How to use Chrome for Testing in Katalon Studio

🧐 What is Chrome for Testing?

A new Chrome flavor that specifically targets web app testing and automation use cases. Chrome for Testing has been created purely for browser automation and testing purposes, and is not suitable for daily browsing.

❓ Why do we need Chrome for Testing?

One of Chrome's most notable features is its ability to auto-update. **Users** are happy to know they're running an up-to-date and secure browser version including modern Web Platform features, browser features, and bug fixes at all times.

However, as a **developer** running a suite of end-to-end tests you might have an entirely different perspective:

- You want consistent, reproducible results across repeated test runs—but this may not happen if the browser executable or binary decides to update itself in between two runs.
- You want to pin a specific browser version and check that version number into your source code repository, so that you can check out old commits and branches and re-run the tests against the browser binary from that point in time.

None of this is possible with an auto-updating browser binary. As a result, you may not want to use your regular Chrome installation for automated testing. This is the fundamental mismatch between what's good for regular browser users versus what's good for developers doing automated testing.

Instructions

The easiest way to download Chrome for Testing binaries for your platform is by using our `@puppeteer/browsers` command-line utility, available via `npm`. Here are some examples:

```
# Download the latest available Chrome for Testing binary corresponding to the Stable channel.
npx @puppeteer/browsers install chrome@stable

# Download a specific Chrome for Testing version.
npx @puppeteer/browsers install chrome@116.0.5793.0

# Download the latest available ChromeDriver version corresponding to the Canary channel.
npx @puppeteer/browsers install chromedriver@canary

# Download a specific ChromeDriver version.
npx @puppeteer/browsers install chromedriver@116.0.5793.0
```

Precondition:

- Install NodeJS

- Install NPX via npm:

`npm install -g npx`

In my case, I'd like to install Chrome v.116

- Open Terminal/CMD Windows and Enter the below command lines:

`npx @puppeteer/browsers install chromedriver@116.0.5793.0`

```
vi.kim@Liam ~ % npx @puppeteer/browsers install chromedriver@116.0.5793.0

Downloading chromedriver r116.0.5793.0 - 8.4 MB [====================] 100% 0.0s
chromedriver@116.0.5793.0 /Users/vi.kim/chromedriver/mac_arm-116.0.5793.0/chrome
driver-mac-arm64/chromedriver
```

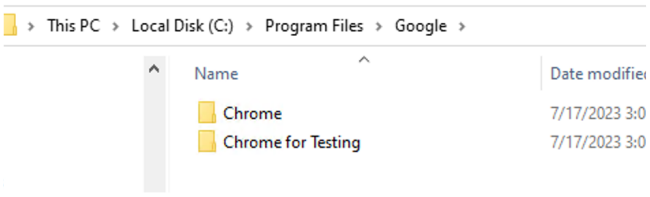
`npx @puppeteer/browsers install chrome@116.0.5793.0`

```
vi.kim@Liam ~ % npx @puppeteer/browsers install chrome@116.0.5793.0

Downloading chrome r116.0.5793.0 - 122 MB [====================] 100% 0.0s
chrome@116.0.5793.0 /Users/vi.kim/chrome/mac_arm-116.0.5793.0/chrome-mac-arm64/G
oogle Chrome for Testing.app/Contents/MacOS/Google Chrome for Testing
```

Windows:

- Go to `C:\Users\testing\chrome\win64-116.0.5793.0\chrome-win64\chrome.exe`, copy the entire folder `win64-116.0.5793.0`
- Go to `C:\Program Files\Google` or `C:\Program Files (x86)\Google` (depend on where Google Chrome is installed on your machine) > Create a folder named **Chrome for Testing**



- And paste the `win64-116.0.5793.0` under this folder

macOS:

- Go to `/Users/vi.kim/chrome/mac_arm-116.0.5793.0/chrome-mac-arm64`, drag the `Google Chrome for Testing` and drag to the **Applications** folder

