

Test Configuration Options

- You can use the [Platform Configurator](#) to get the correct configuration of testing options for your choice of Appium or Selenium tests in your favorite scripting language. The examples in this topic are for Java.
 - You can find out more about Selenium testing options in the [DesiredCapabilities](#) page of the SeleniumHQ wiki
 - You can find out more about Appium testing options in the [Appium Server Capabilities](#) page of the Appium.io website.
-

- Selenium-Specific Options
 - Required Selenium Test Configuration Settings
 - Other Selenium Options
 - Selenium Version
 - Chrome Driver Version
 - Internet Explorer Driver Version
- Appium-Specific Options
 - Required Appium Test Configuration Settings
 - Other Appium Options
- General Options
 - Alerts
 - Auto Accept Alerts
 - Test Annotation
 - Timeouts
- Sauce Testing Options
- Optional Sauce Testing Features

Selenium-Specific Options

You can find out more about Selenium testing options in the [DesiredCapabilities](#) page of the SeleniumHQ wiki

Required Selenium Test Configuration Settings

Setting	Description	Key	Value Type	Example
Browser Name	The name of the browser test against. See the Platform Configurator for available options.	browserName	string	"browserName": "firefox"

Browser Version	The version of the browser you want to use in your test.	version	string	<pre>"version": "61.0"</pre> <div style="border: 1px solid green; padding: 5px; margin: 5px 0;"> <p>Default to Latest Version of Chrome or Firefox</p> <p>If you want to use the latest stable version of Google Chrome or Firefox that Sauce supports, you can use "version": "latest". You can also use "version": "latest-1" or "version": "latest-2", etc. to request the next most recent versions of a browser. For example, if the latest stable version of Chrome is 48, you can request "latest-2" to use Chrome 46.</p> </div> <div style="border: 1px solid green; padding: 5px; margin: 5px 0;"> <p>Microsoft Edge versions</p> <p>Microsoft Edge has two version numbers, the browser application version and the EdgeHTML rendering engine version. For example, the current stable release of Edge as of September 2017 has the browser application version 40.15063 and the EdgeHTML version 15.15063. The Wikipedia page on Microsoft Edge covers this in more detail: https://en.wikipedia.org/wiki/Microsoft_Edge It is the EdgeHTML version that should be specified here, such as "version": "15" or "version": "15.15063".</p> </div>
Platform	Which operating system the browser should be running on. See the Platform Configurator for available options.	platform	string	<pre>"platform": "macOS 10.13"</pre>

Other Selenium Options

Option	Description	Key	Value Type	Example
--------	-------------	-----	------------	---------

Selenium Version

Allows you to choose the version of Selenium you want to use for your test.

For Firefox, the default version of Selenium when no value is specified depends on the version of Firefox.

Firefox and Selenium Versions

Firefox Version	Selenium Version
Firefox 47	Selenium 2.53.1 only
Firefox 46	Selenium 2.51.0, 2.52.0, 2.53.0, 2.53.1
Firefox 44 - 45	Selenium 2.48.0, 2.50.0, 2.51.0, 2.52.0, 2.53.0
Firefox 39 - 43	Selenium 2.47.1, 2.48.0, 2.50.0, 2.51.0, 2.52.0, 2.53.0
Firefox 38	Selenium 2.46.0
Firefox 32 - 37	Selenium 2.45.0
Firefox 26 - 31	Selenium 2.40.0
Firefox 23 - 25	Selenium 2.35.0
Firefox 21 - 22	Selenium 2.33.0
Firefox 20	Selenium 2.31.0
Firefox 19	Selenium 2.30.0
Firefox 17 - 18	Selenium 2.29.0
Firefox 12 - 16	Selenium 2.27.0
Firefox 11 and below	Selenium 2.18.0

"Marionette" replaces the Firefox driver in Firefox 48+. Its binary is called geckodriver (previously wires). The version we use depends on whether the job is Selenium 2 or 3:

