Selenium-4

What's coming our way?

- Hitesh Prajapati

AGENDA

- ❑ Introduction
- ❑ Selenium 4 Features
  - Relative Locators (Friendly Locators Prev.)
  - Capturing Element Screenshot
  - Windows/Tab Handling
  - Chrome Dev Tools
  - Selenium Grid
- ❑ Transition from 3.X to 4.X
- ❑ Q & A

INTRODUCTION

Java

Se

Protractor
end to end testing for Angular

TS

pmc

appium

JS

Cucumber

C#

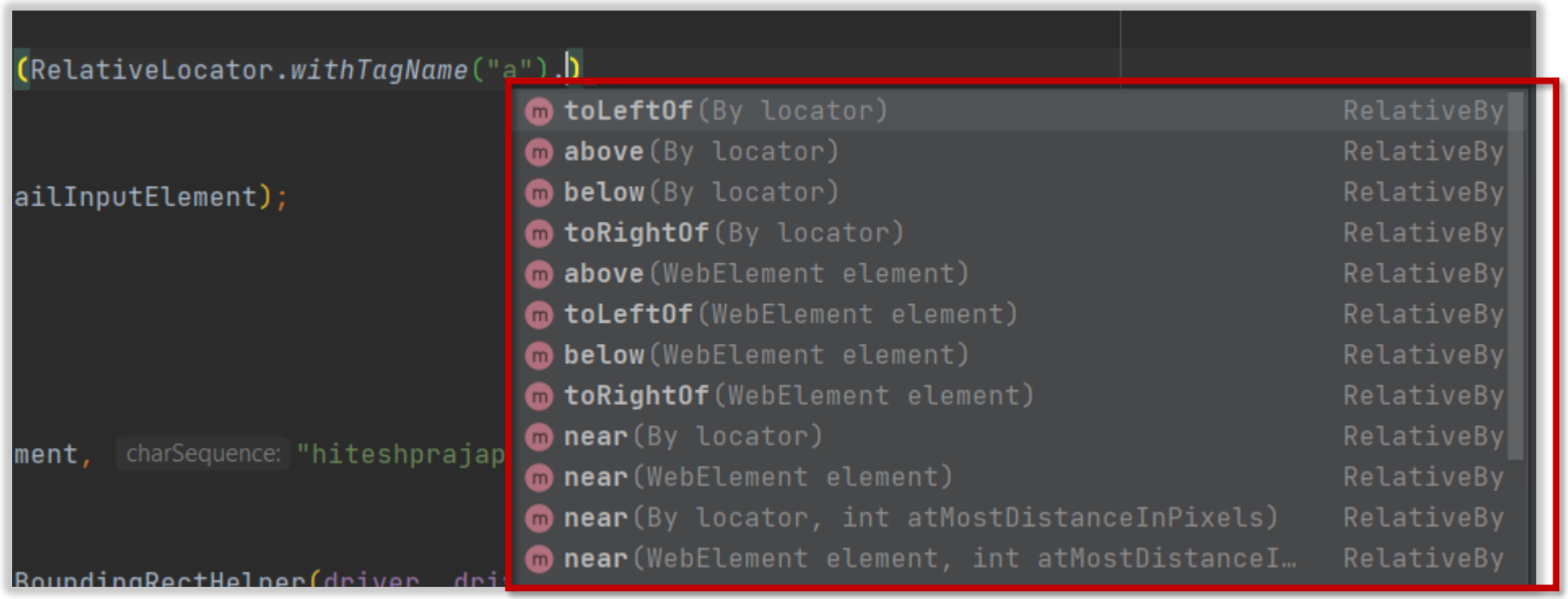
REST-assured

SELENIUM 4 : WHAT'S NEW?

- ❑ Relative Locators (Friendly Locators Prev.)
- ❑ Capturing Element Screenshot
- ❑ Windows/Tab Handling
- ❑ Chrome Dev Tools
- ❑ Selenium Grid

RELATIVE LOCATORS

- Above
- Below
- ToLeftOf
- ToRightOf
- Near

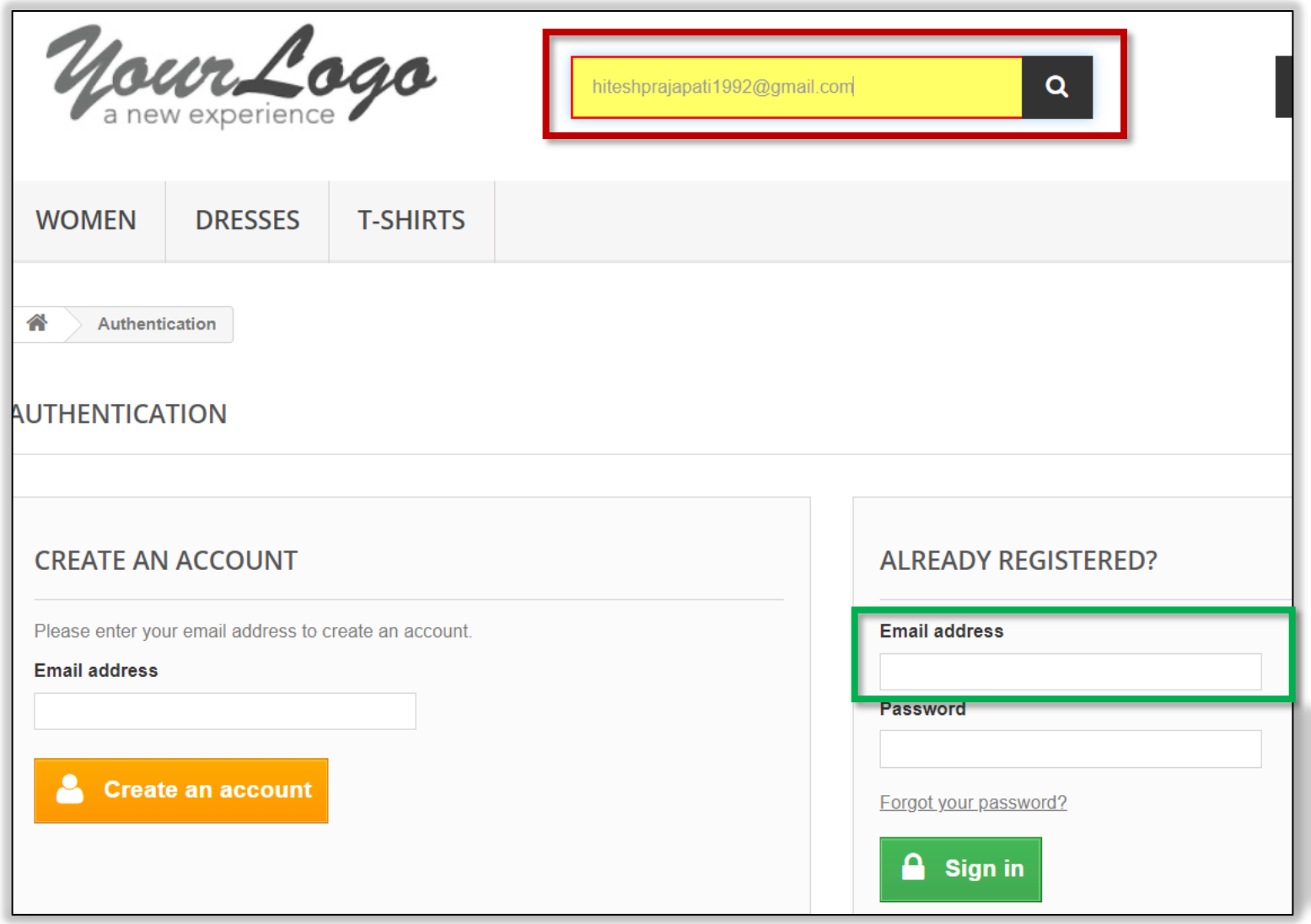


RELATIVE LOCATORS : BAD EXAMPLE

```java
@Test(priority = 2, description = "Locate elements using RelativeLocator strategy. Bad Example")
public void relativeLocatorLoginPageBadExampleTest() {
    // Load URL
    driver.get("http://automationpractice.com/index.php?controller=authentication&back=my-account");

    // Get "Email Address" input element
    WebElement emailInputElement = driver.findElement(RelativeLocator
            .withTagName("input")
            .above(By.id("passwd"))
            .toRightOf(By.id("email_create"))
    );

    // Highlight element and
    JavaScriptHelper.highlightElement(driver, emailInputElement);

    // Wait for 5 sec
    WaitHelper.hardWait( timeInSeconds: 5);

    // Enter Email Address
    ElementHelper.sendKeys(driver, emailInputElement, charSequence: "hiteshprajapati1992@gmail.com");
}
```

