



Driver Actions and Waits



Now... what if you need to emulate a waiting period when you automate? Well, that's when the driver's actions and awaits come in hand!ç

Practice

You are tasked with automating the login functionality of the <https://www.saucedemo.com/v1/> web application.

The login process involves entering a username and password, and then clicking on the login button. However, there might be delays in the rendering of the login page or in the loading of the elements, so you need to implement appropriate wait strategies to handle these scenarios.

Start by creating a new Java Selenium project, and navigate to the web app URL. Identify and declare as WebElements all required elements to correctly execute the login process. Once you have the project configured and running, implement the following waiting strategies on different tests.

a. Implicit Wait:

- Set an implicit wait for 10 seconds using the WebDriver instance.
- Find the necessary elements using the locators and perform actions like sending keys to the username and password fields.
- Click on the login button.
- Observe how the implicit wait handles the delay between element searches and actions.

b. Explicit Wait with Expected Conditions:

- Set an explicit wait using the WebDriverWait class.
- Implement a wait condition for the visibility of the username input field.
- Once the element is visible, send keys to the username field.
- Implement a wait condition for the visibility of the password input field.
- Once the element is visible, send keys to the password field.
- Implement a wait condition for the element to be clickable for the login button.
- Once the button is clickable, click on it.
- Observe how the explicit wait waits for the specific conditions before performing actions.

c. Fluent Wait:

- Implement a fluent wait using the Wait and FluentWait classes.
- Customize the wait conditions and polling interval according to the web page behavior..
- Use fluent wait to wait for the presence of the username input field.
- Once the element is present, send keys to the username field.
- Use fluent wait to wait for the presence of the password input field.
- Once the element is present, send keys to the password field.
- Use fluent wait to wait for the element to be clickable for the login button.
- Once the button is clickable, click on it.
- Observe how fluent wait provides flexibility in setting up wait conditions.

On all the tests defined, identify success/failure situations after clicking the “Log in” button and capture them using web elements and conditionals. Then print messages according to





the test results. Use the driver to close the browser after all tests are executed.

On all test scenarios using different wait strategies, identify any differences in wait times, error handling, or other relevant observations.

This exercise will help you gain hands-on experience in implementing wait strategies in Selenium using Java. It will enhance your understanding of how waits can be used to handle delays and synchronization issues while automating web applications.