- For Selenium 2, the marionette (geckodriver) version is 0.9.0
- For Selenium 3, the marionette (geckodriver) version is 0.11.1
- For Firefox 53 and above, the marionette (geckodriver) version is 0.16.0
- For Firefox 55 and above, the marionette (geckodriver) version is 0.18.0
- For Firefox 57 and above, the marionette (geckodriver) version is 0.23.0

When testing with Chrome and Internet Explorer, Selenium Version is not used to determine the version of the ChromeDriver or IEDriver that is used. For these browsers you should set the driver version as described for the **Chrome Driver Version** and **Internet Explorer Driver Version** options.

seleniumVersion

string

"seleniumVersion": "2.46.0"

Default Selenium Version

By default, Sauce Labs will use the following version of Selenium, depending on your selected combination of browser and operating system. While Selenium 3 is not yet fully implemented as a default version, it is supported for all Chrome and Firefox browsers on Mac and Windows platforms, for Safari 10+ on macOS 10.12 Sierra, and for Microsoft Edge and IE browsers version 10 and above. Currently Sauce Labs supports Selenium 3.4.0+ for Firefox and Safari and Selenium 3.5.0+ for Microsoft Edge and Chrome.

Browser Name	Default Selenium Version
Microsoft Edge	2.52.0
Chrome	Latest Chromedriver
Firefox	Dev: 3.4.0 Beta: 3.4.0 53+: 3.4.0 39+: 2.53.1
Safari	11.0: 3.4.0 < 11.0: 2.48.0
Internet Explorer	2.53.1

You can set the Selenium version for your tests by using the seleniumVersion desired capability:

```
'seleniumVersion' = '3.8.1'
```

Chrome Driver Version

Sauce Labs supports the ChromeDriver version 1 series (i.e. 26.0.1383.0) and the version 2 series (i.e. 2.15). The default version of ChromeDriver when no value is specified depends on the version of Chrome

NOTE: This capability only applies to Desktop Chrome tests

[Chrome and Chrome Driver](#)

[Versions](#)

Chrome Version	Chrome Driver Version
Chrome 69-70	ChromeDriver 2.43
Chrome 68	ChromeDriver 2.41
Chrome 67	ChromeDriver 2.40
Chrome 65-66	ChromeDriver 2.38
Chrome 64	ChromeDriver 2.37
Chrome 62-63	ChromeDriver 2.34
Chrome 61	ChromeDriver 2.32
Chrome 60	ChromeDriver 2.29
Chrome 59	ChromeDriver 2.29
Chrome 58	ChromeDriver 2.29
Chrome 57	ChromeDriver 2.28
Chrome 56	ChromeDriver 2.28
Chrome 55	ChromeDriver 2.28
Chrome 54	ChromeDriver 2.27
Chrome 53	ChromeDriver 2.26
Chrome 52	ChromeDriver 2.24
Chrome 51	ChromeDriver 2.23
Chrome 46-50	ChromeDriver 2.21
Chrome 40-45	ChromeDriver 2.15
Chrome 37-39	ChromeDriver 2.11
Chrome 33-36	ChromeDriver 2.10
Chrome 31-32	ChromeDriver 2.8
Chrome 29-30	ChromeDriver 2.4
Chrome 28 and below	ChromeDriver 26.0.1383.0

chromedriverVersion

string

"chromedriverVersion": "2.15"

[Supported Chrome Drivers](#)

- 21.0.1180.0
- 23.0.1240.0
- 26.0.1383.0
 - 0.6
 - 0.7
 - 0.8
 - 0.9
 - 2.0
 - 2.1
 - 2.2
 - 2.3
 - 2.4
 - 2.5
 - 2.6
 - 2.7
 - 2.8
 - 2.9
 - 2.10
 - 2.11
 - 2.12
 - 2.13
 - 2.14
 - 2.15
 - 2.20
 - 2.21
 - 2.22
 - 2.23
 - 2.24
 - 2.26
 - 2.27
 - 2.28
 - 2.29
 - 2.30
 - 2.31
 - 2.32
 - 2.33
 - 2.34
 - 2.35
 - 2.36
 - 2.37
 - 2.38
 - 2.39
 - 2.40
 - 2.41
 - 2.43