Link To Source Code

RELATIVE LOCATORS : BAD EXAMPLE - RESULT

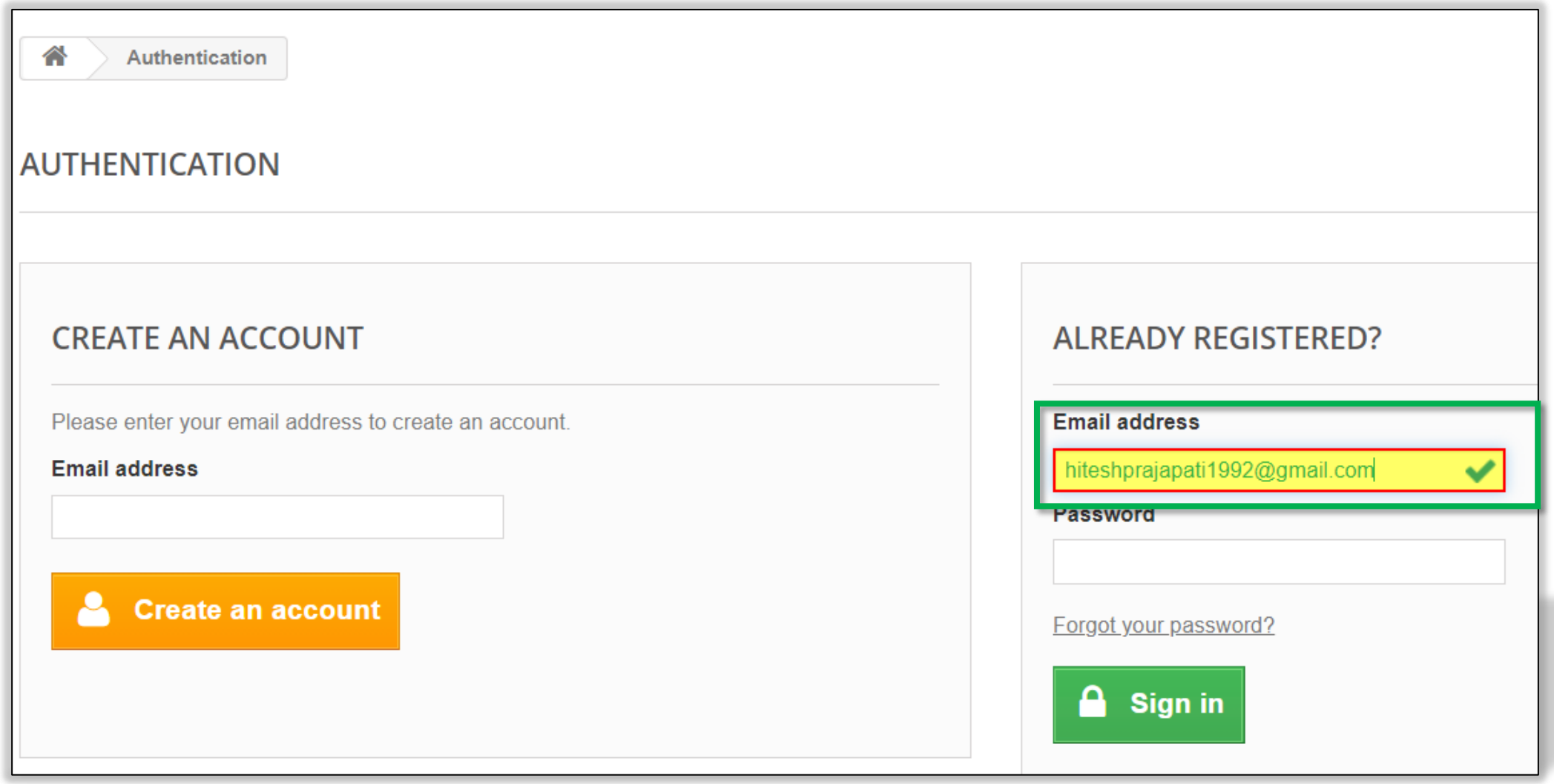


RELATIVE LOCATORS : GOOD EXAMPLE

```java
@Test(priority = 3, description = "Locate elements using RelativeLocator strategy. Good Example")
public void relativeLocatorLoginPageGoodExampleTest() {
    // Load URL
    driver.get("http://automationpractice.com/index.php?controller=authentication&back=my-account");

    // Get "Email Address" input element
    WebElement emailInputElement = driver.findElement(RelativeLocator
            .withTagName("input")
            .above(By.id("passwd"))
            .toRightOf(By.id("email_create"))
            .below(By.id("search_query_top"))
    );

    // Highlight element and
    JavaScriptHelper.highlightElement(driver, emailInputElement);

    // Wait for 5 sec
    WaitHelper.hardWait( timeInSeconds: 5);

    // Enter Email Address
    ElementHelper.sendKeys(driver, emailInputElement,  charSequence: "hiteshprajapati1992@gmail.com");
}
```

Link To Source Code

RELATIVE LOCATORS : GOOD EXAMPLE - RESULT




Demo time

TECH STACK

Java

TestNG

Maven™

RELATIVE LOCATORS : HOW DOES IT WORK?

- ✓ Selenium uses the JavaScript function `getBoundingClientRect()` to find the relative elements
- ✓ The `Element.getBoundingClientRect()` method returns a `DOMRect` object providing information about the size of an element and its position relative to the viewport.
- ✓ The returned DOMRect value contains the `left`, `top`, `right`, `bottom`, `x`, `y`, `width` and `height`.

`RelativeLocator.withTagName().near()`

- ✓ In case of `near()`, it makes use of `getBoundingClientRect()` and tries to locate an element which is at most `50px` away from the specified element.
- ✓ The overloaded method of `near()`, let's you override default behavior of locating an element *(which is `50px`)* by passing pixels value that you want to use.

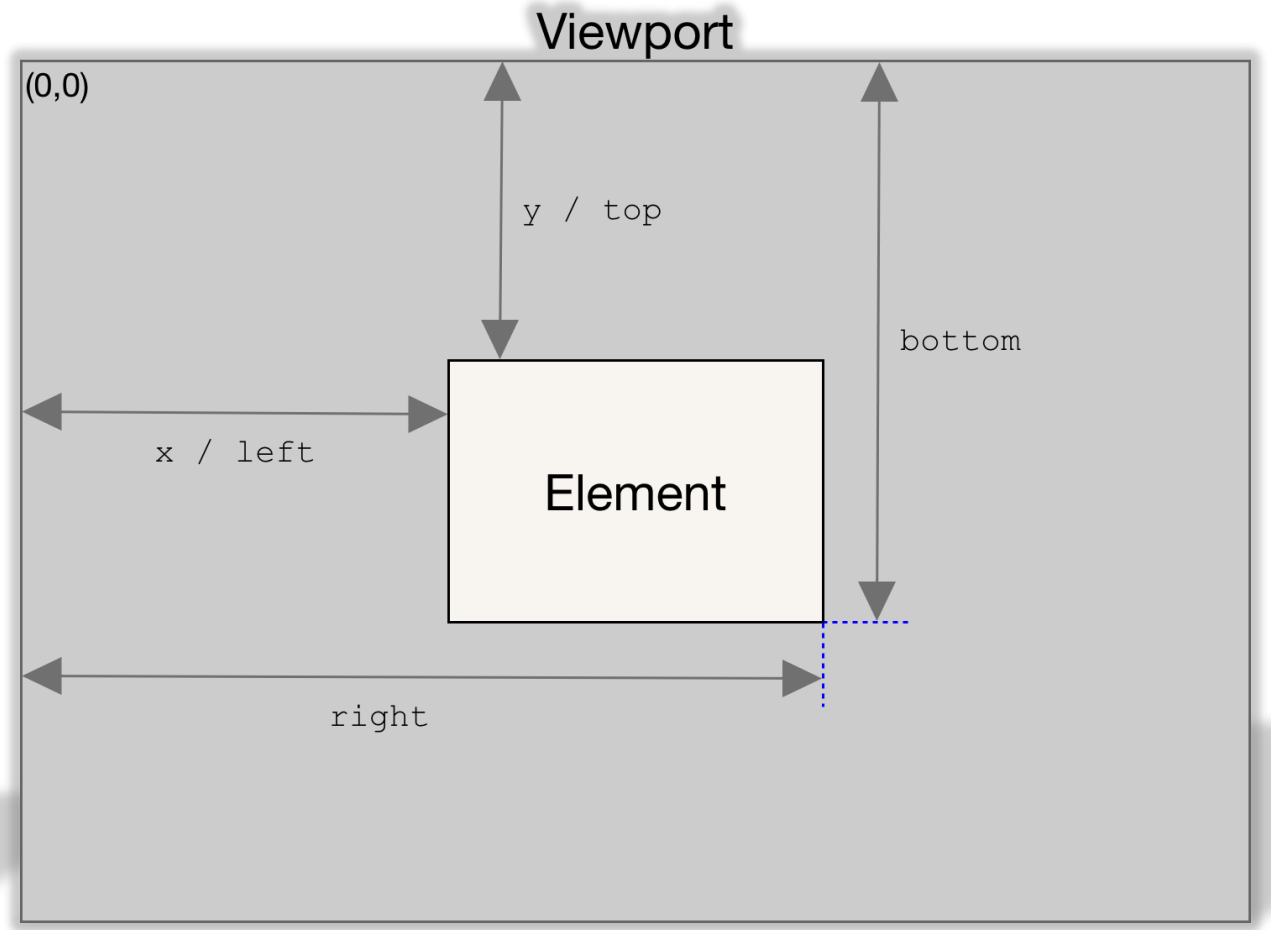
https://www.selenium.dev/documentation/en/getting_started_with_webdriver/locating_elements/

RELATIVE LOCATORS : HOW DOES IT WORK?

