W3C Capabilities Support

Sauce Labs now supports the W3C WebDriver-compliant capabilities and protocol featured in Selenium versions 3.11 and higher. Please note that some extended capabilities are not backwards-compatible with versions prior to 4.0. This article will help ensure that your tests are W3C WebDriver-compliant and can successfully execute on Sauce Labs.

We early adopted the W3C WebDriver specification when it achieved the W3C standard level as the automation protocol for web browsers. As browser vendors update to support W3C WebDriver and shift away from JSON Wire Protocol (JWP), it’s important to update your tests accordingly.

See the following sections for more information:

- Sauce Labs Web Protocol Support
- What You'll Need
- Getting Started
- W3C WebDriver Browser Compatibility
- Verifying Capabilities for W3C WebDriver Compliance
- W3C WebDriver-Compliant Language Changes in Selenium 3.11+
- Instantiating WebDriver with W3C WebDriver-Compliant Capabilities
- Common Test Script Configuration Errors to Avoid

Sauce Labs Web Protocol Support

Browser	W3C Support	JWP Support
Chrome	75 and above	Still supported
Firefox	55 and above	Still supported
Safari	12 and above	Removed in 12.1
Edge	All	Still supported
Internet Explorer	All	Still supported

What You'll Need

To ensure W3C WebDriver compliance, you'll need to:

- Use Selenium version 3.11 or higher.
- Switch completely from using legacy JWP desired capabilities to the new W3C WebDriver capabilities. Note that their naming conventions are slightly different – for example:

JWP (legacy)	W3C (new)
`platform`	`platformName`
`version`	`browserVersion`

  For more information on W3C WebDriver-compliant capabilities, head to the official W3C recommendation website.

- Include our custom **`sauce:options`** capabilities (e.g., `name`, `build`) in your Sauce Labs test scripts, bundled together, as seen in this example:

```
browserName: 'firefox',
platformName: 'macOS 10.15',
browserVersion: 'latest'
sauce:options: {
  name: 'My test name',
  build: 'My build',
  username: "SAUCE_USERNAME",
  accessKey: "SAUCE_ACCESS_KEY"
  seleniumVersion: "3.141.59"
}
```

  - `accessKey`
  - `appiumVersion`
  - `avoidProxy`
  - `build`
  - `captureHtml`
  - `chromedriverVersion`
  - `commandTimeout`
  - `crmuxdriverVersion`
  - `customData`
  - `disablePopupHandler`
  - `extendedDebugging`
  - `firefoxAdapterVersion`
  - `firefoxProfileUrl`
  - `idleTimeout`
  - `iedriverVersion`
  - `maxDuration`
  - `name`
  - `parentTunnel`
  - `passed`
  - `prerun`
  - `preventRequeue`
  - `priority`
  - `proxyHost`
  - `public`
  - `recordLogs`
  - `recordScreenshots`
  - `recordVideo`
  - `restrictedPublicInfo`
  - `screenResolution`
  - `seleniumVersion`
  - `source`
  - `tags`
  - `timeZone`
  - `tunnelIdentifier`
  - `username`
  - `videoUploadOnPass`

  More information: Test Configuration Options.

Getting Started

Below are a couple of mobile and web test script examples you can use to get up and running quickly:

W3C WebDriver Browser Compatibility

The following browser versions are compatible with the W3C WebDriver protocol:

- Firefox version 53 and higher
- Google Chrome version 61 and higher
- Internet Explorer 11

By default, Sauce Labs uses older versions of Firefox, IE, and Safari. This is important to know since newer commands and configurations may not be supported in those versions.

`chromeOptions()`

For tests on Google Chrome versions 74 and lower, the W3C WebDriver capability must be set as an experimental option. ChromeDriver version 75 runs in W3C WebDriver standard compliant mode by default, so setting this capability won't be necessary in the future. Here's an example:

```
ChromeOptions chOpts = new ChromeOptions();
chOpts.setExperimentalOption("w3c", true);
```

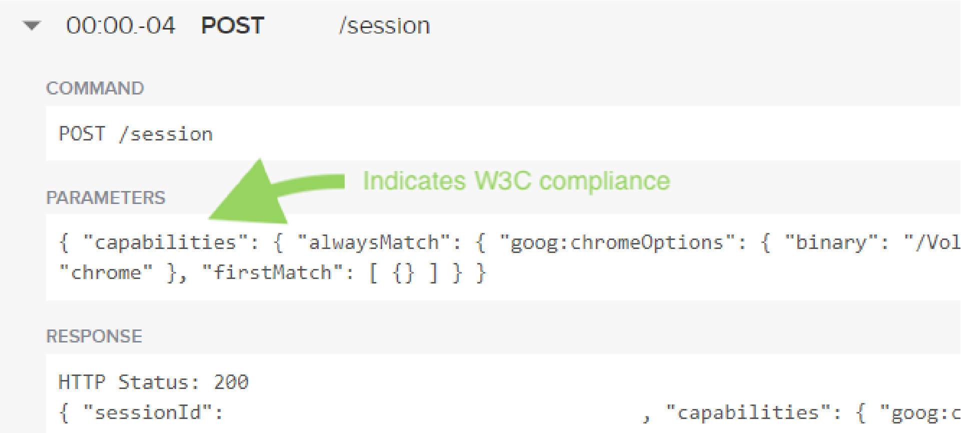
NOTE: `w3c` must be set as a boolean value (e.g., `true` in Java and `True` in Python) – not a string (e.g., "true").

