
sauceclient

Release 1.0.1

Nov 11, 2018

Contents

1	Info	3
2	About sauceclient	5
3	About Sauce Labs	7
4	Install	9
5	Example Usage	11
6	Related Links	13
7	Compatibility Note	15
8	Python Module Reference	17
8.1	sauceclient Module	17
8.2	tests Module	20
9	Running Selenium WebDriver	23
9.1	Selenium WebDriver in Python (local and remote)	23
9.1.1	Requirements	23
9.1.2	Selenium - Local WebDriver example	23
9.1.3	Selenium - Remote WebDriver example	23
9.1.4	Running a Test From Local WebDriver	24
9.1.5	Running a Test From Sauce Labs	24
	Python Module Index	27

Python client library for Sauce Labs API.

CHAPTER 1

Info

- Author: Corey Goldberg (2013-2017)
- Docs: <https://sauceclient.readthedocs.io>
- Releases: <https://pypi.python.org/pypi/sauceclient>
- Dev: <https://github.com/cgoldberg/sauceclient>
- License: Apache v2.0

CHAPTER 2

About sauceclient

sauceclient is a Python client library for Sauce Labs API. You can manage a Selenium test environment and retrieve test assets from Sauce.

The API gives remote access to:

- Test Jobs & Assets (Results, Logs, Videos, Screenshots)
- Account
- Platform Information
- JavaScript Unit Tests
- Temporary Storage
- Tunnels

Note: *sauceclient* is not used for running tests on Sauce Labs' services. Test execution is handled by [Selenium WebDriver](#).

CHAPTER 3

About Sauce Labs

Sauce Labs is a service for running remote Selenium WebDriver tests. It ghas VM's with 800+ browser/OS combinations, allowing comprehensive cross-browser cross-platform test coverage.

CHAPTER 4

Install

- with *pip*:

```
pip install sauceclient
```

Example Usage

- public access:

```
import sauceclient

sc = sauceclient.SauceClient()
status = sc.information.get_status()
```

- with authorization:

```
import sauceclient

sc = sauceclient.SauceClient(
    'sauce-username',
    'sauce-access-key',
)
jobs = sc.jobs.get_jobs(full=True, limit=5)
```


CHAPTER 6

Related Links

- [Sauce Labs](#)
- [Sauce Labs REST API documentation](#)
- [Python bindings for Selenium WebDriver](#)

CHAPTER 7

Compatibility Note

Due to changes in the SauceLabs REST API, some of sauceclient's classes and methods have been changed or renamed since the pre-1.0 releases.

8.1 sauceclient Module

Sauce Labs REST API client

Copyright (c) 2013-2017 Corey Goldberg

This file is part of: sauceclient <https://github.com/cgoldberg/sauceclient>

License: Apache Version 2.0 <http://www.apache.org/licenses/LICENSE-2.0>

Sauce Labs REST API documentation: <http://saucelabs.com/docs/rest>

class `sauceclient.Account` (*client*)

Account Methods

These methods provide user account information and management. - <https://wiki.saucelabs.com/display/DOCS/Account+Methods>

change_access_key ()

Change access key of your account.

create_user (*username, password, name, email*)

Create a sub account.

get_activity ()

Check account concurrency limits.

get_concurrency ()

Check account concurrency limits.

get_siblings ()

Get a list of sibling accounts associated with provided account.

get_subaccount_info ()

Get information about a sub account.

get_subaccounts ()

Get a list of sub accounts associated with a parent account.

get_usage (*start=None, end=None*)
Access historical account usage data.

get_user ()
Access basic account information.

class sauceclient.**Analytics** (*client*)
Analytics Methods

These methods provide user account information and management. - <https://wiki.saucelabs.com/display/DOCS/Analytics+Methods>

get_build_trends (*start=None, end=None, time_range=None, scope=None, owner=None, status=None, pretty=False, os=None, browser=None*)

get_concurrency (*start=None, end=None, interval=None, time_range=None, scope=None, owner=None, status=None, pretty=False*)

get_error_trends (*start=None, end=None, time_range=None, scope=None, owner=None, status=None, pretty=False, os=None, browser=None*)

get_test_trends (*start=None, end=None, interval=None, time_range=None, scope=None, owner=None, status=None, pretty=False, os=None, browser=None*)

get_tests (*start=None, end=None, size=None, time_range=None, scope=None, owner=None, status=None, pretty=False, error=None, build=None, skip=None, missing_build=False*)

class sauceclient.**Information** (*client*)
Information Methods

Information resources are publicly available data about Sauce Lab's service. - <https://wiki.saucelabs.com/display/DOCS/Information+Methods>

get_appium_eol_dates ()
Get a list of Appium end-of-life dates. Dates are displayed in Unix time.

