

Organizer:



Webbicus

Host:



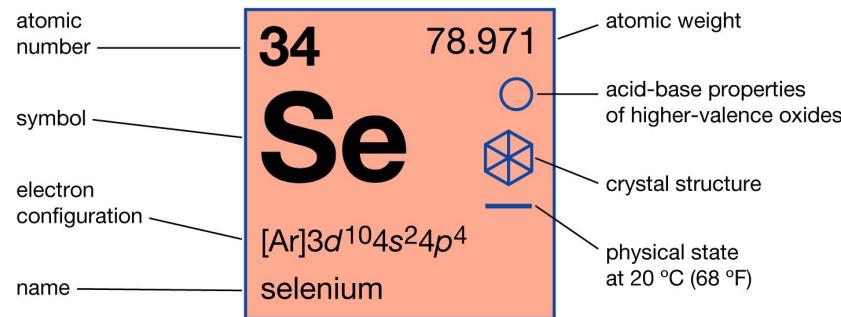
Road to Selenium 5

Boni García

Associate Professor, Universidad Carlos III de Madrid

What is Selenium?

Selenium



Other nonmetals

Solid



Hexagonal

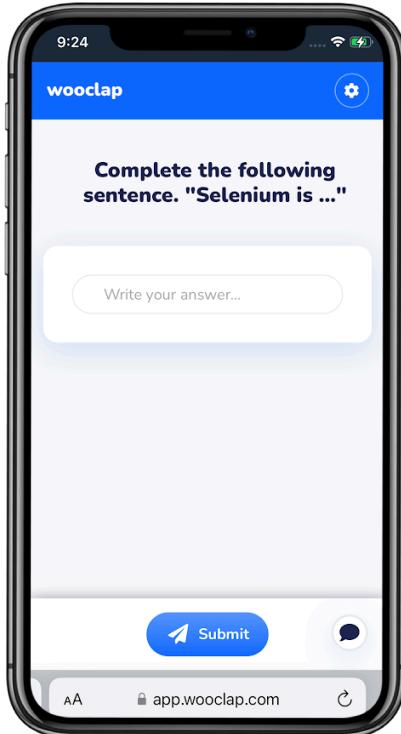
Strongly acidic

What is Selenium?

(for software qa/devs)



What is Selenium?



<https://app.wooclap.com/IEDBHR>

What is Selenium?

- Selenium WebDriver (often known as simply **Selenium**) is a multilanguage **browser automation library**



<https://selenium.dev/>

- Maintained by the Selenium project since 2004
- Languages: officially supported in Java, JavaScript, Python, .Net, and Ruby



- Browsers: any browser with a driver compliant with the **W3C WebDriver** protocol



What is Selenium?



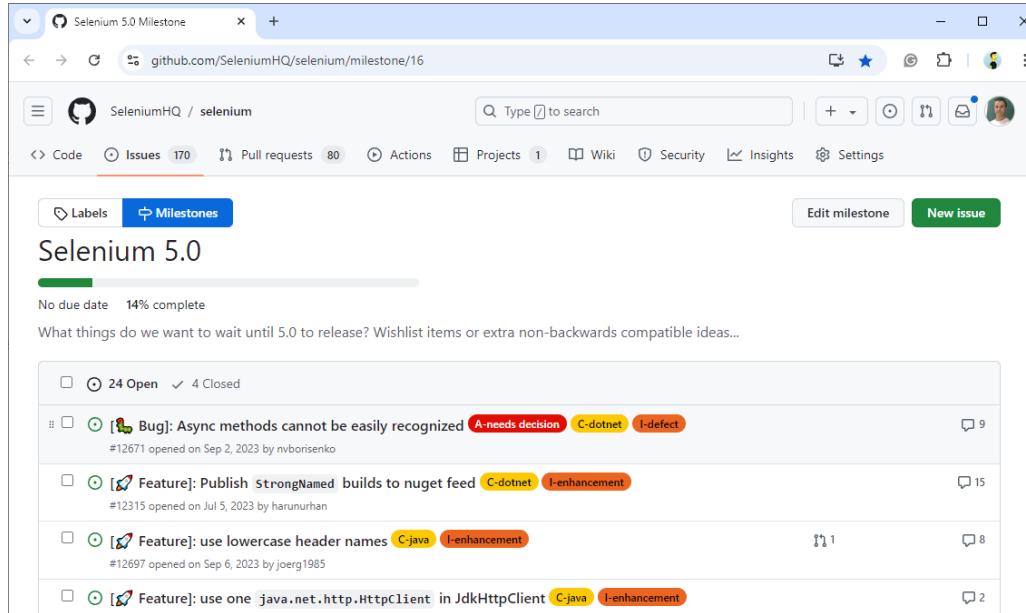
- Current Selenium stable version: **4.25** (released on September 23)



