

---

# **simple-salesforce Documentation**

***Release 0.72.1***

**Nick Catalano, community contributors**

**Nov 09, 2017**



---

## Contents

---

<b>1</b>	<b>Examples</b>	<b>3</b>
<b>2</b>	<b>Record Management</b>	<b>5</b>
<b>3</b>	<b>Queries</b>	<b>7</b>
<b>4</b>	<b>Other Options</b>	<b>9</b>
<b>5</b>	<b>Using Apex</b>	<b>11</b>
<b>6</b>	<b>Additional Features</b>	<b>13</b>
<b>7</b>	<b>API documentation</b>	<b>15</b>
7.1	simple_salesforce package . . . . .	15
<b>8</b>	<b>Release history</b>	<b>27</b>
8.1	v0.72 . . . . .	27
8.2	v0.71 . . . . .	27
8.3	v0.70 . . . . .	27
8.4	v0.69 . . . . .	27
8.5	v0.68.2 . . . . .	28
<b>9</b>	<b>Authors &amp; License</b>	<b>29</b>
<b>10</b>	<b>Indices and tables</b>	<b>31</b>
	<b>Python Module Index</b>	<b>33</b>



Simple Salesforce is a basic Salesforce.com REST API client built for Python 2.6, 2.7, 3.3 and 3.4. The goal is to provide a very low-level interface to the REST Resource and APEX API, returning a dictionary of the API JSON response.

You can find out more regarding the format of the results in the [Official Salesforce.com REST API Documentation](#)

Contents:



# CHAPTER 1

---

## Examples

---

There are two ways to gain access to Salesforce

The first is to simply pass the domain of your Salesforce instance and an access token straight to `Salesforce()`

For example:

```
from simple_salesforce import Salesforce
sf = Salesforce(instance='na1.salesforce.com', session_id='')
```

If you have the full URL of your instance (perhaps including the schema, as is included in the OAuth2 request process), you can pass that in instead using `instance_url`:

```
from simple_salesforce import Salesforce
sf = Salesforce(instance_url='https://na1.salesforce.com', session_id='')
```

There are also two means of authentication, one that uses username, password and security token and the other that uses IP filtering, username, password and organizationId

To login using the security token method, simply include the Salesforce method and pass in your Salesforce username, password and token (this is usually provided when you change your password):

```
from simple_salesforce import Salesforce
sf = Salesforce(username='myemail@example.com', password='password', security_token=
    ↪ 'token')
```

To login using IP-whitelist Organization ID method, simply use your Salesforce username, password and organizationId:

```
from simple_salesforce import Salesforce
sf = Salesforce(password='password', username='myemail@example.com', organizationId=
    ↪ 'OrgId')
```

If you'd like to enter a sandbox, simply add `sandbox=True` to your `Salesforce()` call.

For example:

```
from simple_salesforce import Salesforce
sf = Salesforce(username='myemail@example.com.sandbox', password='password', security_
↳token='token', sandbox=True)
```

Note that specifying if you want to use a sandbox is only necessary if you are using the built-in username/password/security token authentication and is used exclusively during the authentication step.

If you'd like to keep track where your API calls are coming from, simply add `client_id='My App'` to your `Salesforce()` call.

```
from simple_salesforce import Salesforce
sf = Salesforce(username='myemail@example.com.sandbox', password='password', security_
↳token='token', sandbox=True, client_id='My App')
```

If you view the API calls in your Salesforce instance by Client Id it will be prefixed with `RestForce/`, for example `RestForce/My App`.

When instantiating a *Salesforce* object, it's also possible to include an instance of *requests.Session*. This is to allow for specialized session handling not otherwise exposed by `simple_salesforce`.

For example:

```
from simple_salesforce import Salesforce
import requests

session = requests.Session()
# manipulate the session instance (optional)
sf = Salesforce(
    username='user@example.com', password='password', organizationId='OrgId',
    session=session)
```



## CHAPTER 2

---

### Record Management

---

To create a new 'Contact' in Salesforce:

```
sf.Contact.create({'LastName': 'Smith', 'Email': 'example@example.com'})
```

This will return a dictionary such as {u'errors': [], u'id': u'003e0000003GuNXAA0', u'success': True}

To get a dictionary with all the information regarding that record, use:

```
contact = sf.Contact.get('003e0000003GuNXAA0')
```

To get a dictionary with all the information regarding that record, using a **custom** field that was defined as External ID:

```
contact = sf.Contact.get_by_custom_id('My_Custom_ID__c', '22')
```