Syntax

```
domRect = element.getBoundingClientRect();
```

```
document.getElementById('passwd').getBoundingClientRect();
▼DOMRect {x: 793.6000366210938, y: 307.3999938964844, width:
    bottom: 334.3999938964844
    height: 27
    left: 793.6000366210938
    right: 1064.6000366210938
    top: 307.3999938964844
    width: 271
    x: 793.6000366210938
    y: 307.3999938964844
  ▶__proto__: DOMRect
```



https://developer.mozilla.org/en-US/docs/Web/API/Element/getBoundingClientRect

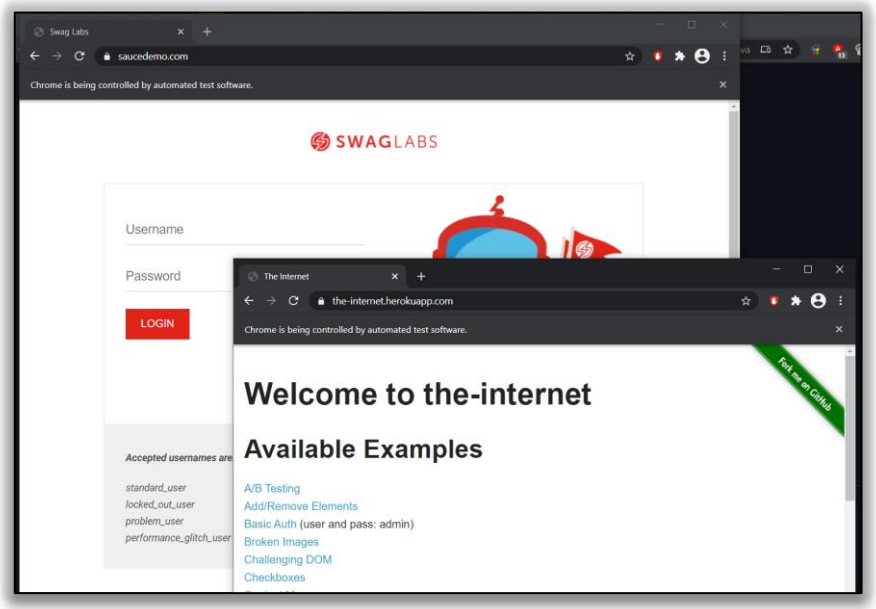
CAPTURING ELEMENT SCREENSHOT

```java
@Test(priority = 0, description = "Capture element screenshot using Selenium 4.")
public void captureElementScreenshotTestOne() throws IOException {
    // Load URL
    driver.get("https://www.saucedemo.com/");

    // Wait for Page to load
    SauceDemoLoginPage.waitForPageToLoad(driver);

    // Get Bot Element
    WebElement sauceBotImageElement = driver.findElement(By.className("bot_column"));

    // Print dimension of Bot - getRect is introduced in Selenium 4.
    Rectangle botRect = sauceBotImageElement.getRect();
    System.out.println("Height x Width : [" + botRect.getHeight() + " x " + botRect.getWidth() + "]");
    System.out.println("(X, Y) : (" + botRect.getX() + ", " + botRect.getY() + ")");

    // Capture Screenshot and store it
    byte[] sauceBotImageAsBytes = sauceBotImageElement.getScreenshotAs(OutputType.BYTES);

    // Build path where to save screenshot
    String ssFilePath = SCREENSHOT_DIR + "Sauce Bot Image - " + DateUtility.getCurrentTimeStamp() + ".png";

    // Save Screenshot
    FileUtils.writeByteArrayToFile(new File(ssFilePath), sauceBotImageAsBytes);
}
```

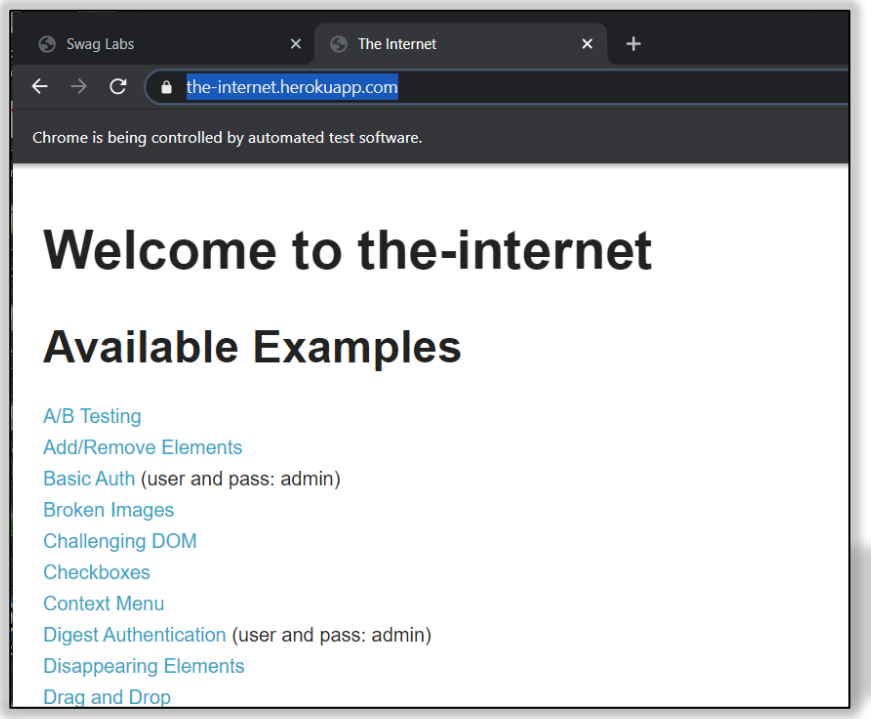
Link To Source Code

```java
@Test(priority = 0, description = "Working with Windows using Selenium 4")
public void windowSwitchTest() {
    // Load URL in 1st window
    driver.get("https://www.saucedemo.com/");

    // Wait for Sauce Login Page to be ready
    SauceDemoLoginPage.waitForPageToLoad(driver);

    // Open 2nd Window
    driver.switchTo().newWindow(WindowType.WINDOW);

    // Load URL in 2nd Window
    driver.get("https://the-internet.herokuapp.com/");
rajapati, 25-01-2021 02:25 PM • initial commit
    // Login
    theInternetLogin(driver);

    // Switch to 1st window
    List<String> windows = new ArrayList<>(driver.getWindowHandles());
    // Print window IDs
    windows.forEach(logger::info);

    // Switch to 1st window
    driver.switchTo().window(windows.get(0));

    // Login To Sauce Page
    SauceDemoLoginPage.usernameElement(driver).sendKeys( ...keysToSend: "standard_user");
    SauceDemoLoginPage.passwordElement(driver).sendKeys( ...keysToSend: "secret_sauce");
    SauceDemoLoginPage.loginButtonElement(driver).click();
}
```

WINDOW HANDLING



TAB HANDLING




```java
@Test(priority = 1, description = "Working with Multiple Tabs Using Selenium 4")
public void tabSwitchTest() {
    // Load URL in 1st window
    driver.get("https://www.saucedemo.com/");

    // Wait for Sauce Login Page to be ready
    SauceDemoLoginPage.waitForPageToLoad(driver);

    // Open 2nd Window
    driver.switchTo().newWindow(WindowType.TAB);

    // Load URL in 2nd Window
    driver.get("https://the-internet.herokuapp.com/");

    // Login
    theInternetLogin(driver);

    // Switch to 1st window
    List<String> windows = new ArrayList<>(driver.getWindowHandles());

    // Print window IDs
    windows.forEach(logger::info);

    // Switch to 1st window
    driver.switchTo().window(windows.get(0));

    // Login To Sauce Page
    SauceDemoLoginPage.usernameElement(driver).sendKeys( ...keysToSend: "standard_user");
    SauceDemoLoginPage.passwordElement(driver).sendKeys( ...keysToSend: "secret_sauce");
    SauceDemoLoginPage.loginButtonElement(driver).click();
}
```

CHROME DEV TOOLS

- Basic Authentication
- Access Console Logs
- Mocking Geo-Location
- Load In-Secure Web Sites
- Block Specific Resources/URLs
- Simulate Network Speed
- Simulate Device Mode
- Changing User Agents
- Capture HTTP Requests
- Simulate Time Zones

WHAT'S CHANGED?