<https://www.selenium.dev/blog/2024/selenium-4-25-released/>

What is Selenium?

- Next major release (**Selenium 5**) is currently in development



The screenshot shows a GitHub browser interface for the 'SeleniumHQ / selenium' repository. The title bar says 'Selenium 5.0 Milestone'. The URL in the address bar is 'github.com/SeleniumHQ/selenium/milestone/16'. The main content is a list of issues under the 'Selenium 5.0' milestone. There are 24 open issues and 4 closed issues. The issues are categorized by type: Bug, Feature, and Enhancement. Some issues have labels like 'C-dotnet', 'I-defect', 'C-java', and 'I-enhancement'. The first few issues are:

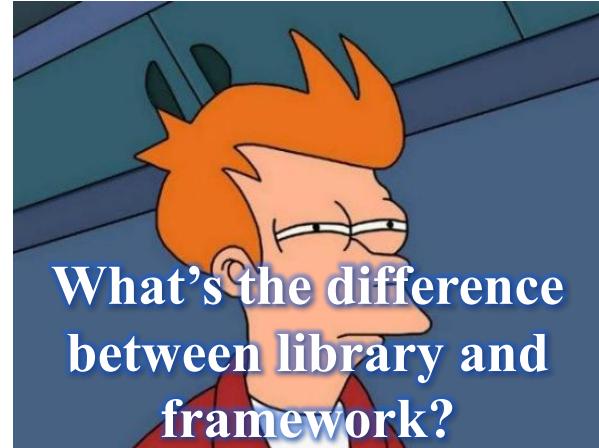
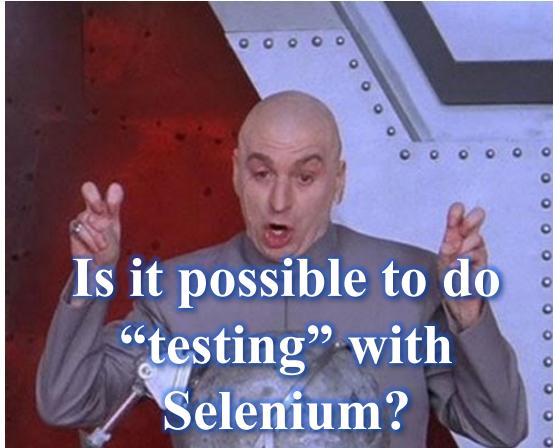
- [Bug] Async methods cannot be easily recognized (#12671)
- [Feature] Publish StrongNamed builds to nuget feed (#12315)
- [Feature] use lowercase header names (#12697)
- [Feature] use one `java.net.http.HttpClient` in JdkHttpClient (#12700)

<https://github.com/SeleniumHQ/selenium/milestone/16>

What is NOT Selenium?