<h2>Internet Explorer Driver Version</h2>	<p>The Internet Explorer Driver defaults to version 2.53.1 when no version is specified.</p> <p>Note that the versions of Internet Explorer Driver we have available correspond to major Selenium releases - we do not have all the minor point releases (e.g. 3.12.0.4) available.</p> <p>We recommend setting the Selenium Version (see above) to correspond with the Internet Explorer Driver version you select.</p> <p>Sauce Labs supports launching 64-bit IE on our 64-bit VMs: Windows 7, Windows 8, and Windows 8.1. This provides a workaround for two known Selenium issues:</p> <ol style="list-style-type: none"> Using a 32 bit driver on a 64 bit operating system causes Selenium's screenshot feature to only capture the part of the page currently visible in the browser viewport Selenium Issue 5876. Using a 64 bit driver on a 64 bit operating system causes text entry to be extremely slow Selenium Issue 5516. 	<p>iedriverVersion</p>	<p>string</p>	<pre>"iedriverVersion": "2.46.0"</pre> <p>Supported IE Drivers</p> <ul style="list-style-type: none"> • x64_2.29.0 • x64_2.39.0 • x64_2.40.0 • x64_2.41.0 • x64_2.42.0 • x64_2.43.0 • x64_2.44.0 • x64_2.45.0 • x64_2.46.0 • x64_2.48.0 • 2.21.1 • 2.21.2 • 2.24.0 • 2.25.3 • 2.26.0 • 2.28.0 • 2.29.0 • 2.30.1 • 2.31.0 • 2.32.2 • 2.33.0 • 2.34.0 • 2.35.0 • 2.35.1 • 2.35.2 • 2.35.3 • 2.36.0 • 2.37.0 • 2.38.0 • 2.39.0 • 2.40.0 • 2.41.0 • 2.42.0 • 2.43.0 • 2.44.0 • 2.45.0 • 2.46.0 • 2.48.0 • 2.49.0 • 2.50.0 • 2.51.0 • 2.52.0 • 2.52.1 • 2.52.2 • 2.53.0 • 2.53.1 • 3.0.0 • 3.1.0 • 3.2.0 • 3.3.0 • 3.3.1 • 3.4.0 • 3.5.0 • 3.5.1 • 3.6.0 • 3.7.0 • 3.8.0 • 3.9.0 • 3.10.0 • 3.11.0 • 3.11.1 • 3.12.0 • 3.13.0 • 3.14.0 • 3.141.0
---	---	------------------------	---------------	--

Appium-Specific Options

You can find out more about more about Appium-specific options in [the Appium Server Capabilities page of the Appium.io website](#)

Required Appium Test Configuration Settings

Platform Configurator

Setting	Description	Key	Value Type	Example
---------	-------------	-----	------------	---------

<p>Appium Version</p>	<p>The version of Appium that you want to run your tests with.</p>	<p>appiumVersion</p>	<p>string</p>	<p>"appiumVersion": "1.5.3"</p> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p>Default Appium Version</p> <p>If you don't select an Appium Version for your test in the Platform Configurator, this capability will automatically default to the latest version of Appium that is compatible with your selected OS. If you prefer to use a different version of Appium for your test, enter the version number you want as the value for the <code>appiumVersion</code> capability. You can find the release notes for each Appium version at the Appium GitHub repository. In order for you to have a window of time to check the compatibility of your test suites with the latest Appium version, it won't be set as the default version on Sauce until one week after the version release.</p> </div>
<p>Browser Name</p>	<p>The mobile web browser that will be automated in the simulator, emulator or device.</p>	<p>browserName</p>	<p>string</p>	<p>"browserName": "Safari"</p> <div style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p>Empty String for Mobile Native and Hybrid Apps</p> <p>If you're testing a mobile native application or a mobile hybrid application, the value for this capability should be an empty string. Check out Native and Hybrid Applications: What's the Difference? for more information.</p> </div>

