

The Serenity Reference Manual:

<http://thucydides.info/docs/serenity-staging/>

<http://go-gaga-over-testing.blogspot.in/2015/11/complete-list-of-serenity-properties.html>

Serenity Properties

Absolute path of the property file where Serenity system property defaults are defined.
Defaults to `~/serenity.properties`

`webdriver.driver`

What browser do you want your tests to run in, for example firefox, chrome, phantomjs or iexplorer. You can also use the **driver** property as a shortcut.

`webdriver.provided_type`

If using a provided driver, what type is it. The implementation class needs to be defined in the `webdriver.provided.{type}` system property.

`webdriver.base.url`

The default starting URL for the application, and base URL for relative paths.

`webdriver.remote.url`

The URL to be used for remote drivers (including a selenium grid hub or SauceLabs URL)

`phantomjs.webdriver.port`

What port to run PhantomJS on (used in conjunction with `webdriver.remote.url` to register with a Selenium hub, e.g. `-Dphantomjs.webdriver=5555 -Dwebdriver.remote.url=http://localhost:4444/wd/hub`)

`webdriver.remote.driver`

The driver to be used for remote drivers

`serenity.driver.capabilities`

A set of user-defined capabilities to be used to configure the WebDriver driver. Capabilities should be passed in as a space or semi-colon-separated list of key:value pairs, e.g. "build:build-1234; max-duration:300; single-window:true; tags:[tag1,tag2,tag3]"

`webdriver.timeouts.implicitlywait`

How long webdriver waits for elements to appear by default, in milliseconds.

`webdriver.wait.for.timeout`

How long webdriver waits by default when you use a fluent waiting method, in milliseconds.

`webdriver.chrome.driver`

Path to the Chrome driver, if it is not on the system path.

`serenity.home`

The home directory for Serenity output and data files - by default, `$USER_HOME/.serenity`

`serenity.outputDirectory`

Where should reports be generated.

`serenity.project.name`

What name should appear on the reports

`serenity.only.save.failing.screenshots`

Should Serenity only store screenshots for failing steps? This can save disk space and speed up the tests a little. It is very useful for data-driven testing. This property is now deprecated. Use `serenity.take.screenshots` instead.

`serenity.ext.packages*`

Extension packages. This is a list of packages that will be scanned for custom TagProvider implementations. To add a custom tag provider, just implement the TagProvider interface and specify the root package for this provider in this parameter.

`serenity.verbose.screenshots`

Should Serenity take screenshots for every clicked button and every selected link? By default, a screenshot will be stored at the start and end of each step. If this option is set to true, Serenity will record screenshots for any action performed on a WebElementFacade, i.e. any time you use an expression like `element(...).click()`, `findBy(...).click()` and so on. This will be overridden if the `ONLY_SAVE_FAILING_SCREENSHOTS` option is set to true. @Deprecated This property is still supported, but `serenity.take.screenshots` provides more fine-grained control.

`serenity.take.screenshots`

Set this property to have more finer control on how screenshots are taken. This property can take the following values:

- **FOR_EACH_ACTION** : Similar to `serenity.verbose.screenshots`
- **BEFORE_AND_AFTER_EACH_STEP**,
- **AFTER_EACH_STEP**, and
- **FOR_FAILURES** : Similar to `serenity.only.save.failing.screenshots`

[serenity.report.encoding](#)

Encoding used to generate the CSV exports

[serenity.verbose.steps](#)

Set this property to provide more detailed logging of WebElementFacade steps when tests are run.

[serenity.reports.show.step.details](#)

Should Thucydides display detailed information in the test result tables. If this is set to true, test result tables will display a breakdown of the steps by result. This is false by default.

[serenity.report.show.manual.tests](#)

Show statistics for manual tests in the test reports.

[serenity.report.show.releases](#)

Report on releases (defaults to true).

[serenity.restart.browser.frequency](#)

During data-driven tests, some browsers (Firefox in particular) may slow down over time due to memory leaks. To get around this, you can get Serenity to start a new browser session at regular intervals when it executes data-driven tests.

[thucydides.step.delay](#)

Pause (in ms) between each test step.

[untrusted.certificates](#)

Useful if you are running Firefox tests against an HTTPS test server without a valid certificate. This will make Serenity use a profile with the AssumeUntrustedCertificateIssuer property set.