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



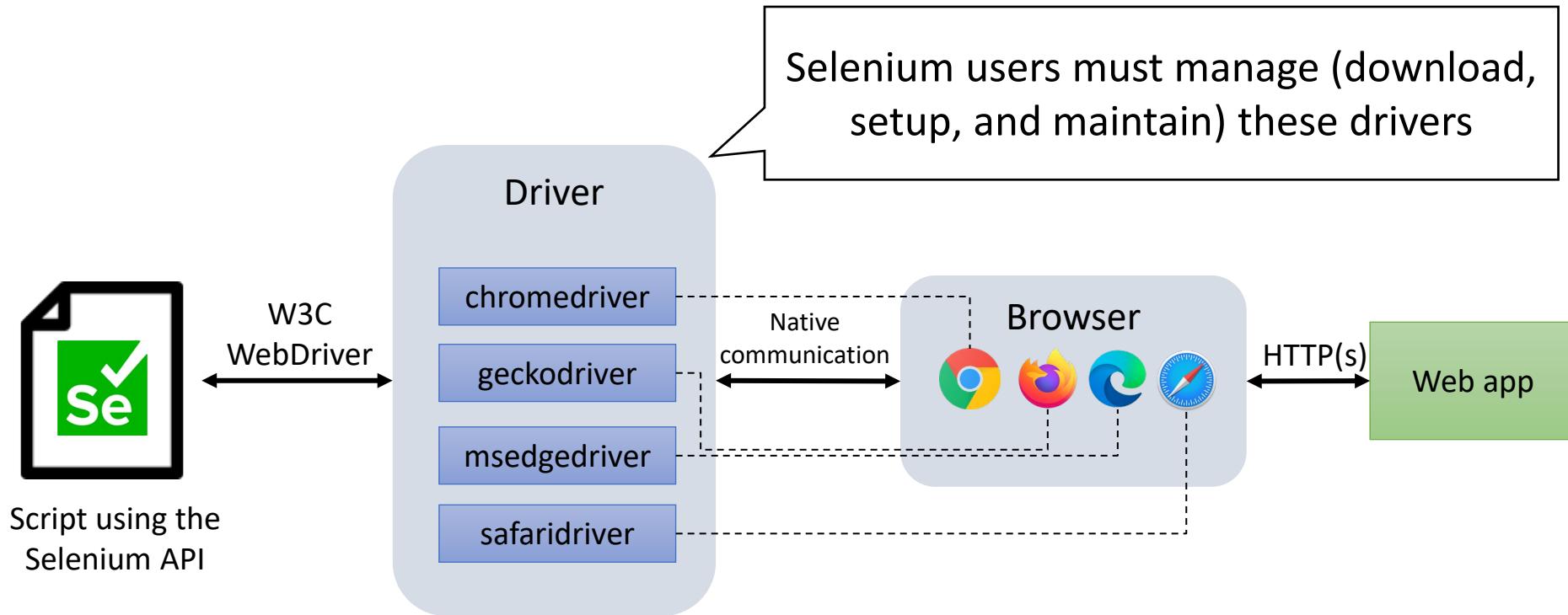
Library vs. Framework



- A **library** is a collection of code that developers can call using an API to solve a given problem
- A **framework** is a library that provides a foundational structure for developing software applications



Selenium Architecture



Hello World (Selenium 3)



```
class ChromeManualTest {  
  
    WebDriver driver;  
  
    @BeforeAll  
    static void setupClass() {  
        System.setProperty("webdriver.chrome.driver", "/path/to/chromedriver");  
    }  
  
    @BeforeEach  
    void setup() {  
        driver = new ChromeDriver();  
    }  
  
    @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();  
    }  
}
```