Device Name	The name of the simulator, emulator, or device you want to use in the test.	deviceName	string	<pre>"deviceName": "Google Nexus 7 HD Emulator"</pre> <div data-bbox="1208 197 1461 848" style="border: 1px solid green; padding: 5px;"> <p>Generic Android Emulator</p> <p>For an Android emulator test you can request a generic Android emulator by using the option "deviceName": "Android Emulator". If you want to use an Android emulator that looks and feels like a specific Android phone or tablet, for example a Google Nexus 7 HD Emulator or a Samsung Galaxy S4, then instead of "deviceName": "Android Emulator", you need to specify the exact Android emulator skin to use, for example "deviceName": "Samsung Galaxy S4 Emulator".</p> </div> <div data-bbox="1208 869 1461 1360" style="border: 1px solid green; padding: 5px;"> <p>Emulator Skins and Configurations</p> <p>Each Android emulator skin will have a different configuration depending on the phone or tablet that it emulates. For example, all the skins have different resolutions, screen dimensions, pixel densities, memory, etc. You can use the Platform Configurator to get a list of the available Android emulator skins for the various Android emulator versions.</p> </div>
Platform Version	The mobile operating system version that you want to use in your test.	platformVersion	string	"platformVersion": "9.1"
Platform Name	The mobile operating system platform you want to use in your test.	platformName	string	"platformName": "iOS"

Application Path	The path to a .ipa, .apk or .zip file containing the app to test. This could be the location of your app in the Temporary Sauce Storage , for example, sauce-storage:myapp.zip, or the URL to a remote location where your app is located, for example http://myappurl.zip/.	app	string	<pre>"app": "sauce-storage:my_app.zip"</pre> <div data-bbox="1209 197 1458 512" style="border: 1px solid green; padding: 5px;"> <p>Required for Mobile Native and Hybrid Apps Only</p> <p>This capability is required only for testing mobile native or mobile hybrid applications. See Native and Hybrid Applications: What's the Difference? for more information</p> </div> <div data-bbox="1209 533 1458 751" style="border: 1px solid green; padding: 5px;"> <p>Not Required for Android</p> <p>This capability is not required for Android if you specify the appPackage and appActivity capabilities.</p> </div>
-------------------------	--	-----	--------	---

Other Appium Options

Option	Description	Key	Value	Example
Appium Version	The version of the Appium driver you want to use. If not specified the test will run against the default Appium version.	appiumVersion	string	It's better to specify the latest Appium version, which is the one suggested by the Platform Configurator , unless you have a reason for testing against some other version.
Device Type	The type of device to emulate. Options are: <ul style="list-style-type: none"> • tablet • phone 	deviceType	string	"deviceType": "tablet"
Device Orientation	The orientation in which the simulator/device will be rendered. Options are: <ul style="list-style-type: none"> • portrait • landscape 	deviceOrientation	string	"deviceOrientation": "portrait"
Automation Engine	The automation engine that will be used. Options are: <ul style="list-style-type: none"> • Appium • UiAutomator2 • Selendroid. The default is Appium.	automationName	string	"automationName": "UiAutomator2"

<h3>Application Package</h3> <p>For Android Only</p>	The Java package of the Android app you want to run.	appPackage	string	<pre>"appPackage": "com.example.android.myApp, com.android.settings"</pre> <p>Automatic Package Detection</p> <p>Appium automatically determines the package to launch, you only need to use this desired capability if you want to specify a package different than the default one.</p>
<h3>Android Activity</h3> <p>For Android Only</p>	The name for the Android activity you want to launch from your package.	appActivity	string	<pre>"appActivity": ".MainActivity"</pre> <p>Don't Forget the Dot!</p> <p>This capability needs to be preceded by a . (dot). For example, . MainActivity instead of MainActivity.</p> <p>Automatic Activity Detection</p> <p>Appium automatically determines the activity to launch, you only need to use this desired capability if you want to specify an activity different than the default one.</p>

General Options

These options can be set for both Selenium and Appium Tests.

