# SELENIUM 4 NEW FEATURES

**Selenium 4 introduced several exciting features to enhance automation testing, making it more efficient, userfriendly, and comprehensive.**

Here's a list of the key new features with explanations and examples:

## 1. Relative Locators (Friendly Locators)

Why: Simplifies locating web elements by describing them relative to other elements, like "above" or "to the right of."

Example:

WebElement loginButton = driver.findElement(with(By.tagName("button")).above(By.id("password")));

This helps in cases where traditional locators might not be consistent or unique across pages.

Shyam Pandhare

## 2. Improved Window and Tab Management

Why: Browser automation often requires switching between tabs or windows. Selenium 4 improves how new tabs and windows are opened and managed.

Example: driver.switchTo().newWindow(WindowType.TAB);

It enables you to easily switch between tabs or windows without managing handles manually.

## 3. New Actions for Handling Authentication Popups

Why: Handling HTTP authentication popups was challenging in previous versions. Selenium 4 introduces support to simplify this.

Example: driver.get("https://username:password@securedsite.com");

This feature is useful for automating applications requiring login authentication without needing manual entry.

# SELENIUM 4 NEW FEATURES

## 4. Improved Selenium Grid

**Why: The Selenium Grid was redesigned to be more scalable and support a modern infrastructure with Docker integration, grid sessions, and observability.**

**Example: Grid now has Docker support, making it easier to set up and manage distributed testing environments. The revamped UI and observability logs allow better insights and debugging capabilities.**

**This is crucial for large teams who need to execute tests in parallel across multiple environments efficiently.**

## 5. Native Support for Chrome DevTools Protocol (CDP)

**Why: Selenium 4 adds CDP support to allow more granular browser control, such as emulating network conditions and capturing console logs.**

**Example:**

```
DevTools devTools = driver.getDevTools();

devTools.createSession();

devTools.send(Network.emulateNetworkConditions(false, 100, 2000, 1000, Optional.empty()));
```

**This allows testers to simulate various network conditions, which is valuable for testing page performance and behavior under different conditions.**

Shyam Pandhare

## 6. Enhanced Documentation and W3C Compliance

**Why: Selenium 4 uses the W3C WebDriver standard by default, improving compatibility and reducing inconsistencies.**

**Example: With W3C compliance, the WebDriver commands are standardized, reducing compatibility issues across different browsers and environments.**

**This change helps achieve better test reliability, especially across different browsers.**

# SELENIUM 4 NEW FEATURES

## 7. New Element Screenshot Functionality

**Why: Provides a way to capture screenshots of individual web elements rather than the whole page, useful for visual testing.**

**Example:**

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

File screenshot = element.getScreenshotAs(OutputType.FILE);
```

**This feature is beneficial for comparing UI elements individually without capturing the entire screen.**

## 8. BiDi (Bidirectional) Communication

**Why: Enables Selenium to listen for events from the browser, providing realtime feedback, especially for scenarios where capturing browser events is necessary.**

**Example: The BiDi protocol can monitor browser console logs or Script errors dynamically.**

**This enhances debugging, especially for monitoring frontend errors or tracking events.**

## 9. Better Support for Web Elements' Actions

**Why: Enhances the `Actions` API to be more intuitive and robust, making complex interactions like draganddrop, hover, and key presses smoother.**

**Example:**

```
Actions actions = new Actions(driver);

actions.moveToElement(element).clickAndHold().moveByOffset(50, 50).release().perform();
```

**Improved interactions streamline complex testing scenarios, especially in applications with interactive components.**

Shyam Pandhare

# SELENIUM 4 NEW FEATURES

## 10. Grid 4 UI and Observability

**Why: A new UI for Selenium Grid allows easier monitoring of sessions, nodes, and logs.**

**Example: The new console provides a comprehensive view of node status, which is helpful for understanding test distributions.**

**This enhanced observability improves test efficiency and diagnostics in distributed setups.**

## 11. Improved Error Messages

**Why: Error messages have been revamped to provide more details, making it easier to identify issues.**

**Example: If an element is not interactable, the error message is more specific about why.**

**This helps developers and testers debug test failures faster by providing more context.**

**Each of these features addresses specific gaps or challenges in Selenium 3, making Selenium 4 a more robust and efficient tool for modern web testing needs. Let me know if you'd like more examples or indepth guidance on any feature!**

Shyam Pandhare