**Interface WebDriver**

* **All Superinterfaces:**

[SearchContext](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/SearchContext.html)

**All Known Implementing Classes:**

[ChromeDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/chrome/ChromeDriver.html), [ChromiumDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/chromium/ChromiumDriver.html), [EdgeDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/edge/EdgeDriver.html), [EventFiringWebDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/support/events/EventFiringWebDriver.html), [FirefoxDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/firefox/FirefoxDriver.html), [InternetExplorerDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/ie/InternetExplorerDriver.html), [OperaDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/opera/OperaDriver.html), [RemoteWebDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/remote/RemoteWebDriver.html), [SafariDriver](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/safari/SafariDriver.html)

public interface **WebDriver**

extends [SearchContext](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/SearchContext.html)

WebDriver is a remote control interface that enables introspection and control of user agents (browsers). The methods in this interface fall into three categories:

* + Control of the browser itself
  + Selection of [WebElement](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html)s
  + Debugging aids

Key methods are [get(String)](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#get(java.lang.String)), which is used to load a new web page, and the various methods similar to [findElement(By)](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#findElement(org.openqa.selenium.By)), which is used to find [WebElement](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html)s.

Currently, you will need to instantiate implementations of this interface directly. It is hoped that you write your tests against this interface so that you may "swap in" a more fully featured browser when there is a requirement for one.

Most implementations of this interface follow [W3C WebDriver specification](https://w3c.github.io/webdriver/)

* + ***Nested Class Summary***

|  |  |  |
| --- | --- | --- |
| **Nested Classes** | | |
| **Modifier and Type** | **Interface** | **Description** |
| static interface | [**WebDriver.ImeHandler**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.ImeHandler.html) | **Deprecated.**  Will be removed. |
| static interface | [**WebDriver.Navigation**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Navigation.html) |  |
| static interface | [**WebDriver.Options**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Options.html) | An interface for managing stuff you would do in a browser menu |
| static interface | [**WebDriver.TargetLocator**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html) | Used to locate a given frame or window. |
| static interface | [**WebDriver.Timeouts**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Timeouts.html) | An interface for managing timeout behavior for WebDriver instances. |
| static interface | [**WebDriver.Window**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Window.html) |  |

* + ***Method Summary***

|  |  |  |
| --- | --- | --- |
| **All Methods**[**Instance Methods**](javascript:show(2);)[**Abstract Methods**](javascript:show(4);) | | |
| **Modifier and Type** | **Method** | **Description** |
| void | [**close**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#close())() | Close the current window, quitting the browser if it's the last window currently open. |
| [**WebElement**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html) | [**findElement**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#findElement(org.openqa.selenium.By))​([**By**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/By.html) by) | Find the first [WebElement](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html) using the given method. |
| java.util.List<[**WebElement**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html)> | [**findElements**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#findElements(org.openqa.selenium.By))​([**By**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/By.html) by) | Find all elements within the current page using the given mechanism. |
| void | [**get**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#get(java.lang.String))​(java.lang.String url) | Load a new web page in the current browser window. |
| java.lang.String | [**getCurrentUrl**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#getCurrentUrl())() | Get a string representing the current URL that the browser is looking at. |
| java.lang.String | [**getPageSource**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#getPageSource())() | Get the source of the last loaded page. |
| java.lang.String | [**getTitle**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#getTitle())() | Get the title of the current page. |
| java.lang.String | [**getWindowHandle**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#getWindowHandle())() | Return an opaque handle to this window that uniquely identifies it within this driver instance. |
| java.util.Set<java.lang.String> | [**getWindowHandles**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#getWindowHandles())() | Return a set of window handles which can be used to iterate over all open windows of this WebDriver instance by passing them to [switchTo()](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#switchTo()).[WebDriver.Options.window()](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Options.html#window()) |
| [**WebDriver.Options**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Options.html) | [**manage**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#manage())() | Gets the Option interface |
| [**WebDriver.Navigation**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Navigation.html) | [**navigate**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#navigate())() | An abstraction allowing the driver to access the browser's history and to navigate to a given URL. |
| void | [**quit**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#quit())() | Quits this driver, closing every associated window. |
| [**WebDriver.TargetLocator**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html) | [**switchTo**](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#switchTo())() | Send future commands to a different frame or window. |

* + ***Method Detail***
    - **get**

void get​(java.lang.String url)

Load a new web page in the current browser window. This is done using an HTTP POST operation, and the method will block until the load is complete (with the default 'page load strategy'. This will follow redirects issued either by the server or as a meta-redirect from within the returned HTML. Should a meta-redirect "rest" for any duration of time, it is best to wait until this timeout is over, since should the underlying page change whilst your test is executing the results of future calls against this interface will be against the freshly loaded page. Synonym for [WebDriver.Navigation.to(String)](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Navigation.html#to(java.lang.String)).

