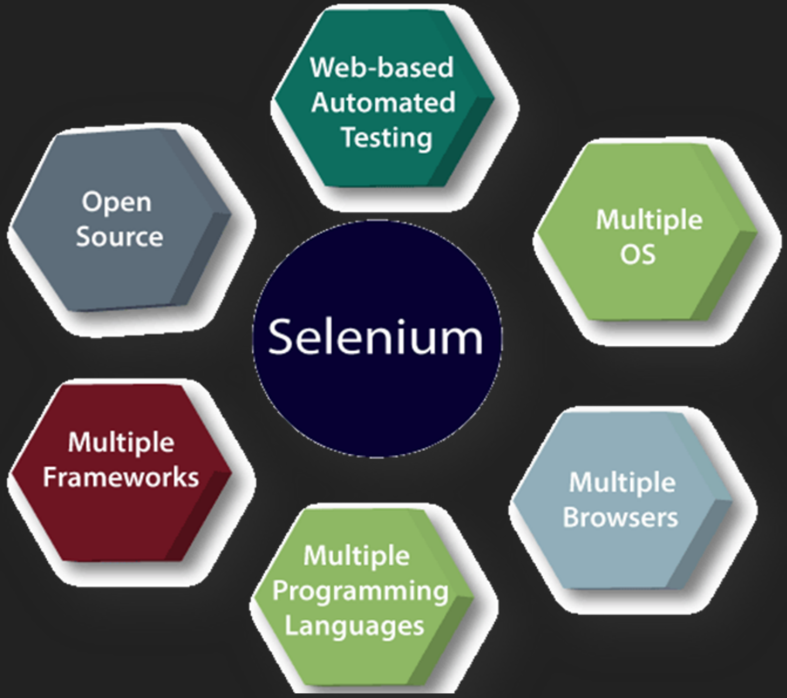
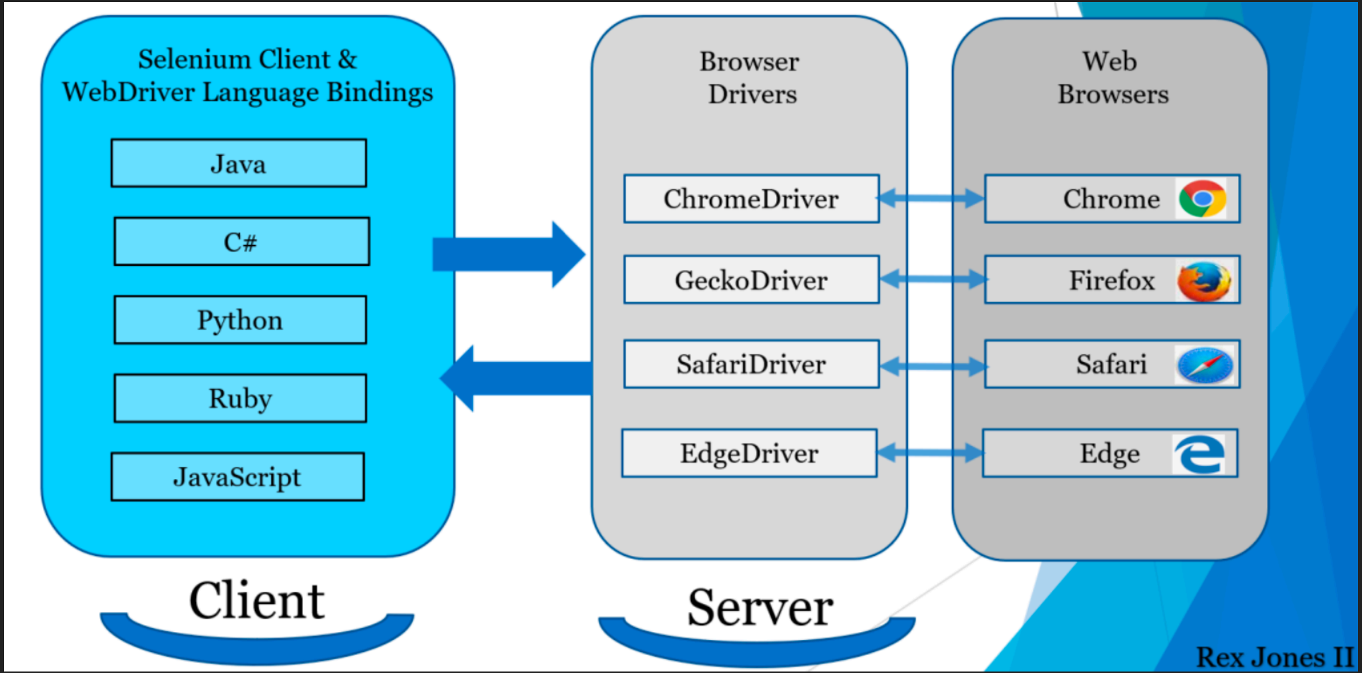
SELENIUM