Selenium 3.x

ChromeDriver / EdgeDriver ← RemoteWebDriver

```
public class ChromeDriver extends RemoteWebDriver implements LocationContext, WebStorage, HasTouchScre
    private RemoteLocationContext locationContext;
    private RemoteWebStorage webStorage;
    private TouchScreen touchScreen;
    private RemoteNetworkConnection networkConnection;

    public ChromeDriver() { this(ChromeDriverService.createDefaultService(), new ChromeOptions()); }
```

Selenium 4.x

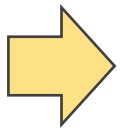
ChromeDriver / EdgeDriver ← ChromiumDriver ← RemoteWebDriver

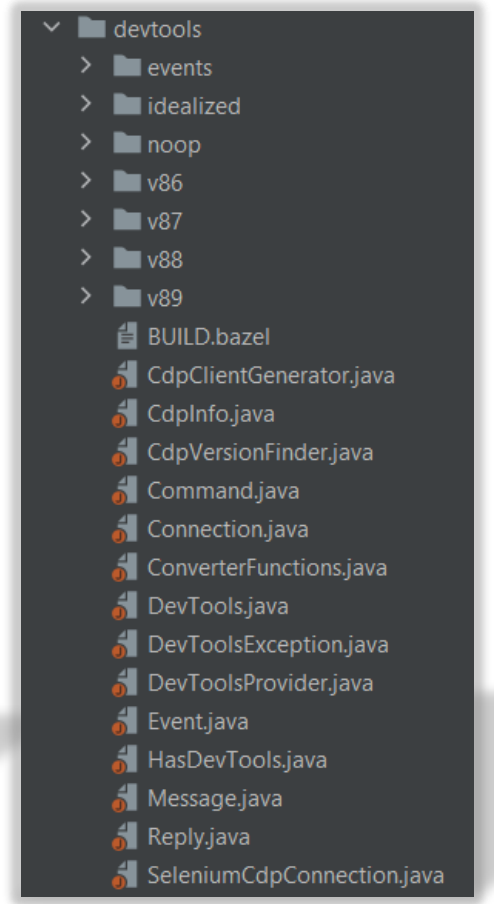
```
public class ChromeDriver extends ChromiumDriver {

  /**
   * Creates a new ChromeDriver using the {@link ChromeDriverService#createDefaultService default}
   * server configuration.
   *
   * @see #ChromeDriver(ChromeDriverService, ChromeOptions)
   */
  public ChromeDriver() { this(ChromeDriverService.createDefaultService(), new ChromeOptions()); }
```

WHAT'S CHANGED?

ChromeDriver ← ChromiumDriver ← RemoteWebDriver

```
public class ChromiumDriver extends RemoteWebDriver implements
  HasAuthentication,
  HasDevTools,
  HasLogEvents,
  HasTouchScreen,
  LocationContext,
  NetworkConnection,
  WebStorage {

  private static final Logger LOG = Logger.getLogger(ChromiumDriver.class.getName());
  private final RemoteLocationContext locationContext;
  private final RemoteWebStorage webStorage;
  private final TouchScreen touchScreen;
  private final RemoteNetworkConnection networkConnection;
  private final Optional<Connection> connection;
  private final Optional<DevTools> devTools;

  protected ChromiumDriver(CommandExecutor commandExecutor, Capabilities capabilities, String capabilityKey) {

  @Override
  public DevTools getDevTools() {
    return devTools.orElseThrow(() -> new WebDriverException("Unable to create DevTools connection"));
  }
```





https://github.com/SeleniumHQ/selenium/tree/trunk/java/client/src/org/openqa/selenium/devtools

CDP : BASIC AUTHENTICATION

Selenium 3.x http://UserName:Password@WebSiteUrl

Selenium 4.x

Network.enable

Enables network tracking, network events will now be delivered to the client.

PARAMETERS

maxTotalBufferSize optional	***integer*** *Buffer size in bytes to use when preserving network payloads (XHRs, etc).* EXPERIMENTAL
maxResourceBufferSize optional	***integer*** *Per-resource buffer size in bytes to use when preserving network payloads (XHRs, etc).* EXPERIMENTAL
maxPostDataSize optional	***integer*** *Longest post body size (in bytes) that would be included in requestWillBeSent notification*

Network.setExtraHTTPHeaders #

Specifies whether to always send extra HTTP headers with the requests from this page.

PARAMETERS

headers	***Headers*** *Map with extra HTTP headers.*

```java
@Test(priority = 1, description = "Basic Authentication Using Chrome Dev Tools")
public void basicAuthenticationUsingCDPTest() {
    // Get Dev Tools
    DevTools devTools = getDevTools();

    // Create CDP Session
    devTools.createSession();

    // Enable Network
    devTools.send(Network.enable(Optional.empty(), Optional.empty(), Optional.empty()));

    // Send Authentication Header
    Map<String, Object> requestHeaderMap = new LinkedHashMap<>();

    // Create Basic Authentication String to pass it as Header With Request
    String basicAuthToken = "Basic " +
            new String(Base64.encode(String.format("%s:%s", "guest", "guest").getBytes()));

    // Set value of Authorization to basic auth token
    requestHeaderMap.putIfAbsent("Authorization", basicAuthToken);

    // Send Request header using CDP
    devTools.send(Network.setExtraHTTPHeaders(new Headers(requestHeaderMap)));

    // Load URL
    driver.get("https://jigsaw.w3.org/HTTP/Basic/");
```

Chrome DevTools Protocol : Network Domain

CDP : ACCESS CONSOLE LOGS

Log Domain

Provides access to log entries.

Methods

Log.clear
Log.disable
Log.enable
Log.startViolationsReport
Log.stopViolationsReport

Events

Log.entryAdded

Types

Log.LogEntry
Log.ViolationSetting

Log.LogEntry

Log entry.

Type: ***object***

PROPERTIES

source	***string*** *Log entry source.* *Allowed Values:* xml, javascript, network, storage, appcache, rendering, security, deprecation, worker, violation, intervention, recommendation, other
level	***string*** *Log entry severity.* *Allowed Values:* verbose, info, warning, error
text	***string*** *Logged text.*
timestamp	***Runtime.Timestamp*** *Timestamp when this entry was added.*
url optional	***string*** *URL of the resource if known.*
lineNumber optional	***integer*** *Line number in the resource.*
stackTrace optional	***Runtime.StackTrace*** *JavaScript stack trace.*
networkRequestId optional	***Network.RequestId*** *Identifier of the network request associated with this entry.*
workerId optional	***string*** *Identifier of the worker associated with this entry.*
args optional	***array[Runtime.RemoteObject]*** *Call arguments.*

```java
@Test(priority = 0, description = "Get console logs using Chrome Dev Tools")
public void amazonGetConsoleLogUsingCDPTest() {
    // Create Dev Tools
    DevTools chromeDevTools = ((ChromeDriver) driver).getDevTools();

    // Create Chrome Dev Tools Session
    chromeDevTools.createSession();

    // Enable Console Logs
    chromeDevTools.send(Log.enable());

    // Create List to hold the values of Console LogEntries
    List<LogEntry> logEntries = new ArrayList<>();

    // Add A listener to listen when an Entry is added. Add the entry to List
    chromeDevTools.addListener(Log.entryAdded(), logEntries::add);

    // Load URL
    driver.get("https://www.ebay.com/");
```

Chrome DevTools Protocol : Log Domain

CDP : MOCK GEO-LOCATION

Emulation.setGeolocationOverride

Overrides the Geolocation Position or Error. Omitting any of the parameters emulates position unavailable.

