Selenium 4 features

Selenium 4 was launched on October 13, 2021. Selenium 4 represents the most recent version with advanced features, including Selenium Grid, W3C compliance, an enhanced IDE, new APIs, and more. The architecture of Selenium Grid has undergone a redesign in the latest releases of Selenium 4. You can run parallel and distributed tests on nodes and as well as hubs.

However, the most significant change under the hood is the W3C compliance of WebDriver APIs, ensuring more stable and less error-prone cross-browser tests.

# Improved Grid

[Selenium Grid](https://toolsqa.com/selenium-webdriver/%22https:/toolsqa.com/selenium-webdriver/selenium-grid-how-to-easily-setup-a-hub-and-node/%22) is now enhanced with Docker support, simplifying the setup and scaling of Selenium Grid using containers. It also supports IPv6 addresses and HTTPS communication. In Selenium 4, the grid experience is smoother, as there's no longer a need to set up and start hubs and nodes separately. Once you initiate a Selenium server, the grid acts as both a hub and node.

## Different Grid Types

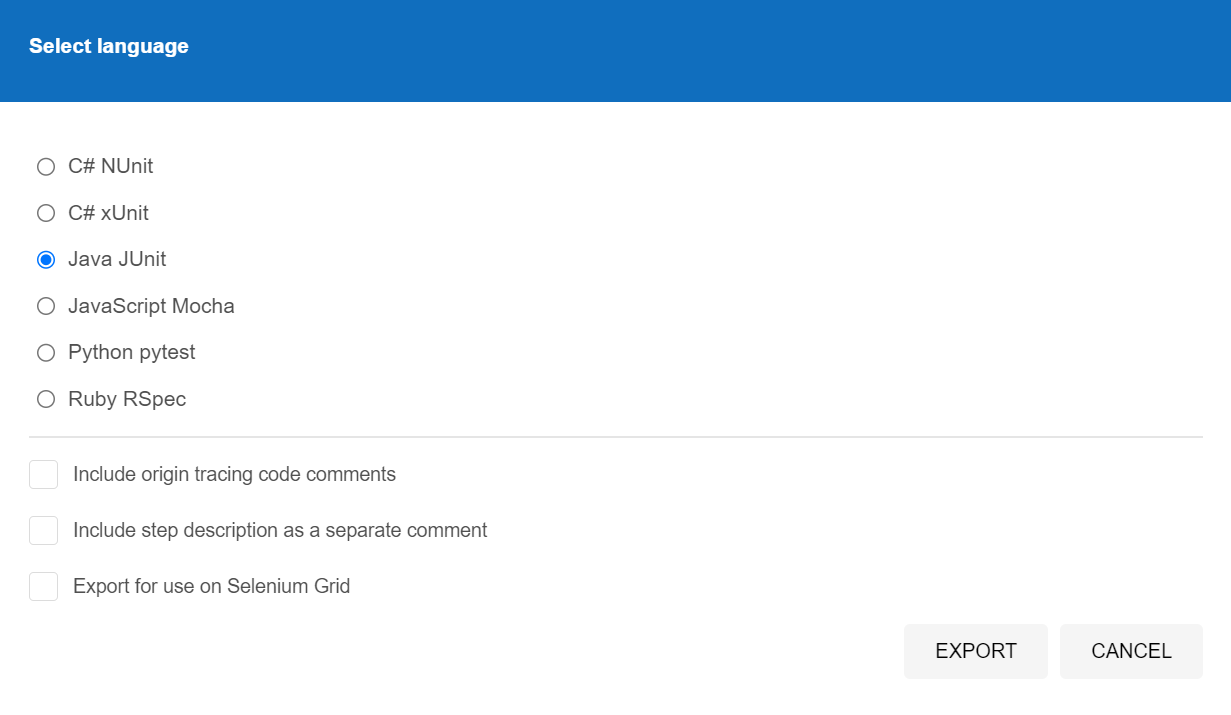
Selenium 4 supports three grid types:

* Standalone Mode
* Classical Grid (Hub and Node)
* Fully Distributed (Router, Distributor, Session, and Node)

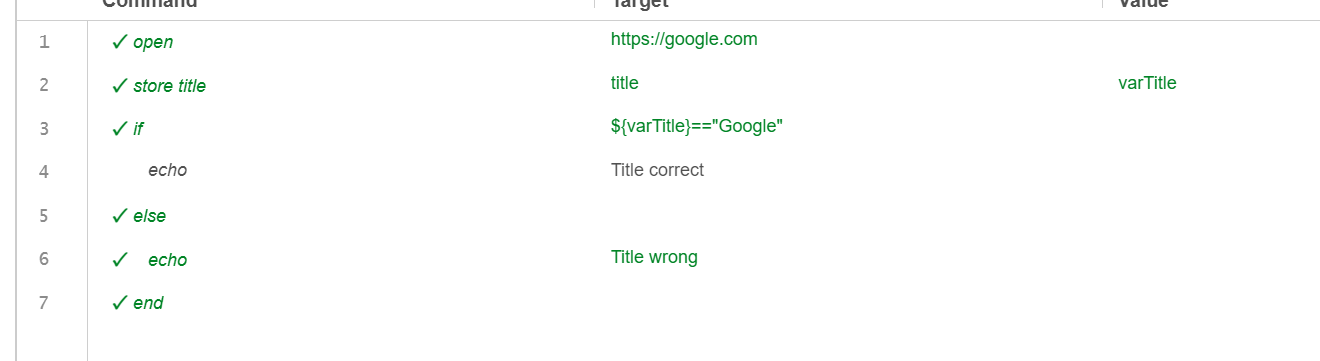
# Upgraded Selenium IDE

The new Selenium 4 IDE provides:

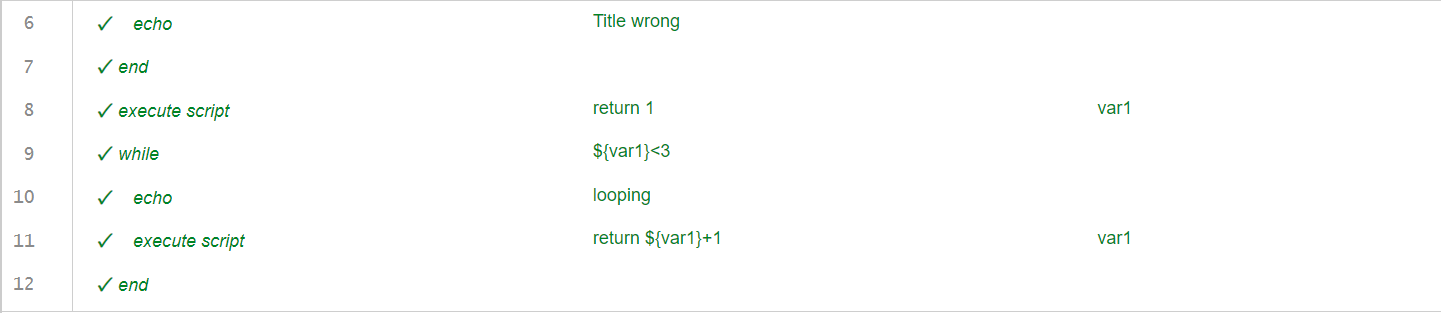
1. Improved GUI
2. The new IDE comes with an enhanced element locator strategy which helps locate an element in case the web element couldn’t be located. Records multiple locators.
3. Built-in wait mechanism
4. We have the export feature to export in different language options



1. Debugging option by adding breakpoint and highlighting the element.
2. Control flow statements added



---------------------While-------------------------------



JavaScript to get Page URL

execute script

return window.location.href

pageURL

# Relative Locators

This functionality brings a new way to [locate elements](https://docs.saucelabs.com/web-apps/automated-testing/selenium/#finder-methods) to help you find the ones that are nearby other elements. Relative locators are used to identify an element that does not have enough information to locate it uniquely on a web application.

* above
* below
* toLeftOf
* toRightOf
* near

import org.openqa.selenium.support.locators.RelativeLocator

WebDriver driver=**new** ChromeDriver();

driver.get("https://magento.softwaretestingboard.com/customer/account/create/");

WebElement element1 = driver.findElement(By.*cssSelector*("label[for='firstname']"));

WebElement e = driver.findElement(RelativeLocator.*with*(By.*tagName*("input")).below(element1));

e.sendKeys("ABCD");

# Chrome Devtools Protocol

This feature can be used for emulating different devices.

ChromeDriver driver=**new** ChromeDriver();

DevTools devTool = driver.getDevTools(); // Create devTool instance

devTool.createSession();

devTool.send(Emulation.setDeviceMetricsOverride(

400,

500,

50,

**true**,

Optional.empty(),

Optional.empty(),

Optional.empty(),

Optional.empty(),

Optional.empty(),

Optional.empty(),

Optional.empty(),

Optional.empty(),

Optional.empty(),

Optional.empty()

));

driver.get("https://magento.softwaretestingboard.com/customer/account/create/");

------------------Another method------------------

ChromeDriver driver=**new** ChromeDriver();

driver.executeCdpCommand("Emulation.setDeviceMetricsOverride",Map.*of*(

"width", 500,

"height", 600,

"deviceScaleFactor", 50,

"mobile", **true**

));

# Better Window/Tab Management in Selenium 4

There are several instances in test automation wherein one might need to open a particular link in a new tab or window to perform certain actions. To achieve this in Selenium 3, QAs had to create a new driver object and then perform the **switch operation** using the **WindowHandle method** to perform subsequent steps.

In selenium 4 it can be done as below

driver.get("https://www.google.com/");

// Opens a new window and switches to new window

driver.switchTo().newWindow(WindowType.***WINDOW***);

// Opens BrowserStack homepage in the newly opened window

driver.navigate().to("https://www.linkedin.com/");

For a tab, WINDOW will be replaced with tab

# Deprication of desired capabilities

Desired Capabilities were primarily used in the test scripts to define the test environment (browser name, version, operating system) for execution on the Selenium Grid.

In Selenium 4, capabilities objects are replaced with Options. This means testers now need to create an Options object, set test requirements, and pass the object to the Driver constructor.