WHY SELENIUM



ARCHITECTURE



CONFIGURE SELENEIUM

Prerequisite:

- Java
- Any IDE

Using Selenium jar as an External dependency

- Using Maven
- Using Gradle

MY MACHINE CONFIG

Selenium Version : 4.9.0

Browsers Configuration

Firefox Version: 110

https://download-installer.cdn.mozilla.net/pub/firefox/releases/110.0/win64/en-US/

Chrome Version: 121

ChromeDriver: 120

GeckoDriver: https://github.com/mozilla/geckodriver/releases/tag/v0.32.0

webDriverManager: version 5.2.0

Firefox Release

https://download-installer.cdn.mozilla.net/pub/firefox/releases/

DRIVER CONFIGURATION

Browser	Supported OS	Maintained by	Download
Chromium/Chrome	Windows/macOS/Linux	Google	Downloads
Firefox	Windows/macOS/Linux	Mozilla	Downloads
Edge	Windows 10	Microsoft	Downloads
Safari	macOS El Capitan and newer	Apple	Built in
Opera	Windows/macOS/Linux	Opera	Downloads

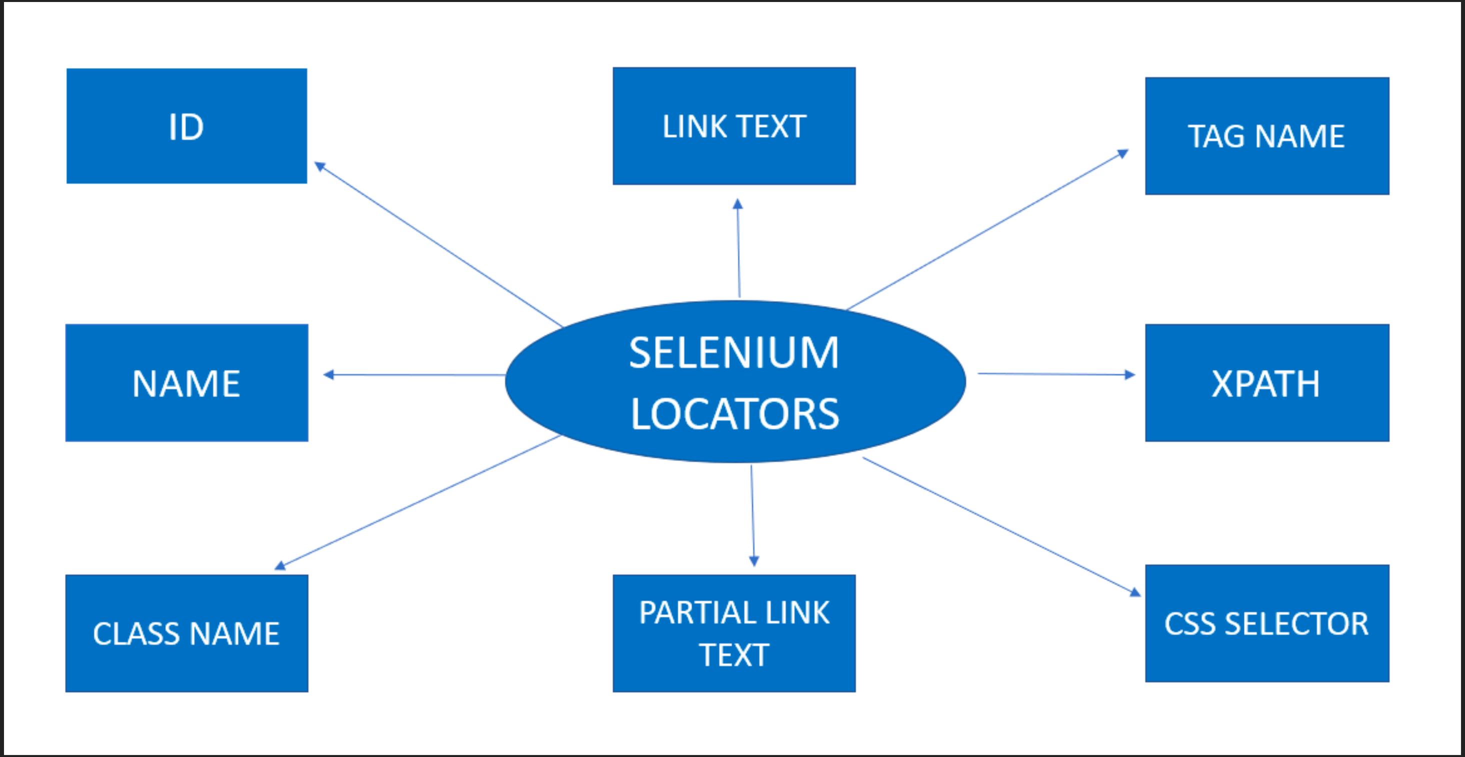
BROWSER NAVIGATION

- driver.manage().window().maximize();
- driver.get("https://www.google.co.in/");
- driver.getCurrentUrl();
- driver.navigate().to("");
- driver.navigate().refresh();
- driver.navigate().forward();
- driver.navigate().back();
- driver.getTitle();

DRIVER QUIT

- driver.close
- driver.quit

LOCATORS


ID
LINK TEXT
TAG NAME
NAME
SELENIUM LOCATORS
XPATH
CLASS NAME
PARTIAL LINK TEXT
CSS SELECTOR

WEBELEMENT

WebElement represents an HTML element.

HTML documents are made up by HTML elements.

HTML elements are written with a start tag, with an end tag, with the content in between: <tagname> content </tagname>

```
WebElement element = driver.findElement(By.id(“UserName”));
```

- element.clear();
- element.sendKeys("text");
- element.click();
- element.submit();
- element.getText();
- element.getTagName();
- element.getAttribute();

FINDELEMENT VS FINDELEMENTS

Selenium WebDriver provides two methods using which we can find an element or list of elements on a web page. These are:

findElement(): This method uniquely finds a web element on the web page.

findElements(): This method finds a list of web elements on the web page.

```
WebElement elementName = driver.findElement (By.LocatorStrategy("LocatorValue"));
```

This command accepts the "By " object as the argument and returns a WebElement object.

```
List<WebElement> elementName = driver.findElements(By.LocatorStrategy("LocatorValue"));
```

findElement() which returned a unique element. If there are no matching elements, then an empty list returns.

```
List<WebElement> elementName = driver.findElements(By.LocatorStrategy("LocatorValue"));
[element1, element2, element3]
elementName.get(1).click();
elementName.click();
```