PARAMETERS

`latitude`	***number***	*Mock latitude*
optional		
`longitude`	***number***	*Mock longitude*
optional		
`accuracy`	***number***	*Mock accuracy*
optional		

```
@Test(priority = 1, description = "Mock geo-location using Chrome Dev Tools", dataProvider = "getLatLong")
public void mockGeoLocationUsingCDPTest(String location, Number latitude, Number longitude, Number accuracy) {
    // Get Chrome Dev Tools
    DevTools chromeDevTools = getDevTools();

    // Create Session
    chromeDevTools.createSession();

    // Emulate Geo Location  Prajapati, 25-01-2021 11:24 PM • GeoLocationTests: Test Cases added for Mocking Geo
    chromeDevTools.send(Emulation.setGeolocationOverride(
            Optional.of(latitude),
            Optional.of(longitude),
            Optional.of(accuracy)
    ));

    // Load URL
    driver.get("https://whatmylocation.com/");
```

Chrome DevTools Protocol : Emulation Domain

CDP : LOAD INSECURE WEBSITES

Security Domain

Security

Methods

Security.disable
Security.enable
Security.handleCertificateError DEPRECATED
Security.setOverrideCertificateErrors DEPRECATED
Security.setIgnoreCertificateErrors EXPERIMENTAL

Events

Security.securityStateChanged
Security.certificateError DEPRECATED
Security.visibleSecurityStateChanged EXPERIMENTAL

Security.setIgnoreCertificateErrors
EXPERIMENTAL

Enable/disable whether all certificate errors should be ignored.

PARAMETERS

`ignore` ***boolean***
If true, all certificate errors will be ignored.

```java
@Test(priority = 1, description = "Load Insecure website using Chrome Dev Tools")
public void loadInsecureWebsiteUsingCDPTestOne() {

    // Get Chrome Dev Tools
    DevTools chromeDevTools = getDevTools();

    // Create Session
    chromeDevTools.createSession();

    // Enable Security
    chromeDevTools.send(Security.enable());

    // Set ignore Certificate error to true
    chromeDevTools.send(Security.setIgnoreCertificateErrors(true));

    // Load Url
    driver.get("https://expired.badssl.com/");
```

Chrome DevTools Protocol : Security Domain

CDP : BLOCK SPECIFIC RESOURCE / URL

Network.setBlockedURLs EXPERIMENTAL

Blocks URLs from loading.

PARAMETERS

`urls` ***array[string]***
URL patterns to block. Wildcards ('') are allowed.*

```java
@Test(priority = 0, description = "Block Images Using Chrome Dev Tools")
public void blockImagesUsingCDPTest() {
    // Get Dev Tools
    DevTools chromeDevTools = getDevTools();

    // Create Session
    chromeDevTools.createSession();

    // Enable network
    chromeDevTools.send(Network.enable(Optional.empty(), Optional.empty(), Optional.empty()));

    // Block Specific file types by intercepting network
    chromeDevTools.send(Network
            .setBlockedURLs(ImmutableList.of(
                    "*.jpg",
                    "*.png",
                    "*.jpeg",
                    "*.svg",
                    "*.gif"))
    );

    // Load URl
    driver.get("https://www.amazon.in/");
```

Chrome DevTools Protocol : Network Domain

CDP : SIMULATE NETWORK SPEED

Network.emulateNetworkConditions

Activates emulation of network conditions.

PARAMETERS

offline	***boolean*** *True to emulate internet disconnection.*
latency	***number*** *Minimum latency from request sent to response headers received (ms).*
downloadThroughput	***number*** *Maximal aggregated download throughput (bytes/sec). -1 disables download throttling.*
uploadThroughput	***number*** *Maximal aggregated upload throughput (bytes/sec). -1 disables upload throttling.*
connectionType optional	***ConnectionType*** *Connection type if known.*

```java
@Test(priority = 0, dataProvider = "getNetworkSpeeds")
public void phpTravelsNetworkSimulationOne(boolean disableNetwork, int latency, int download, int upload) {
    DevTools chromeDevTools = ((ChromeDriver) driver).getDevTools();

    // Create session
    chromeDevTools.createSession();

    // Enables network tracking, network events will now be delivered to the client.
    chromeDevTools.send(Network.enable(Optional.of(1000000), Optional.empty(), Optional.empty()));

    // Emulate Network Conditions
    chromeDevTools.send(Network.emulateNetworkConditions(
            disableNetwork,
            latency,
            download,
            upload,
            Optional.of(ConnectionType.WIFI)
    ));

    List<String> networkLogs = new ArrayList<>();

    chromeDevTools.addListener(Network.loadingFailed(), loadingFailed -> {
        networkLogs.add(loadingFailed.getErrorText());
    });

    StopWatch stopWatch = new StopWatch();
    stopWatch.start();

    try {
        // Load URL
        driver.get("https://www.phptravels.net/");
    } catch (WebDriverException ignored) {}
```

Chrome DevTools Protocol : Network Domain

CDP : SIMULATE DEVICE MODE

Emulation.setDeviceMetricsOverride #

Overrides the values of device screen dimensions (window.screen.width, window.screen.height, window.innerWidth, window.innerHeight, and "device-width"/"device-height"-related CSS media query results).

PARAMETERS

width	***integer*** *Overriding width value in pixels (minimum 0, maximum 10000000). 0 disables the override.*
height	***integer*** *Overriding height value in pixels (minimum 0, maximum 10000000). 0 disables the override.*
deviceScaleFactor	***number*** *Overriding device scale factor value. 0 disables the override.*
mobile	***boolean*** *Whether to emulate mobile device. This includes viewport meta tag, overlay scrollbars, text autosizing and more.*
scale optional	***number*** *Scale to apply to resulting view image.* EXPERIMENTAL
screenWidth optional	***integer*** *Overriding screen width value in pixels (minimum 0, maximum 10000000).* EXPERIMENTAL
screenHeight optional	***integer*** *Overriding screen height value in pixels (minimum 0, maximum 10000000).* EXPERIMENTAL
positionX optional	***integer*** *Overriding view X position on screen in pixels (minimum 0, maximum 10000000).* EXPERIMENTAL
positionY optional	***integer*** *Overriding view Y position on screen in pixels (minimum 0, maximum 10000000).* EXPERIMENTAL
dontSetVisibleSize optional	***boolean*** *Do not set visible view size, rely upon explicit setVisibleSize call.* EXPERIMENTAL
screenOrientation optional	***ScreenOrientation*** *Screen orientation override.*
viewport optional	***Page.Viewport*** *If set, the visible area of the page will be overridden to this viewport. This viewport change is not observed by the page, e.g. viewport-relative elements do not change positions.* EXPERIMENTAL
displayFeature optional	***DisplayFeature*** *If set, the display feature of a multi-segment screen. If not set, multi-segment support is turned-off.* EXPERIMENTAL

```java
@Test(priority = 0, description = "Enable simulation of a device using CDP's send method")
public void phpTravelsSimulatingDeviceTestOne() {

    // Get Dev Tools
    DevTools devTools = ((ChromeDriver) driver).getDevTools();

    // Create Session
    devTools.createSession();

    // Set Simulating device details
    devTools.send(Emulation.setDeviceMetricsOverride(
            width: 320,
            height: 556,
            deviceScaleFactor: 0.5,
            mobile: true,
            Optional.empty(),
            Optional.empty(),
            Optional.empty(),
            Optional.empty(),
            Optional.empty(),
            Optional.empty(),
            Optional.of(new ScreenOrientation(ScreenOrientation.Type.LANDSCAPEPRIMARY,  angle: 90)),
            Optional.empty(),
            Optional.empty())
    );

    // load url
    driver.get("https://www.phptravels.net/");
```

Chrome DevTools Protocol : Emulation Domain

CDP : CHANGING USER AGENTS

What is User Agent?

The user agent string contains information about the user's web browser name, operating system, device type, and other information.

Take this UA string as an example :

```
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_2) AppleWebKit/537.36  (KHTML, like Gecko) Chrome/51.0.2704.84 Safari/537.36
```

The user agent application - is Mozilla version 5.0, or a piece of software compatible with it.

The operating system - is OS X version 10.2.2 (and is running on a Mac).

The client - is Chrome version 51.0.2704.84 and is based on Safari version 537.36.

The engine - responsible for displaying content on this device is AppleWebKit version 537.36 (and KHTML, an open-source layout engine, is present too).

https://www.whoishostingthis.com/tools/user-agent/

CDP : CHANGING USER AGENTS

Network.setUserAgentOverride #

Allows overriding user agent with the given string.