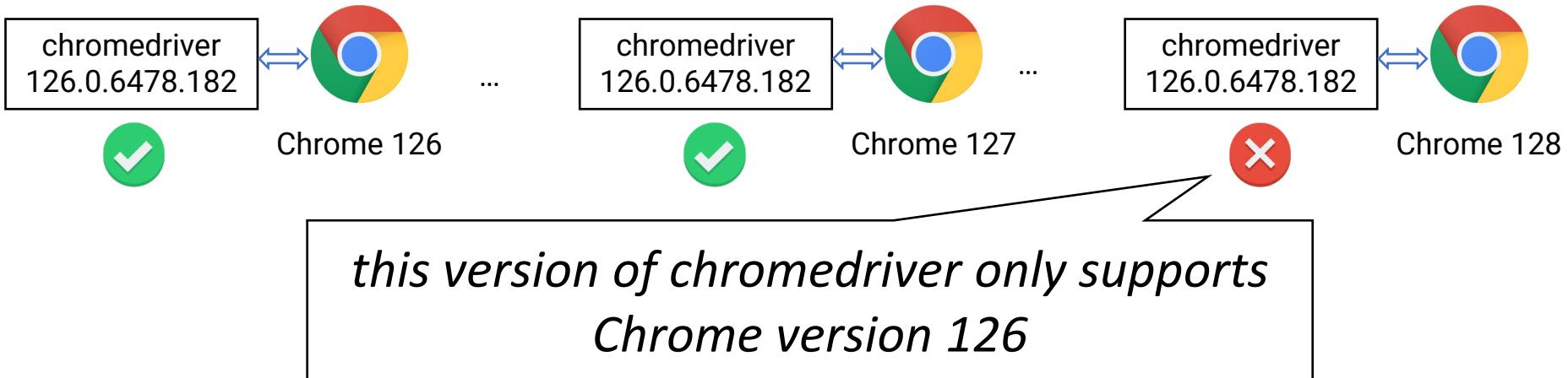
<https://github.com/bonigarcia/selenium-examples>



Driver Management



- Modern web browsers are *evergreen*



Automated Driver Management



WebDriverManager 

<https://bonigarcia.dev/webdrivermanager/>

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

Hello World (Selenium 4)



```
class ChromeWdmTest {  
  
    WebDriver driver;  
  
    @BeforeAll  
    static void setupClass() {  
        WebDriverManager.chromedriver().setup();  
    }  
  
    @BeforeEach  
    void setup() {  
        driver = new ChromeDriver();  
    }  
  
    @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();  
    }  
}
```

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



JUnit 5

AssertJ

WebDriverManager

Automated Driver Management



- In 2021, the Selenium project surveyed its users:

The screenshot shows a blog post on the Selenium website. The title is "Results of the first ever selenium survey". The post summary states: "Summary of the Selenium survey that was collected". It was written by David Burns (@AutomatedTester) on Tuesday, January 12, 2021. The categories listed are "general" and the tags are "webdriver" and "survey". The main content discusses the survey results, mentioning that 59.5% of respondents want Selenium to manage browsers. Below the main content, there are sections for "Batteries included", "Browser Management", and "Frameworks", each with a brief description and some statistics. On the right side of the page, there are two sidebar sections: "Categories" and "Tags".

Categories

- browsers: 1
- conference: 22
- docker: 1
- general: 200
- governance: 29
- releases: 45
- technical: 9