PROBLEM STATEMENT – WEB ELEMENT

ABSOLUTE & RELATIVE XPATH

XPATH

Locator	Explanation
//img[@id='myId']	image element with @id= 'myId'
//img[@id!='myId']	image elements with @id not equal to 'myId'
//img[@name]	image elements that have name attribute
//*[contains(@id, 'Id')]	element with @id containing
//*[starts-with(@id, 'Id')]	element with @id starting with
//*[ends-with(@id, 'Id')]	element with @id ending with
//*[matches(@id, 'r')]	element with @id matching regex ‘r’
//*[@name='myName']	image element with @name= 'myName'
//*[@id='X' or @name='X']	element with @id X or a name X
//*[@name="N"][@value="v"]	element with @name N & specified @value ‘v’
//*[@name="N" and @value="v"]	element with @name N & specified @value ‘v’
//*[@name="N" and not(@value="v")]	element with @name N & not specified @value ‘v’
//input[@type="submit"]	input of type submit
//a[@href="url"]	anchor with target link 'url'
//section[//h1[@id='hi']]	returns <section> if it has an <h1> descendant with @id= 'hi'
//*[@id="TestTable"]//tr[3]//td[2]	cell by row and column
//input[@checked]	checkbox (or radio button) that is checked
//a[@price > 2.50]	'a' with price > 2.5

XPATH METHODS

Locator	Explanation
//table[count(tr) > 1]	return table with more than 1 row
//*[.="t"]	element containing text 't' exactly
//a[contains(text(), "Log Out")]	anchor with inner text containing 'Log Out'
//a[not(contains(text(), "Log Out"))]	anchor with inner text not containing 'Log Out'
//a[not(@disabled)]	all 'a' elements that are not disabled

XPATH AXIS

Ancestor

XPath#1: //div[@class='Mammal']/ancestor::div

XPath#2: //div[@class='Mammal']/
ancestor::div[@class='Animal']

Child

XPath#1: //div[@class='Mammal']/child::div

XPath#2: //div[@class='Mammal']/
child::div[@class='Herbivore']/h5

Descendent

XPath#1: //div[@class='Animal']/descendant::div

Following

XPath: //div[@class='Mammal']/following::div

Following-sibling

XPath: //div[@class='Mammal']/following-sibling::div

Preceding

XPath#1: //div[@class='Other']/preceding::div

Preceding-sibling

XPath#1: //div[@class='Other']/preceding-sibling::div

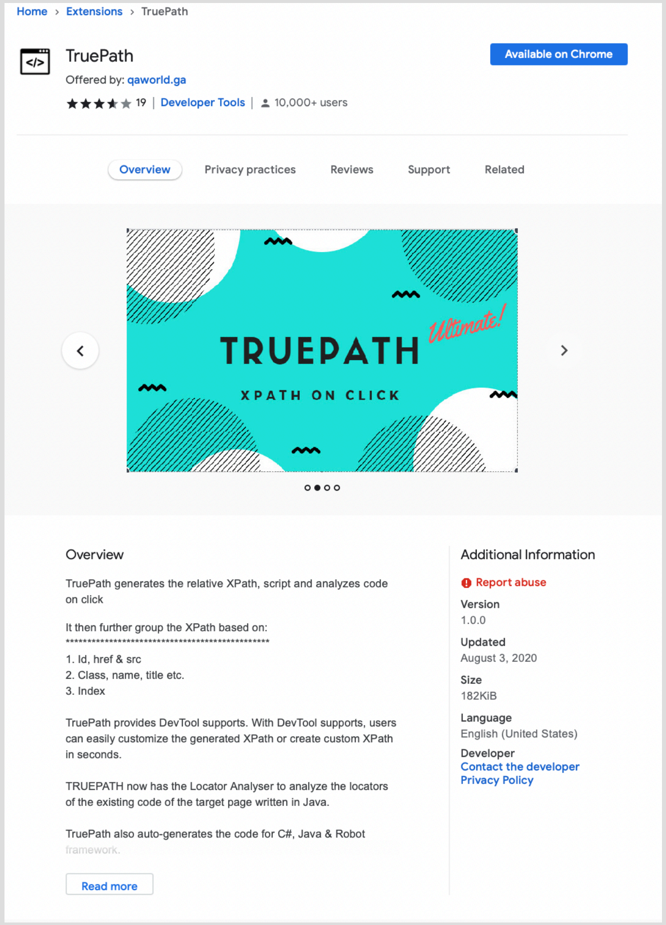
Parent

XPath: //div[@class='Mammal']/parent::div

Example: Preceding (with index)