Suggestion 1: Specify Chrome binary and ChromeDriver in Project

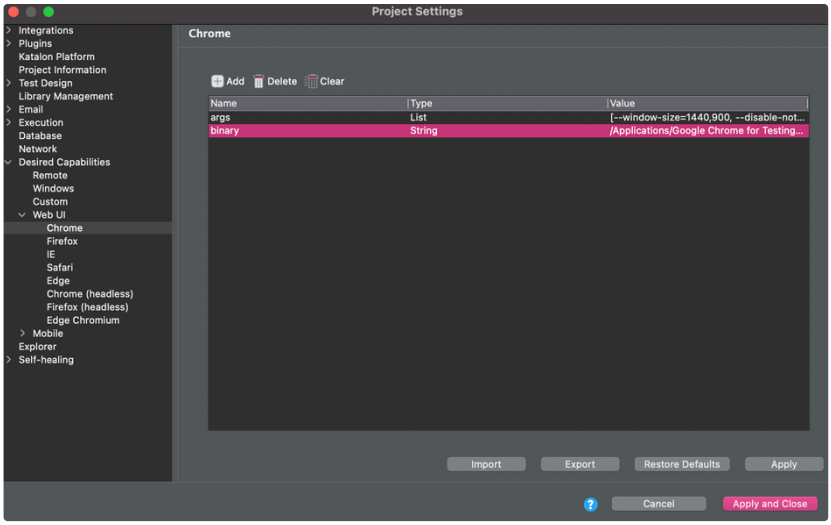
- Open **Katalon Studio** > **Project Settings** > **Desired Capabilities** > **Web UI** > **Chrome** > Add a property:

Windows

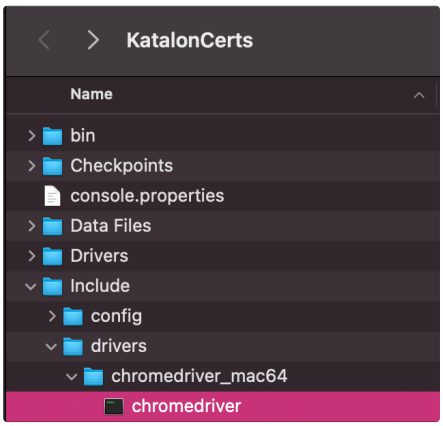
Name | **Type** | **Value**
binary | String | `C:\Program Files\Google\Chrome for Testing\win64-116.0.5793.0\chrome-win64\chrome.exe`

macOS

Name | **Type** | **Value**
binary | String | `/Applications/Google Chrome for Testing.app/Contents/MacOS/Google Chrome for Testing`, or any other path which you actually have on your OS



- Copy the downloaded chromedriver file, navigate to your KS project/Include/drivers `/chromedriver_mac64/` for **macOS** or `/chromedriver_win32/` for **Windows** then paste the chromedriver here:



Ref: Replace a Webdriver

Suggestion 2: Specify Chrome binary and ChromeDriver in test script directly

- Open **Katalon Studio** and use the below script:

```
// Explicitly set the path of Chrome Driver binary
System.setProperty("webdriver.chrome.driver",
"/Users/vi.kim/chromedriver/mac_arm-116.0.5829.0/chromedriver-mac-arm64/chromedriver");
// Explicitly set the path of Chrome Browser binary
import org.openqa.selenium.chrome.ChromeDriver
import org.openqa.selenium.chrome.ChromeOptions
import org.openqa.selenium.remote.DesiredCapabilities
import com.kms.katalon.core.webui.driver.DriverFactory
options = new ChromeOptions();
options.setBinary("/Applications/Google Chrome for Testing.app/Contents/MacOS/Google Chrome for Testing");
```