get_platforms (*automation_api='all'*)
Get a list of objects describing all the OS and browser platforms currently supported on Sauce Labs.

get_status ()
Get the current status of Sauce Labs services.

class sauceclient.**JavaScriptTests** (*client*)
JavaScript Unit Testing Methods

- <https://wiki.saucelabs.com/display/DOCS/JavaScript+Unit+Testing+Methods>

js_tests (*platforms, url, framework*)
Start your JavaScript unit tests on as many browsers as you like with a single request.

js_tests_status (*js_tests*)
Get the status of your JS unit tests.

class sauceclient.**Jobs** (*client*)
Job Methods

- <https://wiki.saucelabs.com/display/DOCS/Job+Methods>

delete_job (*job_id*)
Removes the job from the system with all the linked assets.

delete_job_assets (*job_id*)
Delete all the assets captured during a test run.

get_auth_token (*job_id*, *date_range=None*)
Get an auth token to access protected job resources.
<https://wiki.saucelabs.com/display/DOCS/Building+Links+to+Test+Results>

get_job (*job_id*)
Retreive a single job.

get_job_asset_url (*job_id*, *filename*)
Get details about the static assets collected for a specific job.

get_job_assets (*job_id*)
Get details about the static assets collected for a specific job.

get_jobs (*full=None*, *limit=None*, *skip=None*, *start=None*, *end=None*, *output_format=None*)
List jobs belonging to a specific user.

stop_job (*job_id*)
Terminates a running job.

update_job (*job_id*, *build=None*, *custom_data=None*, *name=None*, *passed=None*, *public=None*, *tags=None*)
Edit an existing job.

class sauceclient.**SauceClient** (*sauce_username=None*, *sauce_access_key=None*, *apibase=None*)
SauceClient class.

get_auth_string ()
Create auth string from credentials.

make_auth_headers (*content_type*)
Add authorization header.

make_headers (*content_type='application/json'*)
Create content-type header.

request (*method*, *url*, *body=None*, *content_type='application/json'*)
Send http request.

exception sauceclient.**SauceException** (**args*, ***kwargs*)
SauceClient exception.

class sauceclient.**Storage** (*client*)
Temporary Storage Methods

- <https://wiki.saucelabs.com/display/DOCS/Temporary+Storage+Methods>

get_stored_files ()
Check which files are in your temporary storage.

upload_file (*filepath*, *overwrite=True*)
Uploads a file to the temporary sauce storage.

class sauceclient.**Tunnels** (*client*)
Tunnel Methods

- <https://wiki.saucelabs.com/display/DOCS/Tunnel+Methods>

delete_tunnel (*tunnel_id*)
Get information for a tunnel given its ID.

get_tunnel (*tunnel_id*)
Get information for a tunnel given its ID.

`get_tunnels ()`
Retrieves all running tunnels for a specific user.

8.2 tests Module

```
class tests.TestSauce (methodName='runTest')

    setUp ()
        Hook method for setting up the test fixture before exercising it.

    test_account_change_access_key (**kwargs)
    test_account_create_user (**kwargs)
    test_account_get_activity (**kwargs)
    test_account_get_concurrency (**kwargs)
    test_account_get_siblings (**kwargs)
    test_account_get_subaccount_info (**kwargs)
    test_account_get_subaccounts (**kwargs)
    test_account_get_usage (**kwargs)
    test_account_get_user (**kwargs)
    test_analytics_get_build_trends (**kwargs)
    test_analytics_get_concurrency (**kwargs)
    test_analytics_get_error_trends (**kwargs)
    test_analytics_get_test_trends (**kwargs)
    test_analytics_get_tests (**kwargs)
    test_bad_request (**kwargs)
    test_information_get_appium_eol_dates (**kwargs)
    test_information_get_platforms (**kwargs)
    test_information_get_status (**kwargs)
    test_javascript_js_tests (**kwargs)
    test_javascript_js_tests_status (**kwargs)
    test_jobs_delete_job (**kwargs)
    test_jobs_delete_job_assets (**kwargs)
    test_jobs_get_auth_token (**kwargs)
    test_jobs_get_job (**kwargs)
    test_jobs_get_job_asset_url (**kwargs)
    test_jobs_get_job_assets (**kwargs)
    test_jobs_get_jobs (**kwargs)
    test_jobs_stop_job (**kwargs)
```



```
test_jobs_update_job (**kwargs)
test_storage_delete_tunnel (**kwargs)
test_storage_get_stored_files (**kwargs)
test_storage_upload_file (**kwargs)
test_tunnels_get_tunnel (**kwargs)
test_tunnels_get_tunnels (**kwargs)
```