XPath: //div[@class='Other']/preceding::div[1]

TRUEPATH



Home › Extensions › TruePath

TruePath

Offered by: qaworld.ga

★★★★★ 19 | Developer Tools | 10,000+ users

Available on Chrome

Overview Privacy practices Reviews Support Related

Overview

TruePath generates the relative XPath, script and analyzes code on click

It then further group the XPath based on:
****

1. Id, href & src
2. Class, name, title etc.
3. Index

TruePath provides DevTool supports. With DevTool supports, users can easily customize the generated XPath or create custom XPath in seconds.

TRUEPATH now has the Locator Analyser to analyze the locators of the existing code of the target page written in Java.

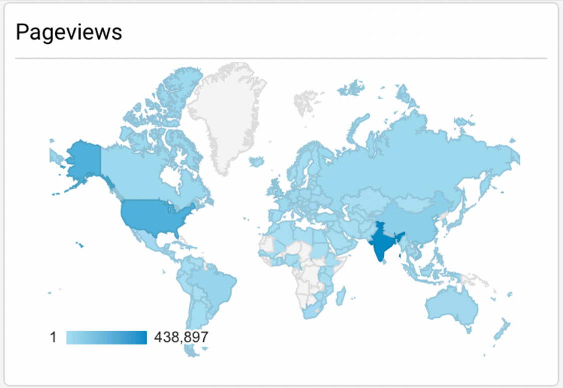
TruePath also auto-generates the code for C#, Java & Robot framework.

Read more

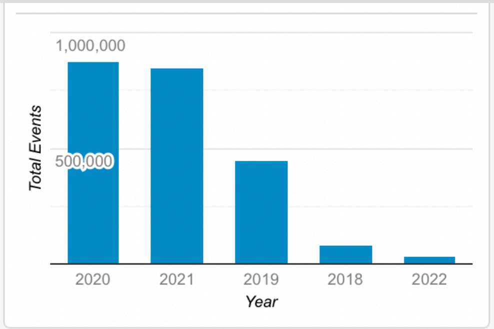
Additional Information

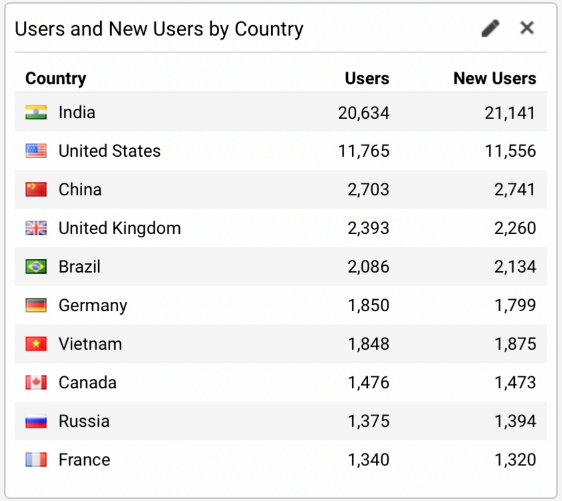
Report abuse

Version	1.0.0
Updated	August 3, 2020
Size	182KiB
Language	English (United States)
Developer	Contact the developer Privacy Policy