Verifying Capabilities for W3C WebDriver Compliance

Here's how to verify if your tests are running under the new W3C WebDriver protocol:

1. Go to the **Test Details** page for your test in Sauce Labs.
2. Click the **Commands** tab.
3. Select the first `Post` command, which will be `Post /session`.
4. Under the **`Parameters`** section, check the capabilities that are used in the test.

- If your **`Parameters`** section begins with `capabilities`, you're running the new W3C WebDriver-compliant version ✓



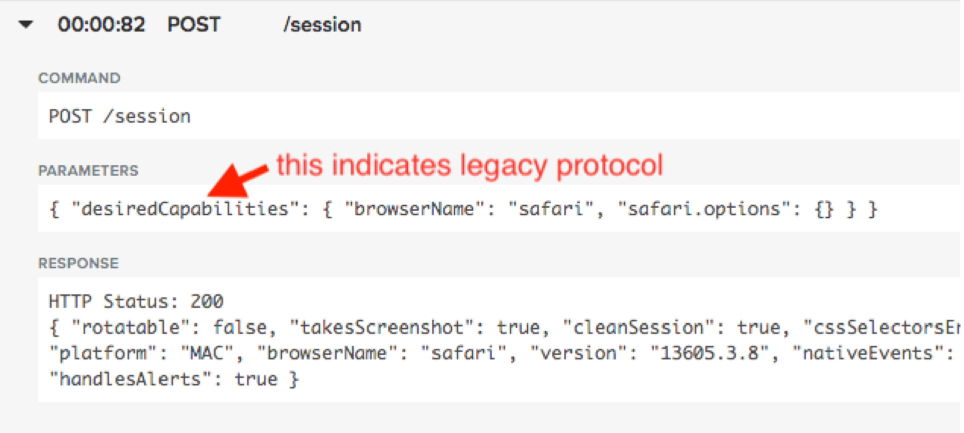
```
POST /session
```

Indicates W3C compliance

```
{ "capabilities": { "alwaysMatch": { "goog:chromeOptions": { "binary": "/Vol
"chrome" }, "firstMatch": [ {} ] } }
```

```
HTTP Status: 200
{ "sessionId":                                  , "capabilities": { "goog:c
```

- If it begins with `desiredCapabilities`, you're running the legacy, non-W3C WebDriver version ✗



```
POST /session
```

this indicates legacy protocol

```
{ "desiredCapabilities": { "browserName": "safari", "safari.options": {} } }
```

```
HTTP Status: 200
{ "rotatable": false, "takesScreenshot": true, "cleanSession": true, "cssSelectorsEr
"platform": "MAC", "browserName": "safari", "version": "13605.3.8", "nativeEvents":
"handlesAlerts": true }
```

W3C WebDriver-Compliant Language Changes in Selenium 3.11+

There are some changes to specific Selenium language bindings you should be aware of when migrating to the W3C WebDriver protocol. Here is an example:

Instantiating WebDriver with W3C WebDriver-Compliant Capabilities

Select a code snippet below in the programming language of your choice, then follow the instructions. You can find more sample code in the Sauce Labs training repository on GitHub.

Common Test Script Configuration Errors to Avoid

Here are some things to keep in mind when configuring your test script capabilities.

Don't Mix W3C Capabilities with JWP Capabilities

It's important not to mix W3C WebDriver-compliant capabilities with legacy JWP capabilities. This will result in a system error when you're spinning up a WebDriver session:

Mixed Capabilities Error

```
selenium.common.exceptions.WebDriverException: Message: Misconfigured -- Mixed Capabilities Error.

W3C keys (platformName/browserVersion) were detected alongside JWP keys (platform/version). To fix this,
replace all JWP keys with W3C keys.

The following desired capabilities were received:
{'browserName': 'chrome',
 'browserVersion': '80',
 'platform': 'Windows'}

See https://wiki.saucelabs.com/display/docs/w3c+capabilities+support for more details.
```

To fix this particular error, you'd need to change `platform` to `platformName` and then change `version` to `browserVersion`:

Solution

```
browserName: 'chrome',
platformName: 'Windows',
browserVersion: '80'
sauce:options: {
  name: 'My test name',
  build: 'My build',
  username: "SAUCE_USERNAME",
  accessKey: "SAUCE_ACCESS_KEY"
  seleniumVersion: "3.141.59"
}
```

Additional Resources

- Sauce Labs Supports the W3C WebDriver Protocol: run compliant tests on every browser
- Test Configuration Options: Sauce Labs capabilities for Selenium and Appium
- Useable code examples for your tests: public GitHub repo with language-specific bindings that leverage our new `sauce:options` capability