**Parameters:**

url - The URL to load. Must be a fully qualified URL

* + - **getCurrentUrl**

java.lang.String getCurrentUrl()

Get a string representing the current URL that the browser is looking at.

See [W3C WebDriver specification](https://w3c.github.io/webdriver/#get-current-url) for more details.

**Returns:** The URL of the page currently loaded in the browser

* + - **getTitle**

java.lang.String getTitle()

Get the title of the current page.

**Returns:** The title of the current page, with leading and trailing whitespace stripped, or null if one is not already set

* + - **findElements**

java.util.List<[WebElement](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html)> findElements​([By](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/By.html) by)

Find all elements within the current page using the given mechanism. This method is affected by the 'implicit wait' times in force at the time of execution. When implicitly waiting, this method will return as soon as there are more than 0 items in the found collection, or will return an empty list if the timeout is reached.

**Specified by:** [findElements](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/SearchContext.html#findElements(org.openqa.selenium.By)) in interface [SearchContext](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/SearchContext.html)

**Parameters:** by - The locating mechanism to use

**Returns:** A list of all matching [WebElement](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html)s, or an empty list if nothing matches

* + - **findElement**

[WebElement](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html) findElement​([By](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/By.html) by)

Find the first [WebElement](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html) using the given method. This method is affected by the 'implicit wait' times in force at the time of execution. The findElement(..) invocation will return a matching row, or try again repeatedly until the configured timeout is reached.

findElement should not be used to look for non-present elements, use [findElements(By)](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#findElements(org.openqa.selenium.By)) and assert zero length response instead.

**Specified by:** [findElement](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/SearchContext.html#findElement(org.openqa.selenium.By)) in interface [SearchContext](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/SearchContext.html)

**Parameters:** by - The locating mechanism to use

**Returns:** The first matching element on the current page

**Throws:** [NoSuchElementException](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/NoSuchElementException.html) - If no matching elements are found

* + - **getPageSource**

java.lang.String getPageSource()

Get the source of the last loaded page. If the page has been modified after loading (for example, by Javascript) there is no guarantee that the returned text is that of the modified page. Please consult the documentation of the particular driver being used to determine whether the returned text reflects the current state of the page or the text last sent by the web server. The page source returned is a representation of the underlying DOM: do not expect it to be formatted or escaped in the same way as the response sent from the web server. Think of it as an artist's impression.

**Returns:** The source of the current page

* + - **close**

void close()

Close the current window, quitting the browser if it's the last window currently open.

* + - **quit**

void quit()

Quits this driver, closing every associated window.

* + - **getWindowHandles**

java.util.Set<java.lang.String> getWindowHandles()

Return a set of window handles which can be used to iterate over all open windows of this WebDriver instance by passing them to [switchTo()](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html#switchTo()).[WebDriver.Options.window()](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Options.html#window())

**Returns:** A set of window handles which can be used to iterate over all open windows.

* + - **getWindowHandle**

java.lang.String getWindowHandle()

Return an opaque handle to this window that uniquely identifies it within this driver instance. This can be used to switch to this window at a later date

**Returns:** the current window handle

* + - **switchTo**

[WebDriver.TargetLocator](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html) switchTo()

Send future commands to a different frame or window.

**Returns:** A TargetLocator which can be used to select a frame or window

* + - **navigate**

[WebDriver.Navigation](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Navigation.html) navigate()

An abstraction allowing the driver to access the browser's history and to navigate to a given URL.

**Returns:** A [WebDriver.Navigation](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Navigation.html) that allows the selection of what to do next

* + - **manage**

[WebDriver.Options](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.Options.html) manage()

Gets the Option interface

**Returns:** An option interface