PARAMETERS

`userAgent`	***string*** *User agent to use.*
`acceptLanguage` optional	***string*** *Browser langugage to emulate.*
`platform` optional	***string*** *The platform navigator.platform should return.*
`userAgentMetadata` optional	***Emulation.UserAgentMetadata*** *To be sent in Sec-CH-UA-* headers and returned in navigator.userAgentData* EXPERIMENTAL

```
@Test(priority = 0, description = "To Change the User Agent of browser", dataProvider = "userAgents")
public void phpTravelsUserAgentSimulationTestOne(String uAOf, String userAgentString) throws Interrupt

    // Get Dev Tools Object
    DevTools chromeDevTools = getDevTools();

    // Create a Dev Tool Session
    chromeDevTools.createSession();

    // Enable Network simulation
    chromeDevTools.send(Network.enable(Optional.empty(), Optional.empty(), Optional.empty()));

    String appleIphoneXRSafari = "Mozilla/5.0 " +
            "(iPhone; CPU iPhone OS 12_0 like Mac OS X) " +
            "AppleWebKit/605.1.15 (KHTML, like Gecko) " +
            "Version/12.0 Mobile/15E148 Safari/604.1";

    // Simulate User Agent
    chromeDevTools.send(Network.setUserAgentOverride(
            appleIphoneXRSafari,
            Optional.empty(),
            Optional.empty(),
            Optional.empty()
    ));

    // Load url
    driver.get("https://gs.statcounter.com/detect");
```

User Agent String

Chrome DevTools Protocol : Network Domain

CDP : CAPTURE HTTP REQUESTS

Network.requestWillBeSent

Fired when page is about to send HTTP request.

PARAMETERS

requestId	***RequestId*** *Request identifier.*
loaderId	***LoaderId*** *Loader identifier. Empty string if the request is fetched from worker.*
documentURL	***string*** *URL of the document this request is loaded for.*
request	***Request*** *Request data.*
timestamp	***MonotonicTime*** *Timestamp.*
wallTime	***TimeSinceEpoch*** *Timestamp.*
initiator	***Initiator*** *Request initiator.*
redirectResponse optional	***Response*** *Redirect response data.*
type optional	***ResourceType*** *Type of this resource.*
frameId optional	***Page.FrameId*** *Frame identifier.*
hasUserGesture optional	***boolean*** *Whether the request is initiated by a user gesture. Defaults to false.*

Network.responseReceived

Fired when HTTP response is available.

PARAMETERS

requestId	***RequestId*** *Request identifier.*
loaderId	***LoaderId*** *Loader identifier. Empty string if the request is fetched from worker.*
timestamp	***MonotonicTime*** *Timestamp.*
type	***ResourceType*** *Resource type.*
response	***Response*** *Response data.*
frameId optional	***Page.FrameId*** *Frame identifier.*

```java
@Test(priority = 0, description = "Capture HTTP Requests using Chrome Dev Tools")
public void captureHTTPRequestsUsingCDPTestOne() {
    // Create Dev Tools Object
    DevTools devTools = getDevTools();

    // Create Session
    devTools.createSession();

    // Enable Network
    devTools.send(Network.enable(Optional.empty(), Optional.empty(), Optional.empty()));

    JSONArray capturedRequestJsonArray = new JSONArray();
    JSONArray responseReceivedJsonArray = new JSONArray();

    List<RequestWillBeSent> sentNetworkLogs = new ArrayList<>();
    List<ResponseReceived> receivedNetworkLogs = new ArrayList<>();

    // Listen to network for Requests being sent
    devTools.addListener(Network.requestWillBeSent(), requestWillBeSent -> {...});

    // Listen to Network for Responses being received
    devTools.addListener(Network.responseReceived(), responseReceived -> {...});

    // Load URL
    driver.get("https://www.amazon.in/");   Prajapati, 27-01-2021 06:18 PM • CaptureHTTPRequestT
```

Chrome DevTools Protocol : Network Domain

CDP : SIMULATE TIME-ZONE

Emulation.setTimezoneOverride EXPERIMENTAL

Overrides default host system timezone with the specified one.

PARAMETERS

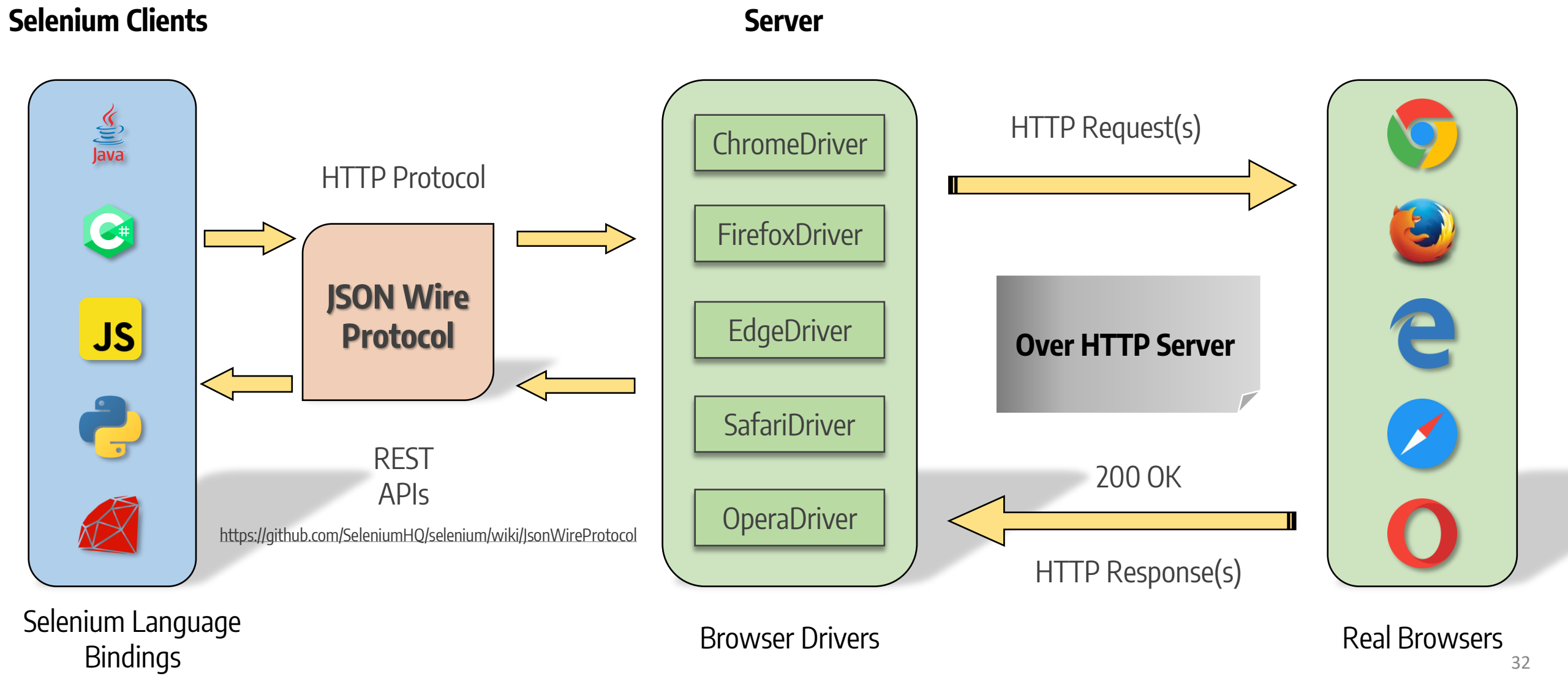
timezoneId	***string*** *The timezone identifier. If empty, disables the override and restores default host system timezone.*

```java
@Test(priority = 1, description = "Simulate TimeZone using Chrome Dev Tools")
public void simulateTimeZoneUsingCDPTest() {
    // Get Dev Tools
    DevTools chromeDevTools = getDevTools();

    // Create Session
    chromeDevTools.createSession();

    // Simulate Time Zone using TimeZone Id -> This Switch is still Experimental
    // Get TimeZone Ids Below
    // https://docs.oracle.com/middleware/12212/wcs/tag-ref/MISC/TimeZones.html
    chromeDevTools.send(Emulation.setTimezoneOverride("Europe/London"));

    // Load URL
    driver.get("https://webbrowsertools.com/timezone/");
```