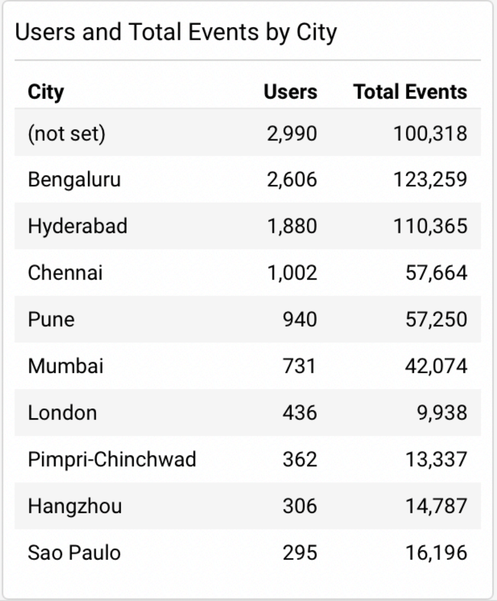
Pageviews





Users and New Users by Country

Country	Users	New Users
India	20,634	21,141
United States	11,765	11,556
China	2,703	2,741
United Kingdom	2,393	2,260
Brazil	2,086	2,134
Germany	1,850	1,799
Vietnam	1,848	1,875
Canada	1,476	1,473
Russia	1,375	1,394
France	1,340	1,320



Users and Total Events by City

City	Users	Total Events
(not set)	2,990	100,318
Bengaluru	2,606	123,259
Hyderabad	1,880	110,365
Chennai	1,002	57,664
Pune	940	57,250
Mumbai	731	42,074
London	436	9,938
Pimpri-Chinchwad	362	13,337
Hangzhou	306	14,787
Sao Paulo	295	16,196

XPATH EXCERCISE

Selenium4

RELATIVE LOCATORS

- Above
- Below
- Left
- Right
- Nearby
- Chaining of relative locators

CONTROLS

- Textbox: Sendkeys
- Buttons: Click & Submit
- Links

CONTROLS

- Radio Button
- Check Box
- Multi Select
- Options
  - isSelected()
  - isDisplayed()
  - isEnabled()

ACTION CLASS

Actions class is an ability provided by Selenium for handling keyboard and mouse events.
In Selenium WebDriver, handling these events includes operations such as drag and drop, clicking on multiple elements with the control key. These operations are performed using the advanced user interactions API. It mainly consists of *Actions* that are needed while performing these operations.

Action class is defined and invoked using the following syntax:

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

action.moveToElement(element).click().perform();
```

Mouse Actions:

1. **doubleClick()**: Performs double click on the element
2. **clickAndHold()**: Performs long click on the mouse without releasing it
3. **moveToElement()**: Shifts the mouse pointer to the center of the element
4. **dragAndDrop()**: Drags the element from one point and drops to another
5. **contextClick()**: Performs right-click on the mouse

Keyboard Actions:

1. **sendKeys()**: Sends a series of keys to the element
2. **keyUp()**: Performs key release
3. **keyDown()**: Performs keypress without release

ALERT

- Simple
- Confirmation
- Prompt

To dismiss

driver.switchTo().alert().dismiss();

To accept

Driver.switchTo().alert().accept();

To capture alert message

driver.switchTo().alert().getText();

To send some data to alert
boxdriver.switchTo().alert().sendKeys("Text");

FRAME & WINDOW

- iFrame
- Window
- Selenium 4 implementation

Frame:

```
driver.switchTo().frame(driver.findElement(By.id("frame1")));
driver.switchTo().parentFrame();
driver.switchTo().defaultContent();
```

Window:

To capture windows id

```
String MainWindow=driver.getWindowHandle();
```

To switch child window

```
driver.switchTo().window(ChildWindow);
```

To switch main window

```
driver.switchTo().window(MainWindow);
```

Selenium4

WINDOW HANDLING

- New Window
- New Tab

JAVASCRIPT