Alerts

Option	Description	Key	Value Type	Example
--------	-------------	-----	------------	---------

<p>Auto Accept Alerts</p> <div style="border: 1px solid yellow; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;">For iOS Only</div>	<p>Setting this option will automatically accept any unexpected browser alerts that come up during your test, such as when Safari pops up the alert "Safari would like to use your current location (Don't Allow Allow)."</p>	<p>autoAcceptAlerts</p>	<p>boolean</p>	<pre>"autoAcceptAlerts": true</pre>
--	---	-------------------------	----------------	-------------------------------------

Test Annotation

You can add these annotations to your tests to make them easier to track and identify.

Option	Description	Key	Value Type	Example
Test Names	Used to record test names for jobs and make it easier to find individual tests	name	string	<pre>"name" : "my example name"</pre>
Build Numbers	Used to associate jobs with a build number or app version, which is then displayed on both the Dashboard and Archives view	build	string	<pre>"build": "build-1234"</pre>
Tagging	User-defined tags for grouping and filtering jobs in the Dashboard and Archives view	tags	list	<pre>"tags": ["tag1", "tag2", "tag3"]</pre>
Pass/Fail Status	Selenium and Appium handle sending commands to control a browser or app, but don't report to the server whether a test passed or failed. To record pass/fail status on Sauce, set the <code>passed</code> flag on the job. Since you can't know in advance whether a test passed or failed, this flag can't be set in the initial configuration.	passed	boolean	<pre>"passed": "true"</pre>
Custom Data	User-defined custom data that will accept any valid JSON object, limited to 64KB in size.	custom-data	object	<pre>"custom-data": { "release": "1.0", "commit": "0k392a9dkjr", "staging": true, "execution_number": 5, "server": "test.customer.com" }</pre>

Timeouts

Option	Description	Key	Value Type	Example
--------	-------------	-----	------------	---------

<p>Maximum Test Duration</p>	<p>As a safety measure to prevent tests from running indefinitely, Sauce limits the duration of tests to 30 minutes by default. You can adjust this limit on per-job basis and the maximum value is 10800 seconds.</p> <div style="border: 1px solid orange; padding: 10px; margin-top: 10px;"> <p>Don't Exceed 30 Minutes</p> <p>A test should never last more than 30 minutes and ideally should take less than five minutes. The 3 hour maximum exists mainly to ease the transition of new users migrating long running tests to Sauce Labs.</p> <p>While our test VMs respect the maxDuration desired capability when it's set in tests, it may not always be precise. Tests will never be timed out before their maxDuration has elapsed and in most cases, they will be timed out very shortly after their maxDuration has elapsed (usually less than 1 second). But, in some rare cases, such as when the test VM is suffering performance problems, they can be allowed to run longer (30 seconds or more).</p> </div>	<p>maxDuration</p>	<p>integer</p>	<p>"maxDuration": 1800</p>
<p>Command Timeout</p>	<p>As a safety measure to prevent Selenium crashes from making your tests run indefinitely, Sauce limits how long Selenium can take to run a command in our browsers. This is set to 300 seconds by default. The value of this setting is given in seconds. The maximum command timeout value allowed is 600 seconds.</p>	<p>commandTimeout</p>	<p>integer</p>	<p>"commandTimeout": 300</p>

Idle Test Timeout	As a safety measure to prevent tests from running too long after something has gone wrong, Sauce limits how long a browser can wait for a test to send a new command. This is set to 90 seconds by default and limited to a maximum value of 1000 seconds. You can adjust this limit on a per-job basis. The value of this setting is given in seconds.	idleTimeout	integer	"idleTimeout": 90
--------------------------	---	-------------	---------	-------------------