Tags

- beta: 12
- browsers: 1
- bug-bash: 2
- chat: 1
- chrome: 1

Selenium users
wanted *batteries*
included

<https://www.selenium.dev/blog/2021/selenium-survey-results/>

Selenium Manager



- I become a Selenium committer on August 2022



Selenium Manager

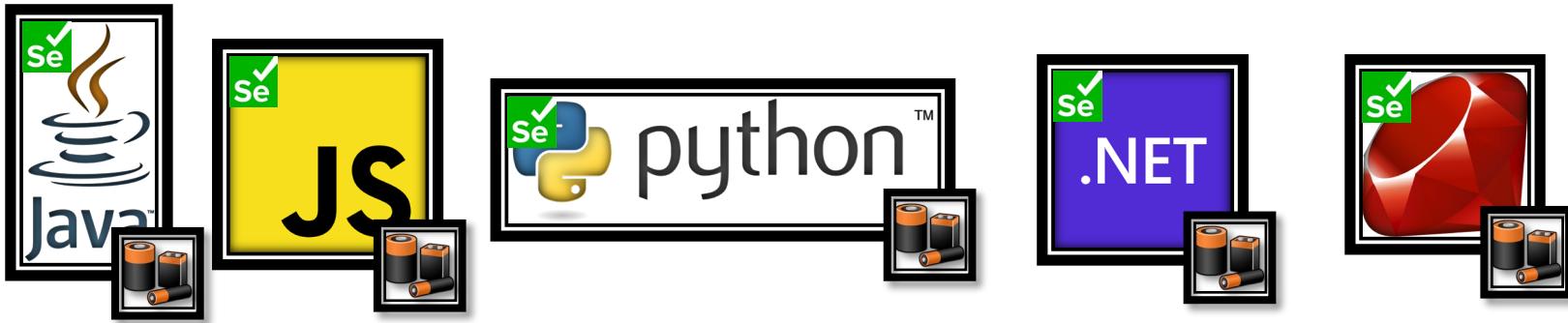
Selenium Manager is
a **CLI tool**
implemented in Rust



Selenium Manager



- Currently, each Selenium release is shipped with Selenium Manager



Hello World (Selenium 5)



```
class ChromeBasicTest {  
  
    WebDriver driver;  
  
    @BeforeEach  
    void setup() {  
        driver = new ChromeDriver();  
    }  
  
    @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();  
    }  
}
```



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

Automated Browser Management



```
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();  
    }  
}
```

What happens if
the local browser is
not available?

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

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",
"canary", "stable", or "esr")

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

Selenium Manager



- Another uses of Selenium Manager include:
 - Advanced configuration (with envs or global configuration file)
 - As a standalone CLI tool

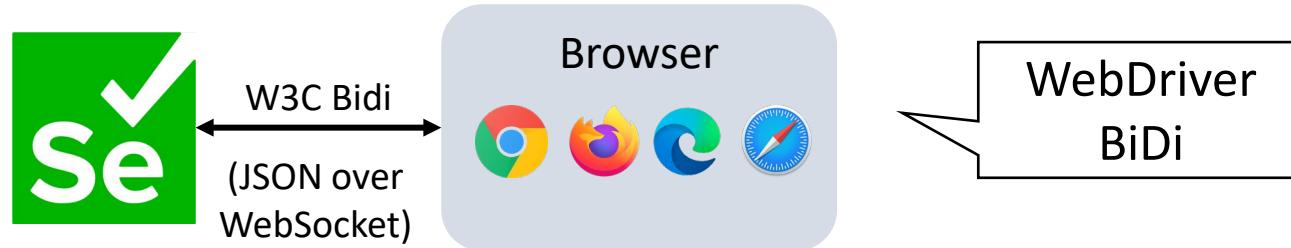
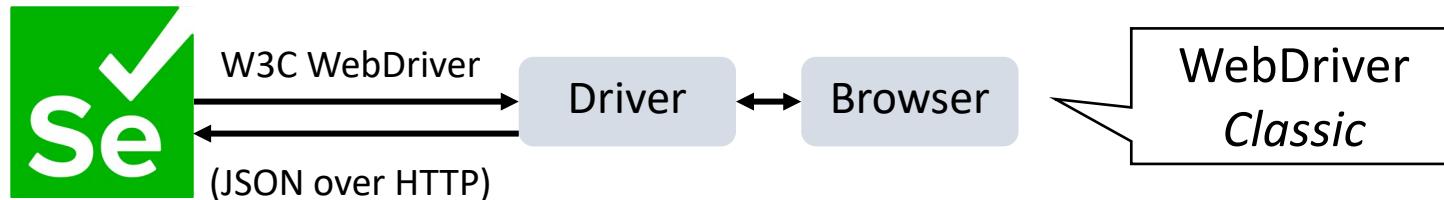


Selenium Manager

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

WebDriver BiDi

- W3C WebDriver is evolving to a new specification called **W3C BiDi (BiDirectional)**