```
JavascriptExecutor jse = (JavascriptExecutor) driver;
jse.executeScript("window.scrollBy(0,250)");
```

WEB DRIVER MANAGER

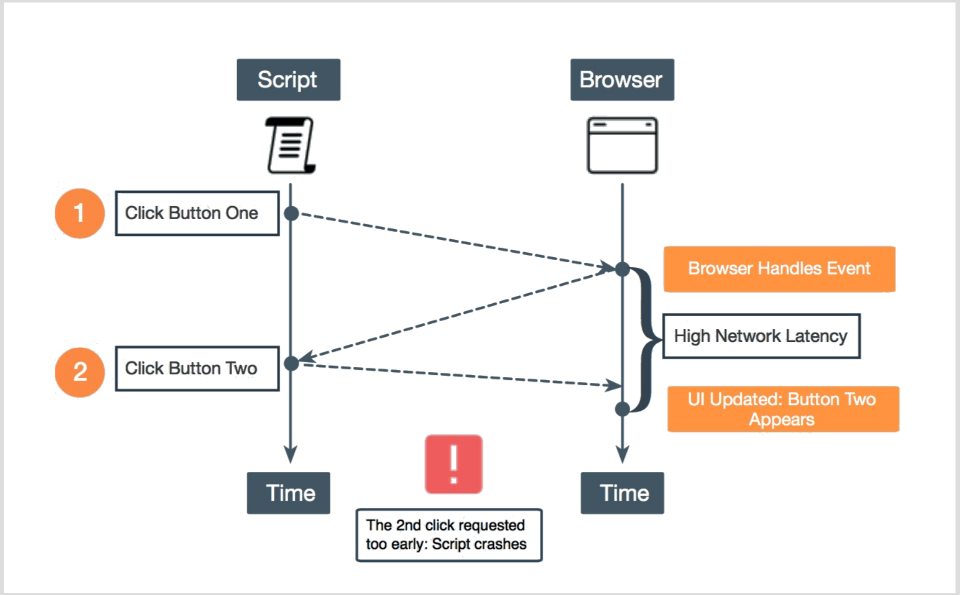
WebDriverManager is an open-source Java library that carries out the management (i.e., download, setup, and maintenance) of the drivers required by Selenium WebDriver (e.g., chromedriver, geckodriver, msedgedriver, etc.) in a fully automated manner.

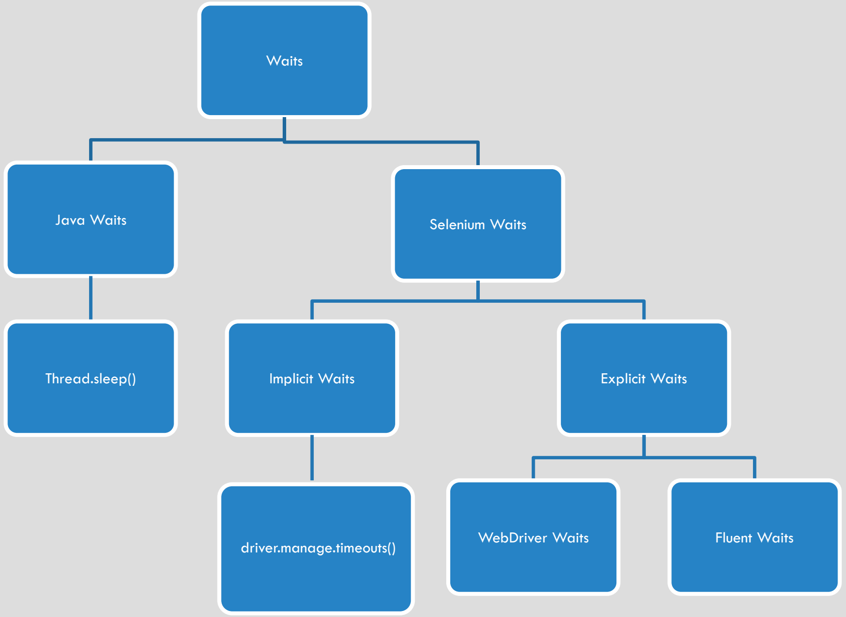
In addition, WebDriverManager provides other relevant features, such as the capability to discover browsers installed in the local system, building WebDriver objects (such as ChromeDriver, FirefoxDriver, EdgeDriver, etc.), and running browsers in Docker containers seamlessly.