Sauce Testing Options

Option	Description	Key	Value Type	Example						
Version (Browser)	If this capability is null, an empty string, or omitted altogether, the latest version of the browser will be used automatically.	version	string or integer	"version": "35"						
Pre-run Executables	<p>You can provide a URL to an executable file, which will be downloaded and executed to configure the VM before the test starts. For faster performance, you may want to upload the executable to temporary Sauce storage. This capability takes a JSON object with four main keys. Check out the topics under Using Pre-Run Executables to Configure Browsers and Virtual Machines for more information.</p> <div data-bbox="402 932 656 1199" style="border: 1px solid green; padding: 5px; margin-bottom: 10px;"> <p>Running Autolt Scripts</p> <p>If you want to run an Autolt script during your test, compile it as an exe, send it using this capability, and set <code>background</code> to <code>true</code> to allow Autolt to continue running throughout the full duration of your test.</p> </div> <div data-bbox="402 1220 656 1436" style="border: 1px solid green; padding: 5px;"> <p>Using Multiple Pre-Run Executables</p> <p>If you need to send multiple pre-run executables, the best way is to bundle them into a single executable file, such as a self-extracting zip file.</p> </div>	<pre>prerun</pre> <p>(primary key)</p>		<pre>"prerun": { "executable": "http://url.to/your/executable.exe", "args": ["--silent", "-a", "-q"], "background": false, "timeout": 120 }</pre> <div data-bbox="1208 827 1461 1129" style="border: 1px solid green; padding: 5px; margin-top: 10px;"> <p>Sending a Single String Instead of JSON</p> <p>If a single string is sent as the <code>prerun</code> capability rather than a JSON object, this string is considered to be the URL to the executable, and the executable launches with <code>background</code> set to <code>false</code>.</p> </div>						
	The URL to the executable you want to run before your browser session starts.	<pre>executable</pre> <p>(secondary key)</p>								
	<p>A list of the command line parameters that you want the executable to receive. Valid arguments are:</p> <table border="1" data-bbox="402 1598 656 1839" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><code>--silent</code> or <code>/S</code></td> <td style="padding: 2px;">Installs the script silently without raising any dialogs</td> </tr> <tr> <td style="padding: 2px;"><code>-a</code></td> <td style="padding: 2px;">Add switches to the command line of the underlying <code>setup.exe</code> process</td> </tr> <tr> <td style="padding: 2px;"><code>-q</code></td> <td style="padding: 2px;">Like <code>--silent</code>, installs the script without raising any dialogs</td> </tr> </table>	<code>--silent</code> or <code>/S</code>	Installs the script silently without raising any dialogs	<code>-a</code>	Add switches to the command line of the underlying <code>setup.exe</code> process	<code>-q</code>	Like <code>--silent</code> , installs the script without raising any dialogs	<pre>args</pre> <p>(secondary key)</p>		
<code>--silent</code> or <code>/S</code>	Installs the script silently without raising any dialogs									
<code>-a</code>	Add switches to the command line of the underlying <code>setup.exe</code> process									
<code>-q</code>	Like <code>--silent</code> , installs the script without raising any dialogs									

	A boolean that defines whether Sauce should wait for this executable to finish before your browser session starts. If <code>background</code> isn't set or is set to <code>false</code> , Sauce will wait for up to 90 seconds for the executable to finish. At that point, the browser will start and your test will proceed.	<code>background</code> (secondary key)		
	The number of seconds Sauce will wait for your executable to finish before your browser session starts. If <code>timeout</code> isn't set, Sauce will wait for up to 90 seconds for the executable to finish. <code>timeout</code> is capped at 360 seconds and won't apply if <code>background</code> is set to <code>true</code> .	<code>timeout</code> (secondary key)		
Identified Tunnels	If you are using Sauce Connect Proxy to test an application that is behind a firewall or on your local machine, you must provide the identifier of the Sauce Connect tunnel to use. Check out Basic Sauce Connect Proxy Setup for more information.	<code>tunnelIdentifier</code>	string	<code>"tunnelIdentifier": "MyTunnel01"</code>
Shared Tunnels	This desired capability will let the test job use any shared tunnels available from the specified parent account, i.e. any account that is upstream in the hierarchy. If using a shared tunnel, you must specify both <code>tunnelIdentifier</code> and <code>parentTunnel</code>. Check out the topic Using Multiple Sauce Connect Tunnels for more information.	<code>parentTunnel</code>	string	<code>"tunnelIdentifier": "ParentTunnelName" "parentTunnel": "<username of parent>"</code>
Specifying the Screen Resolution	This setting specifies which screen resolution should be used during the test session. This feature is available in: <ul style="list-style-type: none"> Windows 7 (except Windows 7 with IE 9) Windows 8 Windows 8.1 Windows 10 <p>Resolutions Available for Windows 7</p> <ul style="list-style-type: none"> 800x600 1024x768 1152x864 1280x768 1280x800 1280x960 1280x1024 1440x900 1600x1200 1680x1050 1920x1080 1920x1200 2560x1600 	<code>screenResolution</code>	string	<code>"screenResolution": "1280x1024"</code>