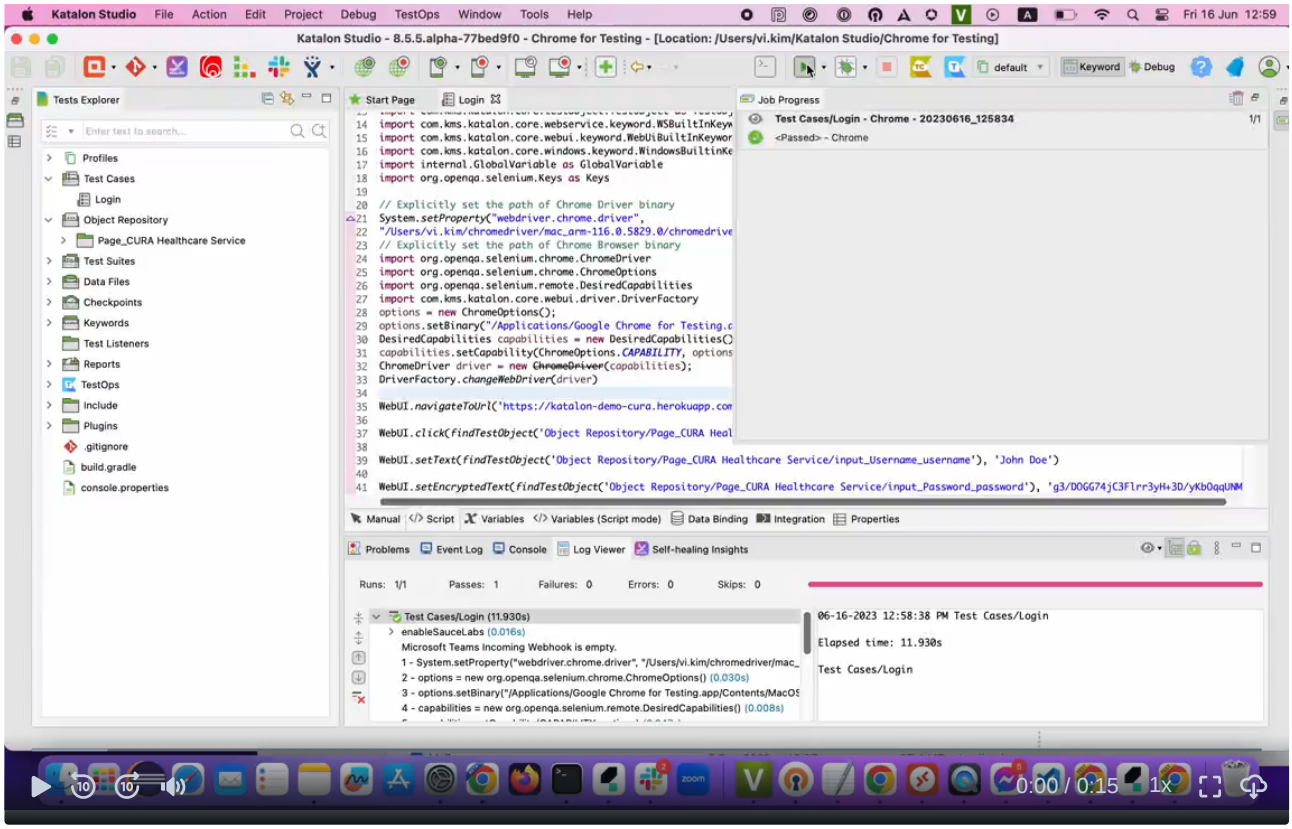
```
DesiredCapabilities capabilities = new DesiredCapabilities();
```

```
capabilities.setCapability(ChromeOptions.CAPABILITY, options);
```

```
ChromeDriver driver = new ChromeDriver(capabilities);
```

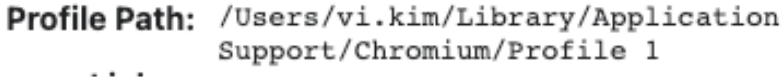
```
DriverFactory.changeWebDriver(driver)
```

```
WebUI.navigateToUrl('  CURA Healthcare Service  ')
```



Configure with a Chrome Profile

- Open **Chrome for Testing** browser and log in to your Chrome profile
- Navigate to **chrome://version**



- Use this sample script:

```
// Explicitly set the path of Chrome Driver binary

System.setProperty("webdriver.chrome.driver",

"/Users/vi.kim/chromedriver/mac_arm-116.0.5829.0/chromedriver-mac-arm64/chromedriver");

// Explicitly set the path of Chrome Browser binary

import org.openqa.selenium.chrome.ChromeDriver
```

```
import org.openqa.selenium.chrome.ChromeOptions
import org.openqa.selenium.remote.DesiredCapabilities
import com.kms.katalon.core.webui.driver.DriverFactory
options = new ChromeOptions();
options.addArguments("user-data-dir=" + "/Users/vi.kim/Library/Application Support/Chromium/")
options.addArguments("profile-directory=Profile 1");
options.setBinary("/Applications/Google Chrome for Testing.app/Contents/MacOS/Google Chrome for Testing");
DesiredCapabilities capabilities = new DesiredCapabilities();
capabilities.setCapability(ChromeOptions.CAPABILITY, options);
ChromeDriver driver = new ChromeDriver(capabilities);
DriverFactory.changeWebDriver(driver)
WebUI.navigateToUrl('CURA Healthcare Service')
```

Ref:

Chrome for Testing: reliable downloads for browser automation - Chrome Developers