromium().launch();
        Browser firefox = playwright.firefox().launch();

        return Stream.of(Arguments.of(chromium), Arguments.of(firefox));
    }

}
```

```java
@ParameterizedClass
@ArgumentsSource(CrossBrowserProvider.class)
class CrossBrowserParent {

    @Parameter
    Browser browser;

    Page page;

    @BeforeEach
    void createContextAndPage() {
        page = browser.newContext().newPage();
    }

    @AfterEach
    void teardown() {
        browser.close();
    }

}
```

JUnit

https://junit.org/



```
Finished after 9.918 seconds

Runs: 2/2        Errors: 0        Failures: 0

CrossBrowserTest [Runner: JUnit 5] (1.187 s)
  [1] browser=com.microsoft.playwright.impl.BrowserImpl@31c7528f (1.187 s)
  [2] browser=com.microsoft.playwright.impl.BrowserImpl@3f07b12c (2.307 s)
```

Conclusions – Key Differences

	Selenium	Playwright
Nature	Browser automation library	End-to-end testing framework (JS/TS) Browser automation library (Java/Python/.NET)
Automation mechanism	Web standards (W3C WebDriver, BiDi)	Custom architecture based on own protocols and patched browsers
Languages	Java, JavaScript, Python, .NET, Ruby	JavaScript, TypeScript, Python, .NET, Java
Browsers	All major browsers	Patched Chromium, Firefox, and WebKit
Maintainer	Selenium project since 2004	Microsoft since 2020

Conclusions – Pros and Cons

	Selenium (Java)	Playwright (Java)
Pros	• Real cross-browser, since it is entirely based on open standards • Rich ecosystem • Improved developer experience (Selenium Manager)	• Great developer experience (modern API, auto-waits, easy setup) • Appealing features (trace viewer, video recording) • Faster execution
Cons	• Does not provides specific features for testing • Waits (implicit/explicit/fluent) should be handled by developers	• The test runner is not available in Java, so the advanced testing features are not available in Java • Rather than actual releases, it uses patched browser versions of Chrome, Firefox, and WebKit

Browser Automation with Java

Thank you so much!

Get these slides at:

https://bonigarcia.dev/

Boni García

boni.garcia@uc3m.es

Read this story at:

https://medium.com/@boni.gg