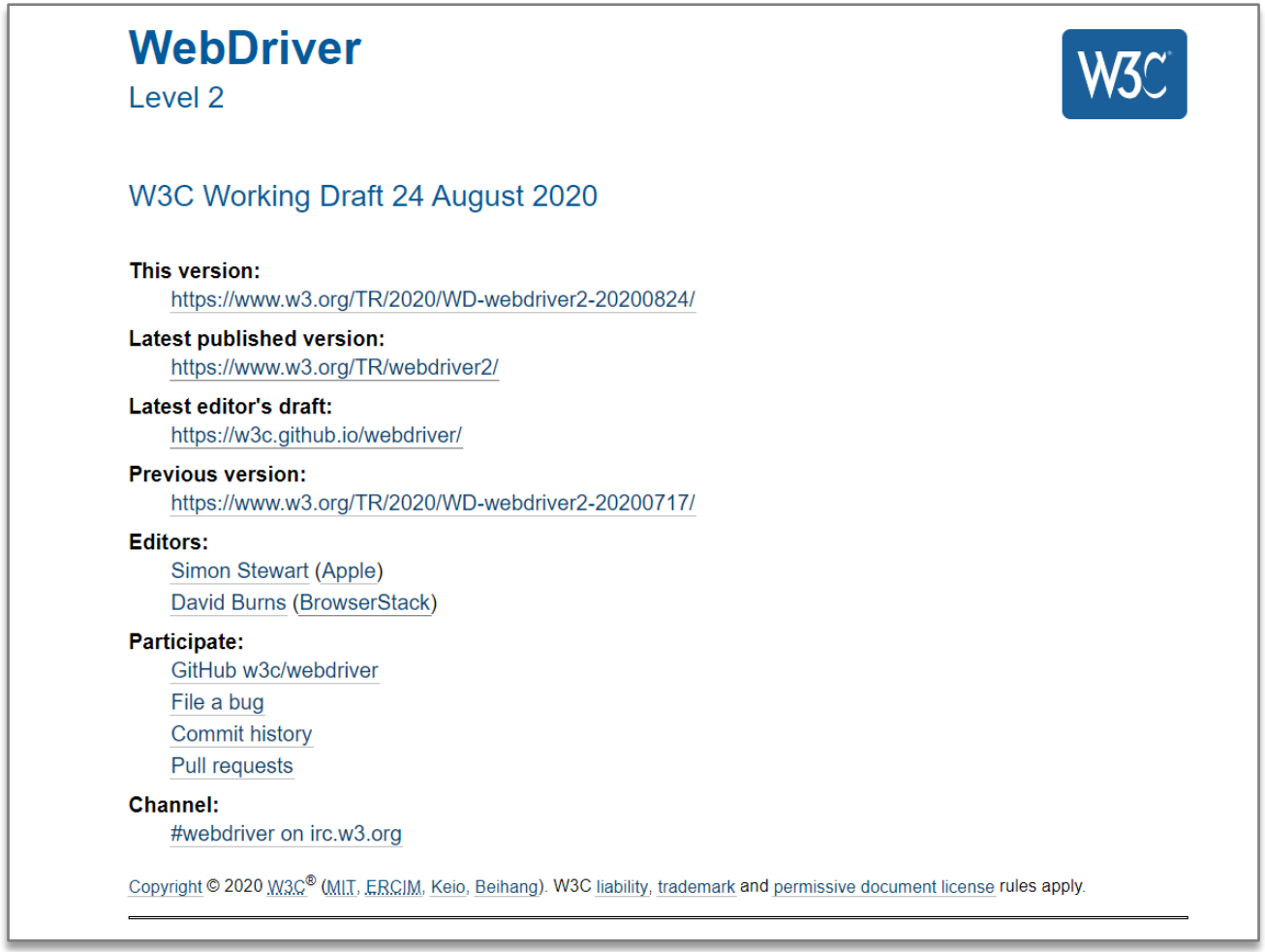
Chrome DevTools Protocol : Emulation Domain

TRANSITION FROM 3.X TO 4.X

SELENIUM 3 ARCHITECTURE



W3C WEB-DRIVER SPECIFICATION



WebDriver
Level 2

W3C Working Draft 24 August 2020

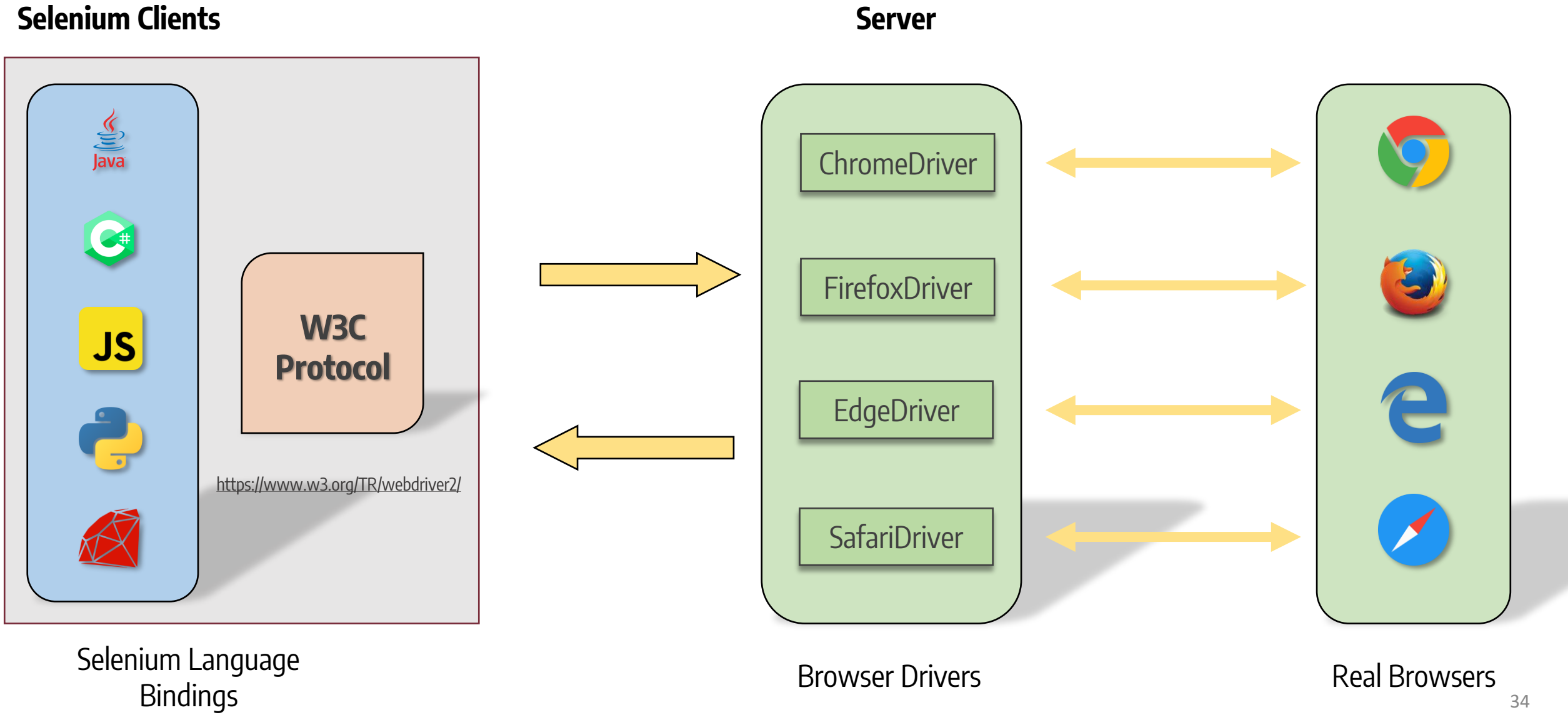
This version:
https://www.w3.org/TR/2020/WD-webdriver2-20200824/
Latest published version:
https://www.w3.org/TR/webdriver2/
Latest editor's draft:
https://w3c.github.io/webdriver/
Previous version:
https://www.w3.org/TR/2020/WD-webdriver2-20200717/
Editors:
Simon Stewart (Apple)
David Burns (BrowserStack)
Participate:
GitHub w3c/webdriver
File a bug
Commit history
Pull requests
Channel:
#webdriver on irc.w3.org

- ❑ **What is W3C ?**
- ❑ **Why W3C?**
- ❑ **What is W3C WebDriver Specification?**

Selenium W3C WebDriver protocol defines the ways that browser actions are executed. WebDriver is a remote control interface that enables introspection and control of user agents. It provides a platform and language-neutral wire protocol as a way for out-of-process programs to remotely instruct the behavior of web browsers.

https://www.w3.org/TR/webdriver2/

SELENIUM 4 ARCHITECTURE



TRANSITION FROM 3.X TO 4.X : WHAT'S CHANGED?