Resolutions Available for

Windows 8, 8.1 and 10

- 800x600
- 1024x768
- 1152x864
- 1280x768
- 1280x800
- 1280x960
- 1280x1024
- 1400x1050
- 1440x900
- 1600x1200
- 1680x1050
- 1920x1080
- 1920x1200
- 2560x1600

- OS X 10.9
- OS X 10.10
- OS X 10.11
- macOS 10.2

Resolutions Available for OS X

10.9

- 800x600
- 1024x768
- 1152x720
- 1152x864
- 1152x900
- 1280x720
- 1280x768
- 1280x800
- 1280x960
- 1280x1024
- 1376x1032
- 1440x900
- 1600x900
- 1600x1200
- 1680x1050
- 1920x1080
- 1920x1200
- 2048x1152
- 2048x1536

	<p> Resolutions Available for OS X 10.10 <ul style="list-style-type: none"> • 800x600 • 1024x768 • 1152x720 • 1152x864 • 1152x900 • 1280x720 • 1280x768 • 1280x800 • 1280x960 • 1280x1024 • 1376x1032 • 1440x900 • 1600x900 • 1600x1200 • 1680x1050 • 1920x1080 • 1920x1200 • 1920x1440 • 2048x1152 • 2048x1536 </p> <p> Resolutions Available for OS X 10.11 <ul style="list-style-type: none"> • 1024x768 • 1152x864 • 1280x960 • 1376x1032 • 1600x1200 • 1920x1440 • 2048x1536 </p> <p> Resolutions Available for macOS 10.12 <ul style="list-style-type: none"> • 1024x768 • 1152x864 • 1280x960 • 1376x1032 • 1400x1050 • 1600x1200 • 1920x1440 • 2048x1536 • 2360x1770 </p> <p>Default screen resolution for Sauce tests when not specified is 1024x768.</p>			
<p>Custom Time Zones</p>	<p>Desktop Test VMs can be configured with custom time zones. This feature should work on all operating systems, however time zones on Windows VMs are approximate. They will default to the time zone that the provided location falls into. You can find a complete list of timezones on Wikipedia. Underscores should be replaced with spaces. Sauce takes only location names (not their paths), as shown in the example below.</p>	<p>timeZone</p>	<p>string</p>	<pre>"timeZone": "Los_Angeles" "timeZone": "Honolulu" "timeZone": "Alaska" "timeZone": "New_York"</pre>

<h2>Avoiding the Selenium Proxy</h2>	<p>By default, Sauce routes traffic from some WebDriver browsers (Edge, Internet Explorer and Safari) through the Selenium HTTP proxy server so that HTTPS connections with self-signed certificates work everywhere. The Selenium proxy server can cause problems for some users. If that's the case for you, you can configure Sauce to avoid using the proxy server and have browsers communicate directly with your servers.</p> <div style="border: 1px solid green; padding: 5px; margin: 10px 0;"> <p>Don't Need the Selenium Proxy with Firefox or Google Chrome</p> <p>Firefox and Google Chrome under WebDriver aren't affected by this flag as they handle invalid certificates automatically and there isn't a need to proxy through Selenium.</p> </div> <div style="border: 1px solid orange; padding: 5px; margin: 10px 0;"> <p>Incompatible with Sauce Connect Proxy</p> <p>This flag is incompatible with Sauce Connect Proxy.</p> </div>	<p>avoidProxy</p>	<p>boolean</p>	<pre>"avoidProxy": true</pre>
<h2>Job Visibility</h2>	<p>Sauce Labs supports several test result visibility levels, which control who can view the test details. The visibility level for a test can be set manually from the test results page, but also programatically when starting a test or with our REST API. For more information about sharing test result, see the topics under Sharing the Results of Sauce Labs Tests.</p> <p>Available visibility levels are:</p>	<p>public</p>	<p>string</p>	<pre>"public": "public"</pre>

