FACULDADE ENGENHARIA

Departamento de Informática

Testing with Selenium & Python

Selenium

- Selenium is a widely-used open-source framework for automating web applications
- It provides a suite of tools that allow developers and testers to simulate user interactions with web browsers
- Valuable for *functional* and *regression testing* in web development

Components

- **Selenium WebDriver:** A core component that allows for programmatically controlling a web browser. WebDriver supports multiple browsers (e.g., Chrome, Firefox, Safari, Edge) and programming languages (e.g., Java, Python, C#, JavaScript), making it versatile for cross-browser testing
- **Selenium IDE:** An integrated development environment for Selenium tests, available as a browser extension. It enables users to record and playback tests, making it suitable for quick test case development and for testers with minimal programming skills.
- **Selenium Grid:** Facilitates running tests on multiple machines with different browsers and operating systems concurrently. It's used for distributed test execution, enabling parallel testing and speeding up the testing process

Use Cases

- **Functional Testing**: Automating user interactions to ensure the application works as expected (e.g., form submissions, navigation)
- **Regression Testing**: Ensuring new changes do not break existing functionalities
- **Cross-Browser Testing**: Verifying that a web application behaves consistently across different browsers
- **Performance Testing**: When integrated with tools like *JMeter* or *Locust*, Selenium can help measure web application performance under load

Step 1: Install Selenium

- Use the following command to install Selenium:

pip install selenium

Step 2: Download WebDriver

- Download a WebDriver for the browser you want to automate (e.g., ChromeDriver): link

- Extract and note the path.

Step 3: Basic Python Script

- Import Selenium and initialize the WebDriver:

```
from selenium import webdriver
driver = webdriver.Chrome(executable_path='path_to_chromedriver')
```

Step 4: Open a Web Page

- Open a website, such as UBI:

driver.get("https://www.ubi.pt")

Step 5: Interact with the Web Page

- Locate elements and interact with them:

```
search_box = driver.find_element(By.className(NAME))
search_box.send_keys('Selenium Python')
search_box.submit()
```

Step 6: Close the Browser

- Finally, close the browser using:

```
driver.quit()
```

Exercise

Create a Python script that automates the following tasks using Selenium:

1. Visit a website.
2. Search for a specific term.
3. Print the title of the resulting page to the console.
4. Extract and print the first paragraph of the article.
5. Take a screenshot of the resulting page and save it to your local system.
6. Interact with multiple elements (e.g., clicking on links or filling out forms).
7. Close the browser.


HOW
WHY
WHERE
WHEN
WHERE
WHY
WHERE
WHO
WHO
WHEN
WHAT
WHAT
WHY
WHAT
WHEN
HOW
WHEN
WHO
WHERE
WHY
HOW