9.1 Selenium WebDriver in Python (local and remote)

9.1.1 Requirements

- Python 2.7 or 3.x
- Selenium 2.x bindings (`pip install selenium`)

9.1.2 Selenium - Local WebDriver example

Let's start with a very simple Selenium WebDriver example...
consider the following Python code:

```
#!/usr/bin/env python

from selenium import webdriver

driver = webdriver.Firefox()
driver.get('http://saucelabs.com/test/guinea-pig')
driver.quit()
```

This code uses `webdriver.Firefox()`, to invoke the local FireFox driver.

9.1.3 Selenium - Remote WebDriver example

Instead of running locally via `webdriver.Firefox()`, we can use `webdriver.Remote()`, and have it execute *from* a remote machine running Selenium Server. In this case, the Sauce Labs cloud:

```
#!/usr/bin/env python

from selenium import webdriver

SAUCE_USERNAME = 'your-username-string'
SAUCE_ACCESS_KEY = 'your-access-key-string'

driver = webdriver.Remote(
    desired_capabilities=webdriver.DesiredCapabilities.FIREFOX,
    command_executor='http://%s:%s@ondemand.saucelabs.com:80/wd/hub' %
        (SAUCE_USERNAME, SAUCE_ACCESS_KEY)
)
driver.get('http://saucelabs.com/test/guinea-pig')
id = self.driver.session_id
print 'Link to your job: https://saucelabs.com/jobs/%s' % id
driver.quit()
```

9.1.4 Running a Test From Local WebDriver

The following Python script executes a simple test against a remote web server. It drives the local FireFox browser:

```
#!/usr/bin/env python

import unittest
from selenium import webdriver

class Selenium2OnLocal(unittest.TestCase):

    def setUp(self):
        self.driver = webdriver.Firefox()

    def test_from_local(self):
        self.driver.get('http://saucelabs.com/test/guinea-pig')
        self.assertEqual('I am a page title - Sauce Labs', self.driver.title)
        body = self.driver.find_element_by_css_selector('body')
        self.assertIn('This page is a Selenium sandbox', body.text)

    def tearDown(self):
        self.driver.quit()

if __name__ == '__main__':
    unittest.main()
```

9.1.5 Running a Test From Sauce Labs

Similar Python script as above, but now executing from Sauce Labs cloud. Notice the use of `webdriver.Remote()` as a replacement driver:

```
#!/usr/bin/env python

import unittest
from selenium import webdriver
```

(continues on next page)

(continued from previous page)

```
SAUCE_USERNAME = 'your-username-string'
SAUCE_ACCESS_KEY = 'your-access-key-string'

class Selenium2OnSauce(unittest.TestCase):

    def setUp(self):
        self.driver = webdriver.Remote(
            desired_capabilities=webdriver.DesiredCapabilities.FIREFOX,
            command_executor='http://%s:%s@ondemand.saucelabs.com:80/wd/hub' %
                (SAUCE_USERNAME, SAUCE_ACCESS_KEY)
        )

    def test_from_sauce(self):
        self.driver.get('http://saucelabs.com/test/guinea-pig')
        self.assertEqual('I am a page title - Sauce Labs', self.driver.title)
        body = self.driver.find_element_by_css_selector('body')
        self.assertIn('This page is a Selenium sandbox', body.text)

    def tearDown(self):
        id = self.driver.session_id
        print 'Link to your job: https://saucelabs.com/jobs/%s' % id
        self.driver.quit()

if __name__ == '__main__':
    unittest.main()
```


s

sauceclient, 17

t

tests, 20

A

Account (class in sauceclient), 17
Analytics (class in sauceclient), 18

C

change_access_key() (sauceclient.Account method), 17
create_user() (sauceclient.Account method), 17

D

delete_job() (sauceclient.Jobs method), 18
delete_job_assets() (sauceclient.Jobs method), 18
delete_tunnel() (sauceclient.Tunnels method), 19

