

# Playwright - Java Cheat Sheet - Browser Auto...

**Author:** Naveen Khunteta

**Linkedin:** <https://www.linkedin.com/in/naveenkhunteta/>

**Playwright - Java Youtube Series:** <https://t.ly/hZ9Rh>



Playwright is an open-source Node.js library for automating web browsers such as Chromium, Firefox, and WebKit through a single API.

It allows developers to write browser automation scripts in a simple and intuitive way using the programming language of their choice, including Java, JavaScript, Python, and others.

Playwright provides powerful features such as cross-browser testing, headless mode, request interception, and more, making it a popular tool for testing and scraping web applications.

Playwright is maintained by Microsoft and is designed to be fast, reliable, and easy to use.

Method/Property	Description
<b>Playwright.create()</b>	Creates a new instance of Playwright.
<b>BrowserType.launch()</b>	Launches a new browser instance.
<b>Browser.newContext()</b>	Creates a new browser context.
<b>Context.newPage()</b>	Creates a new page in the current context.
<b>Page.navigate()</b>	Navigates the current page to a new URL.
<b>Page.querySelector()</b>	Finds the first element that matches the specified select or.

<b>Page.querySelectorAll()</b>	Finds all elements that match the specified selector.
<b>Page.waitForSelector()</b>	Waits for the specified selector to appear on the page.
<b>Page.waitForLoadState()</b>	Waits for the page to reach the specified load state.
<b>Page.title()</b>	Gets the title of the page.
<b>Page.url()</b>	Gets the URL of the page.
<b>Page.goBack()</b>	Navigates the page back in history.
<b>Page.goForward()</b>	Navigates the page forward in history.
<b>Page.reload()</b>	Reloads the page.
<b>Page.close()</b>	Closes the page.
<b>Page.setViewportSize()</b>	Sets the size of the viewport.
<b>Page.setExtraHTTPHeaders()</b>	Sets extra HTTP headers to be sent with every request.
<b>Page.setRequestInterception()</b>	Enables or disables request interception.
<b>Page.setOfflineMode()</b>	Enables or disables offline mode.
<b>Page.route()</b>	Intercepts network requests and invokes the route handler.
<b>Page.setUserAgent()</b>	Sets the user agent for the page.
<b>Page.waitForNavigation()</b>	Waits for the page to finish navigating to a new URL.
<b>Page.waitForPopup()</b>	Waits for a popup window to appear.
<b>Page.waitForDownload()</b>	Waits for a download to finish.
<b>Page.evaluate()</b>	Executes a JavaScript function in the page context.
<b>Page.evaluateHandle()</b>	Evaluates a JavaScript expression in the page context and returns a handle to the result.
<b>Page.addInitScript()</b>	Injects a JavaScript file or script into the page context.

<b>Page.exposeBinding()</b>	Exposes a Java object as a JavaScript function in the page context.
<b>Page.click()</b>	Simulates a click event on an element.
<b>Page.fill()</b>	Sets the value of an input element.
<b>Page.selectOption()</b>	Selects an option from a dropdown menu.
<b>Page.hover()</b>	Simulates a hover event on an element.
<b>Page.focus()</b>	Sets focus on an element.
<b>Page.press()</b>	Sends a keyboard event to an element.
<b>Page.type()</b>	Types the specified text into an element.
<b>Page.keyboard()</b>	Provides methods for simulating keyboard input.
<b>Page.mouse()</b>	Provides methods for simulating mouse input.
<b>Page.touchscreen()</b>	Provides methods for simulating touchscreen input.
<b>ElementHandle.innerText()</b>	Gets the inner text content of an element.
<b>ElementHandle.getAttribute()</b>	Gets the value of the specified attribute for an element.

Method/Property	Description
<b>Browser.newContext(options: Browser.ContextOptions?)</b>	Creates a new browser context.
<b>Browser.close()</b>	Closes the browser.
<b>Browser.newPage(): Page</b>	Creates a new page in the current context.
<b>BrowserContext.pages(): List&lt;Page&gt;</b>	Gets a list of all pages in the current context.
<b>BrowserContext.clearCookies()</b>	Clears all cookies in the current context.

<b>BrowserContext.addCookies(cookies: List&lt;Cookie&gt;)</b>	Adds cookies to the current context.
<b>BrowserContext.clearPermissions()</b>	Clears all permissions in the current context.
<b>BrowserContext.setGeolocation(geolocation: Geolocation)</b>	Sets the geolocation for the current context.
<b>BrowserContext.exposeBinding(name: String, callback: BindingCallback)</b>	Exposes a Java object as a JavaScript function in the current context.
<b>BrowserType.launch(options: BrowserType.LaunchOptions?)</b>	Launches a new browser instance.
<b>BrowserType.connectOverCDP(endpointURL: String)</b>	Connects to an existing browser instance using the Chrome DevTools Protocol.

Method/Property	Description
<b>Playwright.chromium()</b>	Returns a <b>BrowserType</b> instance for Chromium.
<b>Playwright.firefox()</b>	Returns a <b>BrowserType</b> instance for Firefox.
<b>Playwright.webkit()</b>	Returns a <b>BrowserType</b> instance for WebKit.
<b>Playwright.selectors().register(name: String, script: String)</b>	Registers a new selector engine with the specified name and JavaScript script.
<b>Playwright.selectors().register(name: String, scriptPath: Path)</b>	Registers a new selector engine with the specified name and JavaScript script file.
<b>Playwright.selectors().register(name: String, script: URL)</b>	Registers a new selector engine with the specified name and JavaScript script URL.

Method	Description
<b>Browser.newContext()</b>	Creates a new browser context.
<b>Context.newPage()</b>	Creates a new page in the current context.

<b>Page.navigate()</b>	Navigates the current page to a new URL.
<b>Page.querySelector()</b>	Finds the first element that matches the specified selector.
<b>Page.querySelectorAll()</b>	Finds all elements that match the specified selector.
<b>ElementHandle.innerText()</b>	Gets the inner text content of an element.
<b>ElementHandle.getAttribute()</b>	Gets the value of the specified attribute for an element.
<b>ElementHandle.click()</b>	Simulates a click event on the element.
<b>ElementHandle.fill()</b>	Sets the value of an input element.
<b>ElementHandle.selectOption()</b>	Selects an option from a dropdown menu.
<b>ElementHandle.hover()</b>	Simulates a hover event on the element.
<b>ElementHandle.focus()</b>	Sets focus on the element.
<b>ElementHandle.press()</b>	Sends a keyboard event to the element.
<b>ElementHandle.type()</b>	Types the specified text into the element.
<b>ElementHandle.isVisible()</b>	Returns true if the element is visible on the page.
<b>ElementHandle.isDisabled()</b>	Returns true if the element is disabled.
<b>ElementHandle.isEditable()</b>	Returns true if the element is editable.
<b>ElementHandle.boundingBox()</b>	Returns the bounding box of the element.
<b>Page.waitForSelector()</b>	Waits for the specified selector to appear on the page.
<b>Page.waitForLoadState()</b>	Waits for the page to finish loading.
<b>Page.screenshot()</b>	Takes a screenshot of the current page.



# THANKS FOR WATCHING

NAVEEN AUTOMATION LABS