public	Making your test public means that it is accessible to everyone, and may be listed on public web pages and indexed by search engines.
public restricted	If you want to share your job's result page and video, but keep the logs only for you, you can certainly do so with public restricted visibility mode. This visibility mode will hide the fancy job log as well as prohibit access to the raw Selenium log, so that anonymous users with the link will be able to watch the video and screen shots but won't be able to see what's being typed and done to get there. share You can also decide to make your test sharable. Making your test sharable means that it is only accessible to people having valid link and it is not listed on publicly available pages on saucelabs.com or indexed by search engines.
team	If you want to share your jobs with other team members (that were created as a sub-accounts of one parent account), you can use team visibility mode. Making your test accessible by team means that it is only accessible to people under the same root account as you.
private	If you don't want to share your test's result page and video with anyone, you should use private job visibility mode. This way, only you (the owner) will be able to view assets and test result page.

Optional Sauce Testing Features

By default, Sauce Labs captures screenshot and video of your tests. You can disable these and other optional test features.

Option	Description	Key	Value	Example
Disable video recording	By default, Sauce records a video of every test you run. This is generally handy for debugging failing tests, as well as having a visual confirmation that certain feature works (or still works!) However, there is an added wait time for screen recording during a test run.	<code>recordVideo</code>	boolean	<code>"recordVideo": false</code>
Disable video upload for passing tests	As an alternative to disabling video recording, the <code>videoUploadOnPass</code> setting will let you discard videos for passing tests identified using the <code>passed</code> setting. This disables video post-processing and uploading that may otherwise consume some extra time after your test is complete.	<code>videoUploadOnPass</code>	boolean	<code>"videoUploadOnPass": false</code>
Disable step-by-step screenshots	Sauce captures step-by-step screenshots of every test you run. Most users find it very useful to get a quick overview of what happened without having to watch the complete video. However, this feature may add some extra time to your tests. You can avoid this by optionally turning off this feature.	<code>recordScreenshots</code>	boolean	<code>"recordScreenshots": false</code>
Disable log recording	<p>By default, Sauce creates a log of all the actions that you execute to create a report for the test run that lets you troubleshoot test failures more easily.</p> <div style="border: 1px solid orange; padding: 10px; margin-top: 10px;"> <p>Selenium Logs Are Still Recorded</p> <p>This option only disables recording of the <code>log.json</code> file. The <code>selenium-server.log</code> will still be recorded even if you choose to disable recording of the <code>log.json</code>.</p> </div>	<code>recordLogs</code>	boolean	<code>"recordLogs": false</code>

Prioritize Jobs	If you have multiple new jobs waiting to start, for example across a collection of sub-accounts, jobs with a lower priority number take precedence over jobs with a higher number. So, for example, if you have multiple jobs simultaneously waiting to start, we'll first attempt to find resources to start all the jobs with priority 0, then all the jobs with priority 1, etc. When we run out of available virtual machines, or when you hit your concurrency limit, any jobs not yet started will wait. Within each priority level, jobs that have been waiting the longest take precedence.	priority	integer	"priority": 0
Enable WebDriver's automatic screenshots	Selenium WebDriver captures automatic screenshots for every server side failure, for example if an element is not found. Sauce disables this by default to reduce network traffic during tests, resulting in a considerable performance improvement in most tests. You can enable this feature, but keep in mind that it may be detrimental to the performance of your jobs.	webdriver.remote. quietExceptions	boolean	"webdriver.remote. quietExceptions": false
Enable Extended Debugging	Extended debugging records HAR files for some browsers, as well as console. json logs. These are extremely valuable for debugging flaky tests. Default value is false.	extendedDebugging	boolean	"extendedDebugging": true