G

get_activity() (sauceclient.Account method), 17
get_appium_eol_dates() (sauceclient.Information method), 18
get_auth_string() (sauceclient.SauceClient method), 19
get_auth_token() (sauceclient.Jobs method), 18
get_build_trends() (sauceclient.Analytics method), 18
get_concurrency() (sauceclient.Account method), 17
get_concurrency() (sauceclient.Analytics method), 18
get_error_trends() (sauceclient.Analytics method), 18
get_job() (sauceclient.Jobs method), 19
get_job_asset_url() (sauceclient.Jobs method), 19
get_job_assets() (sauceclient.Jobs method), 19
get_jobs() (sauceclient.Jobs method), 19
get_platforms() (sauceclient.Information method), 18
get_siblings() (sauceclient.Account method), 17
get_status() (sauceclient.Information method), 18
get_stored_files() (sauceclient.Storage method), 19
get_subaccount_info() (sauceclient.Account method), 17
get_subaccounts() (sauceclient.Account method), 17
get_test_trends() (sauceclient.Analytics method), 18
get_tests() (sauceclient.Analytics method), 18
get_tunnel() (sauceclient.Tunnels method), 19
get_tunnels() (sauceclient.Tunnels method), 19
get_usage() (sauceclient.Account method), 17
get_user() (sauceclient.Account method), 18

I

Information (class in sauceclient), 18

J

JavaScriptTests (class in sauceclient), 18
Jobs (class in sauceclient), 18
js_tests() (sauceclient.JavaScriptTests method), 18
js_tests_status() (sauceclient.JavaScriptTests method), 18

M

make_auth_headers() (sauceclient.SauceClient method), 19
make_headers() (sauceclient.SauceClient method), 19

R

request() (sauceclient.SauceClient method), 19

S

SauceClient (class in sauceclient), 19
sauceclient (module), 17
SauceException, 19
setUp() (tests.TestSauce method), 20
stop_job() (sauceclient.Jobs method), 19
Storage (class in sauceclient), 19

T

test_account_change_access_key() (tests.TestSauce method), 20
test_account_create_user() (tests.TestSauce method), 20
test_account_get_activity() (tests.TestSauce method), 20
test_account_get_concurrency() (tests.TestSauce method), 20
test_account_get_siblings() (tests.TestSauce method), 20
test_account_get_subaccount_info() (tests.TestSauce method), 20
test_account_get_subaccounts() (tests.TestSauce method), 20
test_account_get_usage() (tests.TestSauce method), 20
test_account_get_user() (tests.TestSauce method), 20

test_analytics_get_build_trends() (tests.TestSauce method), 20
test_analytics_get_concurrency() (tests.TestSauce method), 20
test_analytics_get_error_trends() (tests.TestSauce method), 20
test_analytics_get_test_trends() (tests.TestSauce method), 20
test_analytics_get_tests() (tests.TestSauce method), 20
test_bad_request() (tests.TestSauce method), 20
test_information_get_appium_eol_dates() (tests.TestSauce method), 20
test_information_get_platforms() (tests.TestSauce method), 20
test_information_get_status() (tests.TestSauce method), 20
test_javascript_js_tests() (tests.TestSauce method), 20
test_javascript_js_tests_status() (tests.TestSauce method), 20
test_jobs_delete_job() (tests.TestSauce method), 20
test_jobs_delete_job_assets() (tests.TestSauce method), 20
test_jobs_get_auth_token() (tests.TestSauce method), 20
test_jobs_get_job() (tests.TestSauce method), 20
test_jobs_get_job_asset_url() (tests.TestSauce method), 20
test_jobs_get_job_assets() (tests.TestSauce method), 20
test_jobs_get_jobs() (tests.TestSauce method), 20
test_jobs_stop_job() (tests.TestSauce method), 20
test_jobs_update_job() (tests.TestSauce method), 20
test_storage_delete_tunnel() (tests.TestSauce method), 21
test_storage_get_stored_files() (tests.TestSauce method), 21
test_storage_upload_file() (tests.TestSauce method), 21
test_tunnels_get_tunnel() (tests.TestSauce method), 21
test_tunnels_get_tunnels() (tests.TestSauce method), 21
tests (module), 20
TestSauce (class in tests), 20
Tunnels (class in sauceclient), 19

U

update_job() (sauceclient.Jobs method), 19
upload_file() (sauceclient.Storage method), 19