<https://w3c.github.io/webdriver-bidi/>



WebDriver BiDi

- WebDriver BiDi features:
 - Bidirectional communication using a WebSocket
 - Event-driven architecture (e.g., for log gathering)
 - Support for modern web features (e.g., network interception)
- WebDriver BiDi modules:
 - Browsing context
 - Actions
 - Scripting
 - Logging
 - Network

<https://www.selenium.dev/documentation/webdriver/bidi/w3c/>

WebDriver BiDi



- Some of the WebDriver BiDi features are already available in Selenium 4

```
@BeforeEach  
void setup() {  
    ChromeOptions options = new ChromeOptions();  
    options.enableBiDi();  
    driver = new ChromeDriver(options);  
}
```



```
@BeforeEach  
void setup() {  
    EdgeOptions options = new EdgeOptions();  
    options.enableBiDi();  
    driver = new EdgeDriver(options);  
}
```



```
@BeforeEach  
void setup() {  
    FirefoxOptions options = new FirefoxOptions();  
    options.enableBiDi();  
    driver = new FirefoxDriver(options);  
}
```



To use WebDriver BiDi in Selenium, first we need to enable it using browser options

WebDriver BiDi

A screenshot of a Microsoft Edge browser window. The title bar says 'Hands-On Selenium WebDriver'. The address bar shows 'bonigarcia.dev/selenium-webdriver-java/console-logs.html'. The main content area displays a web page titled 'Hands-On Selenium WebDriver with Java Practice site'. Below the title is a small icon of two people. A section titled 'Console logs' contains the text: 'This page makes call to JavaScript's console (log, info, warn, error).'. On the right side of the browser, the DevTools Console tab is open, showing the following log entries:

Level	Message	File	Line
Info	This a call to 'console.log'	console-logs.html	54
Info	This a call to 'console.info'	console-logs.html	55
Warning	⚠ This a call to 'console.warn'	console-logs.html	56
Error	✖ This a call to 'console.error'	console-logs.html	57
Error	✖ Uncaught Error: This a forced error at console-logs.html:58:11	console-logs.html	58

Copyright © 2021-2024 [Boni Garcia](#)

<https://bonigarcia.dev/selenium-webdriver-java/console-logs.html>



WebDriver BiDi

```
@Test
void test() {
    List<GenericLogEntry> logs = new ArrayList<>();
    try (LogInspector logInspector = new LogInspector(driver)) {
        logInspector.onGenericLog(logs::add);
        logInspector.onConsoleEntry(logs::add);
        logInspector.onJavaScriptException(logs::add);
    }

    driver.get(
        "https://bonigarcia.dev/selenium-webdriver-java/console-logs.html");

    new WebDriverWait(driver, Duration.ofSeconds(5))
        .until(_d -> logs.size() > 3);

    for (GenericLogEntry log : logs) {
        System.out.println(log.getText());
    }
}
```

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



WebDriver BiDi

```
class ChromeBidiNetworkTest {

    WebDriver driver;

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

    @Test
    void test() throws Exception {
        try (Network network = new Network(driver)) {
            CompletableFuture<ResponseDetails> future = new CompletableFuture<>();
            network.onResponseCompleted(future::complete);
            driver.get("https://bonigarcia.dev/selenium.webdriver-java/");
            ResponseDetails response = future.get(5, TimeUnit.SECONDS);
            assertThat(response.getRequest().getMethod()).isEqualTo("GET");
            assertThat(response.getResponseData().getStatus()).isEqualTo(200);
        }
    }

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

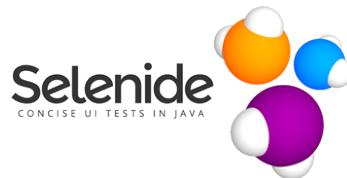
Work in progress

<https://github.com/SeleniumHQ/selenium/issues/13993>

Selenium Ecosystem



- Tools (e.g., testing frameworks) based on Selenium:



- WebDriver ecosystem:



- WebDriver BiDi ecosystem:



<https://www.selenium.dev/ecosystem/>



Thank You!

Any questions?

boni.garcia@uc3m.es