[refuse.untrusted.certificates](#)

Don't accept sites using untrusted certificates. By default, Thucydides accepts untrusted certificates - use this to change this behaviour.

[serenity.timeout](#)

How long should the driver wait for elements not immediately visible.

[serenity.browser.width](#) and [serenity.browser.height](#)

Resize the browser to the specified dimensions, in order to take larger screenshots. This should work with Internet Explorer and Firefox, but not with Chrome.

`serenity.resized.image.width`

Value in pixels. If set, screenshots are resized to this size. Useful to save space.

`serenity.keep.unscaled.screenshots`

Set to true if you wish to save the original unscaled screenshots. This is set to false by default.

`serenity.store.html.source`

Set this property to true to save the HTML source code of the screenshot web pages. This is set to false by default.

`serenity.issue.tracker.url`

The URL used to generate links to the issue tracking system.

`serenity.activate.firebugs`

Activate the Firebugs and FireFinder plugins for Firefox when running the WebDriver tests. This is useful for debugging, but is not recommended when running the tests on a build server.

`serenity.batch.strategy`

Defines batch strategy. Allowed values - `DIVIDE_EQUALLY` (default) and `DIVIDE_BY_TEST_COUNT`. `DIVIDE_EQUALLY` will simply divide the tests equally across all batches. This could be inefficient if the number of tests vary a lot between test classes. A `DIVIDE_BY_TEST_COUNT` strategy could be more useful in such cases as this will create batches based on number of tests.

`serenity.batch.count`

If batch testing is being used, this is the size of the batches being executed.

`serenity.batch.number`

If batch testing is being used, this is the number of the batch being run on this machine.

`serenity.use.unique.browser`

Set this to run all web tests in a single browser.

`serenity.locator.factory`

Set this property to override the default locator factory with another locator factory (for ex., `AjaxElementLocatorFactory` or `DefaultElementLocatorFactory`). By default, Serenity uses a custom locator factory called `DisplayedElementLocatorFactory`.

`serenity.native.events`

Activate and deactivate native events for Firefox by setting this property to true or false.

`security.enable_java`

Set this to true to enable Java support in Firefox. By default, this is set to false as it slows down the web driver.

`serenity.test.requirements.basedir`

The base folder of the sub-module where the jBehave stories are kept. It is assumed that this directory contains sub folders src/test/resources. If this property is set, the requirements are read from src/test/resources under this folder instead of the classpath or working directory. This property is used to support situations where your working directory is different from the requirements base dir (for example when building a multi-module project from parent pom with requirements stored inside a sub-module)

`serenity.proxy.http`

HTTP Proxy URL configuration for Firefox and PhantomJS

`serenity.proxy.http_port`

HTTP Proxy port configuration for Firefox and PhantomJS

`serenity.proxy.type`

HTTP Proxy type configuration for Firefox and PhantomJS

`serenity.proxy.user`

HTTP Proxy username configuration for Firefox and PhantomJS

`serenity.proxy.password`

HTTP Proxy password configuration for Firefox and PhantomJS

`serenity.logging`

Three levels are supported: QUIET, NORMAL and VERBOSE

`serenity.test.root`

The root package for the tests in a given project. If provided, Serenity will use this as the root package when determining the capabilities associated with a test. If you are using the File System Requirements provider, Thucydides will expect this directory structure to exist at the top of the requirements tree. If you want to exclude packages in a requirements definition and start at a lower level in the hierarchy, use the `serenity.requirement.exclusionsproperty`.

This is also used by the `PackageAnnotationBasedTagProvider` to know where to look for annotated requirements.

[serenity.requirements.dir](#)

Use this property if you need to completely override the location of requirements for the File System Provider.

[serenity.use.requirements.directories](#)

By default, Thucydides will read requirements from the directory structure that contains the stories. When other tag and requirements plugins are used, such as the JIRA plugin, this can cause conflicting tags. Set this property to false to deactivate this feature (it is true by default).

[serenity.annotated.requirements.dir](#)

Use this property if you need to completely override the location of requirements for the Annotated Provider. This is recommended if you use File System and Annotated provider simultaneously. The default value is stories.

[serenity.requirements.types](#)

The hierarchy of requirement types. This is the list of requirement types to be used when reading requirements from the file system and when organizing the reports. It is a comma-separated list of tags. The default value is: capability, feature.