```
<dependency>
      <groupId>io.github.bonigarcia</groupId>
      <artifactId>webdrivermanager</artifactId>
      <version>5.0.3</version>
</dependency>
```

```
WebDriverManager.chromedriver().setup();
```

SYNCHRONISATION





Selenium4

SYNTAX CHANGES

Duration:

```
driver.manage().timeouts().~~implicitlyWait~~(30, TimeUnit.SECONDS);

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));
```

```
 WebDriverWait wait = new WebDriverWait(driver,5);

 WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(5));
```

OTHER

- HTML Table
- Screen Prints
  - Native Selenium Screen Print
  - Full Screen-Ashot
  - Full Screen-Shutterbug
- Chrome Options:
  - https://www.tabnine.com/code/java/methods/org.openqa.selenium.chrome.ChromeOptions/addArguments

SCREEN SHOT

Selenium4

```
WebDriver augmentedDriver = new Augmenter().augment(driver);

File file = ((HasFullPageScreenshot) augmentedDriver).getFullPageScreenshotAs(OutputType.FILE);

Path fullPageScreenshot = Paths.get(directory + “TakeFullPageScreenshotFirefox.png");

Files.move(file.toPath(), fullPageScreenshot);
```

PRINT-PDF

Selenium4

```
ChromeOptions options = new ChromeOptions();

options.setHeadless(true);

ChromeDriver driver = new ChromeDriver(options);

Path printPage = Paths.get(directory + "PrintPageFirefox.pdf");

Pdf print = driver.print(new PrintOptions());

Files.write(printPage, OutputType.BYTES.convertFromBase64Png(print.getContent()));
```

CHROME DEV TOOL

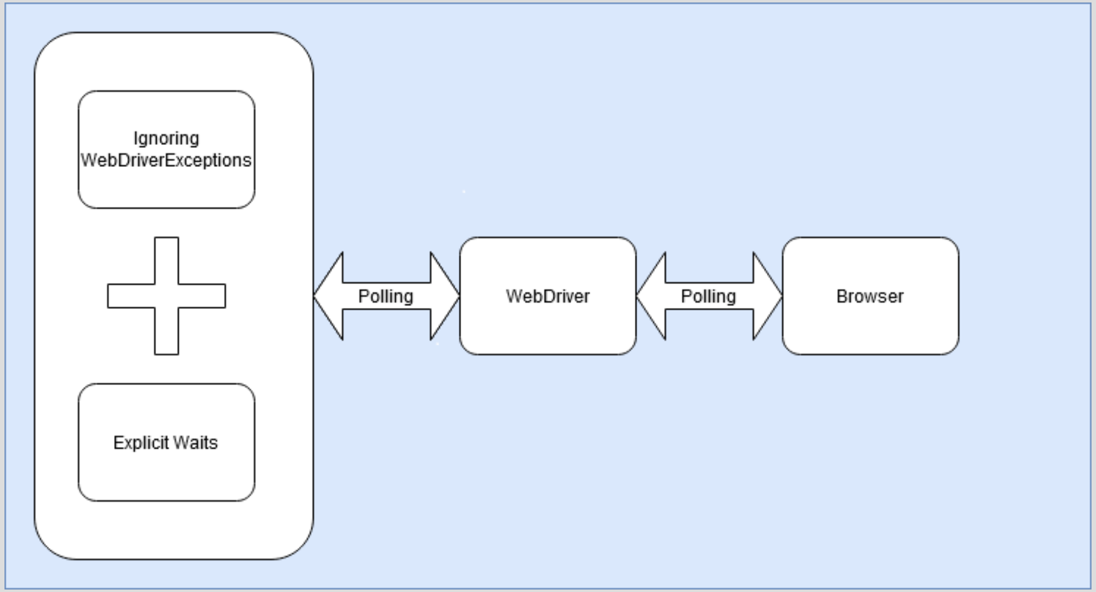
Game Changer

Selenium4

CHROME DEV TOOL SUPPORT(CHROME ONLY)

- Simulating Device Mode
- Simulate Network Speed-2G, 3G, 4G speeds etc
- Mocking Geolocation
- HTTP Requests
- Access Console Log
- Basic Authentication

FLUENT WAIT


Ignoring WebDriverExceptions
Explicit Waits
Polling
WebDriver
Polling
Browser