	< 3.8	>= 3.8.0 to 3.141.59	>= 4.X
Protocol	{JSON} Wire Protocol Over HTTP	{JSON} Wire Protocol Over HTTP + W3C	W3C
ChromeDriver	< v75.x	>= v75.x	>= v75.x
GeckoDriver	< v47.x	< v47.x (SE < v3.11) / >= v47.x (SE >= v3.11)	>= v47.x

```
ChromeDriver was started successfully.
Feb 01, 2021 1:45:37 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
```

https://firefox-source-docs.mozilla.org/testing/geckodriver/Usage.html

SELENIUM 4.0 : WEB-DRIVER TEST RESULTS

Result Summary

Driver Type	Browser Version	Driver Version	Total Tests	Skip	Pass	Fail	Pass Rate	Duration (sec)
IE	Microsoft Internet Explorer (11.1000.18936.0 [11.0.130])	IEDriverServer.exe (3.141.59.2)	995	86	904	5	99.45%	1616.805
Firefox	Mozilla Firefox Release (68.0)	geckodriver.exe (0.24)	995	82	890	23	97.48%	945.865
Firefox Nightly	Mozilla Firefox Nightly (70.0a1 [2019-07-11])	geckodriver.exe (0.24)	995	82	890	23	97.48%	940.483
Chrome Stable	Google Chrome (75.0.3770.100)	chromedriver.exe (75.0.3770.90)	995	48	943	4	99.58%	1366.967
Chrome Dev	Google Chrome (77.0.3833.0)	chromedriver.exe (77.0.3851.0)	995	48	944	3	99.68%	1433.096
Edge (Chromium)	Microsoft Edge (77.0.211.3)	MSEdgeDriver.exe (77.0.217.0)	995	48	941	6	99.37%	1331.926
Edge (Legacy)	Microsoft Edge (44.18936.1000.0/EdgeHTML 18.18936)	MicrosoftWebDriver.exe (18936)	995	100	852	43	95.20%	806.200
Safari	Apple Safari (12.1.1, WebKit 14607.2.6.1.1)	safaridriver (12.1.1)	995	80	756	159	82.62%	275.661
Safari Tech Preview	Apple Safari Technology Preview Release 87 (Safari 13.0, WebKit 14608.1.33.1)	safaridriver (13)	995	80	760	155	83.06%	325.686

Notes

- Tests were executed on 12 July 2019
- Tests were run from the .NET bindings tests of the Selenium project master branch. Tests and language bindings code current as of commit 9ec49e496ed1527376ddee2b5282bb0e0e80e79f
- Except for Safari, tests were run on Windows 10, Insider build 18936
- Safari tests were run on macOS Mojave (10.14.5)
- Tests are no longer tracking Google Chrome using the OSS protocol dialect, as Chrome 75 moved to the W3C dialect by default
- Tests are now tracking the Chromium-based Microsoft Edge, using its Dev Channel, and EdgeHTML-based Edge is now referred to as Edge (Legacy)

http://webdriver-herald.herokuapp.com/

ADVANTAGES OF W3C - WD

✓ Standards

- Your tests should run more consistently between browsers because this is a standard that all browser vendors will develop. This means common code for all browsers.

✓ Stability

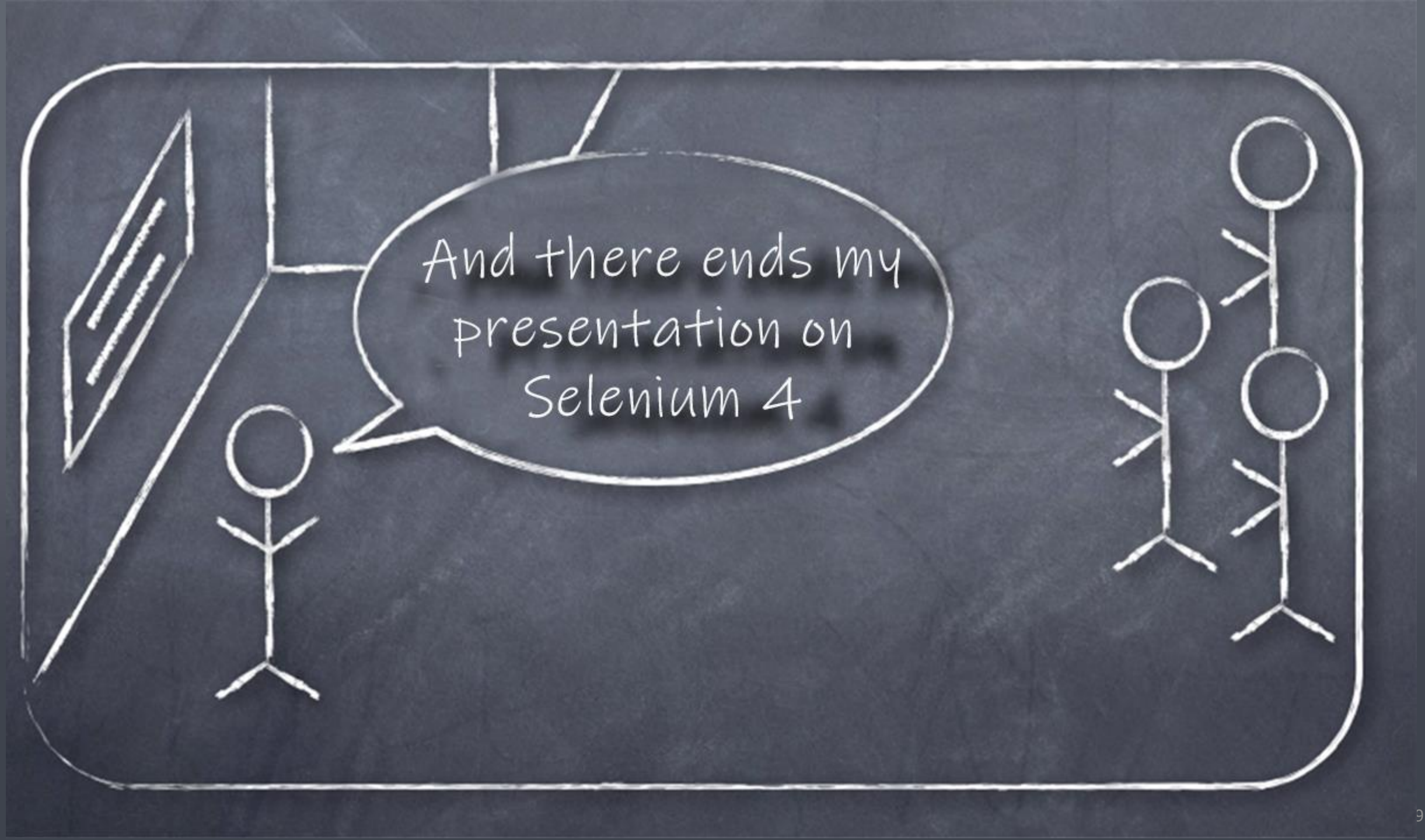
- More stable test case execution
- Less number of exceptions while executing tests across various browsers.
  - StaleElementException
  - ElementNotInteractableException
  - ElementClickInterceptedException, etc....

TRANSITION FROM 3.X TO 4.X : ACTION API

```
package
org.openqa.selenium.interactions
```

	< v4.x		>= v4.x
Click	`.moveToElement(WebElement).click()`	+	`.click(WebElement)`
Click And Hold	`.moveToElement(WebElement).clickAndHold()`	+	`.clickAndHold(WebElement)`
Context Click	`.moveToElement(WebElement).contextClick()`	+	`.contextClick(WebElement)`
Double Click	`.moveToElement(WebElement).doubleClick()`	+	`.doubleClick(WebElement)`
Release Left Mouse	`.moveToElement(WebElement).release()`	+	`.release(WebElement)`

https://www.selenium.dev/selenium/docs/api/java/index.html?deprecated-list.html


And there ends my presentation on Selenium 4


Q
A


ANY
QUESTIONS???
quickmeme.com

REFERENCES

Demo Source Code : https://github.com/HItesh007/selenium4-demo

Relative Locator : https://www.selenium.dev/documentation/en/getting_started_with_webdriver/locating_elements/

Bounding Client Doc : https://developer.mozilla.org/en-US/docs/Web/API/Element/getBoundingClientRect

JSON Wire Protocol : https://github.com/SeleniumHQ/selenium/wiki/JsonWireProtocol

W3C WebDriver Specification : https://www.w3.org/TR/webdriver2/

Selenium 4 - WD Test Results : http://webdriver-herald.herokuapp.com/

Selenium 4 – Deprecated List : https://www.selenium.dev/selenium/docs/api/java/index.html?deprecated-list.html

User Agent : https://www.whoishostingthis.com/tools/user-agent/

Explore User Agent String : https://developers.whatismybrowser.com/useragents/explore/

Chrome Dev Tools Documentation

Network Domain : https://chromedevtools.github.io/devtools-protocol/tot/Network/

Log Domain : https://chromedevtools.github.io/devtools-protocol/tot/Log/

Emulation Domain : https://chromedevtools.github.io/devtools-protocol/tot/Emulation/

Security Domain : https://chromedevtools.github.io/devtools-protocol/tot/Security/

THANK YOU

HITESH PRAJAPATI

hiteshprajapati1992@gmail.com

/in/hiteshprajapati007/

/HItesh007

/Hitesh_prajapati1992