[serenity.requirement.exclusions](#)

When deriving requirement types from a path, exclude any values from this comma-separated list.

[serenity.test.requirements.basedir](#)

The base directory in which requirements are kept. It is assumed that this directory contains sub folders src/test/resources. If this property is set, the requirements are read from src/test/resources under this folder instead of the classpath or working directory. If you need to set an independent requirements directory that does not follow the src/test/resources convention, use `serenity.requirements.dir1` instead

This property is used to support situations where your working directory is different from the requirements base dir (for example when building a multi-module project from parent pom with requirements stored inside a sub-module).

[serenity.release.types](#)

What tag names identify the release types (e.g. Release, Iteration, Sprint). A comma-separated list. By default, "Release, Iteration"

[serenity.locator.factory](#)

Normally, Serenity uses SmartElementLocatorFactory, an extension of the AjaxElementLocatorFactory when instantiating page objects. This is to ensure that web elements are available and usable before they are used. For alternative behaviour, you can set this value to DisplayedElementLocatorFactory, AjaxElementLocatorFactory or DefaultElementLocatorFactory.

chrome.switches

Arguments to be passed to the Chrome driver, separated by commas.

webdriver.firefox.profile

The path to the directory of the profile to use when starting firefox. This defaults to webdriver creating an anonymous profile. This is useful if you want to run the web tests using your own Firefox profile. If you are not sure about how to find the path to your profile, look here: <http://support.mozilla.com/en-US/kb/Profiles>. For example, to run the default profile on a Mac OS X system, you would do something like this:

```
$ mvn test -Dwebdriver.firefox.profile=/Users/johnsmart/Library/Application\
Support/Firefox/Profiles/2owb5g1d.default
```

On Windows, it would be something like:

```
C:\Projects\myproject>mvn test -Dwebdriver.firefox.profile=C:\Users\John
Smart\AppData\Roaming\Mozilla\Firefox\Profiles\mvxjy48u.default
```

firefox.preferences

A semicolon separated list of Firefox configuration settings. For ex.,

```
-
Dfirefox.preferences="browser.download.folderList=2;browser.download.manager.show
WhenStarting=false;browser.download.dir=c:\downloads"
```

Integer and boolean values will be converted to the corresponding types in the Firefox preferences; all other values will be treated as Strings. You can set a boolean value to true by simply specifying the property name, e.g. -Dfirefox.preferences=app.update.silent.

A complete reference to Firefox's configuration settings is given [here](#).

serenity.csv.extra.columns

Add extra columns to the CSV output, obtained from tag values.

serenity.console.headings

Write the console headings using ascii-art ("ascii", default value) or in normal text ("normal")

tags

Comma separated list of tags. If provided, only junit classes and/or methods with tags in this list will be executed. For example,

```
mvn verify -Dtags="iteration:11"
```

```
mvn verify -Dtags="color:red,flavor:strawberry"
```

output.formats

What format should test results be generated in. By default, this is "json,xml".

narrative.format

Set this property to *asciidoc* to activate using Asciidoc format in narrative text.

jira.url

If the base JIRA URL is defined, Serenity will build the issue tracker url using the standard JIRA form.

jira.project

If defined, the JIRA project id will be prepended to issue numbers.

jira.username

If defined, the JIRA username required to connect to JIRA.

jira.password

If defined, the JIRA password required to connect to JIRA.

show.pie.charts

Display the pie charts on the dashboard by default. If this is set to false, the pie charts will be initially hidden on the dashboard.

dashboard.tag.list

If set, this will define the list of tag types to appear on the dashboard screens

dashboard.excluded.tag.list:

If set, this will define the list of tag types to be excluded from the dashboard screens

json.pretty.printing

Format the JSON test outcomes nicely. "true" or "false", turned off by default.

simplified.stack.traces

Stack traces are by default decluttered for readability. For example, calls to instrumented code or internal test libraries is removed. This behaviour can be deactivated by setting this property to false.

`serenity.dry.run`

Run through the steps without actually executing them.

`feature.file,language`

What (human) language are the Cucumber feature files written in? Defaults to "en".

`serenity.maintain.session`

Keep the Thucydides session data between tests. Normally, the session data is cleared between tests.