To change that contact's last name from 'Smith' to 'Jones' and add a first name of 'John' use:

```
sf.Contact.update('003e0000003GuNXAA0', {'LastName': 'Jones', 'FirstName': 'John'})
```

To delete the contact:

```
sf.Contact.delete('003e0000003GuNXAA0')
```

To retrieve a list of deleted records between 2013-10-20 to 2013-10-29 (datetimes are required to be in UTC):

```
import pytz
import datetime
end = datetime.datetime.now(pytz.UTC) # we need to use UTC as salesforce API_
↳ requires this!
sf.Contact.deleted(end - datetime.timedelta(days=10), end)
```

To retrieve a list of updated records between 2014-03-20 to 2014-03-22 (datetimes are required to be in UTC):

```
import pytz
import datetime
end = datetime.datetime.now(pytz.UTC) # we need to use UTC as salesforce API requires,
↪ this
sf.Contact.updated(end - datetime.timedelta(days=10), end)
```

Note that Update, Delete and Upsert actions return the associated [Salesforce HTTP Status Code](#)

Use the same format to create any record, including 'Account', 'Opportunity', and 'Lead'. Make sure to have all the required fields for any entry. The [Salesforce API](#) has all objects found under 'Reference -> Standard Objects' and the required fields can be found there.

## CHAPTER 3

---

### Queries

---

It's also possible to write select queries in Salesforce Object Query Language (SOQL) and search queries in Salesforce Object Search Language (SOSL).

SOQL queries are done via:

```
sf.query("SELECT Id, Email FROM Contact WHERE LastName = 'Jones'")
```

If, due to an especially large result, Salesforce adds a `nextRecordsUrl` to your query result, such as `"nextRecordsUrl" : "/services/data/v26.0/query/01gD00000002HU6KIAW-2000"`, you can pull the additional results with either the ID or the full URL (if using the full URL, you must pass 'True' as your second argument)

```
sf.query_more("01gD00000002HU6KIAW-2000")
sf.query_more("/services/data/v26.0/query/01gD00000002HU6KIAW-2000", True)
```

As a convenience, to retrieve all of the results in a single local method call use

```
sf.query_all("SELECT Id, Email FROM Contact WHERE LastName = 'Jones'")
```

SOSL queries are done via:

```
sf.search("FIND {Jones}")
```

There is also 'Quick Search', which inserts your query inside the {} in the SOSL syntax. Be careful, there is no escaping!

```
sf.quick_search("Jones")
```

Search and Quick Search return `None` if there are no records, otherwise they return a dictionary of search results.

More details about syntax is available on the [Salesforce Query Language Documentation Developer Website](#)



## CHAPTER 4

---

### Other Options

---

To insert or update (upsert) a record using an external ID, use:

```
sf.Contact.upsert('customExtIdField__c/11999',{'LastName': 'Smith','Email':  
↪ 'smith@example.com'})
```

To retrieve basic metadata use:

```
sf.Contact.metadata()
```

To retrieve a description of the object, use:

```
sf.Contact.describe()
```

To retrieve a description of the record layout of an object by its record layout unique id, use:

```
sf.Contact.describe_layout('39wmxcw9r23r492')
```

To retrieve a list of top level description of instance metadata, user:

```
sf.describe()  
  
for x in sf.describe()["subjects"]:  
    print x["label"]
```



## CHAPTER 5

---

### Using Apex

---

You can also use this library to call custom Apex methods:

```
payload = {
  "activity": [
    {"user": "12345", "action": "update page", "time": "2014-04-21T13:00:15Z"}
  ]
}
result = sf.apexexecute('User/Activity', method='POST', data=payload)
```

This would call the endpoint `https://<instance>.salesforce.com/services/apexrest/User/Activity` with `data=` as the body content encoded with `json.dumps`

You can read more about Apex on the [Force.com Apex Code Developer's Guide](#)





## CHAPTER 6

---

### Additional Features

---

There are a few helper classes that are used internally and available to you.

Included in them are `SalesforceLogin`, which takes in a username, password, security token, optional boolean sandbox indicator and optional version and returns a tuple of (`session_id`, `sf_instance`) where `session_id` is the session ID to use for authentication to Salesforce and `sf_instance` is the domain of the instance of Salesforce to use for the session.

For example, to use `SalesforceLogin` for a sandbox account you'd use:

```
from simple_salesforce import SalesforceLogin
session_id, instance = SalesforceLogin(
    username='myemail@example.com.sandbox',
    password='password',
    security_token='token',
    sandbox=True)
```

Simply leave off the final `True` if you do not wish to use a sandbox.

Also exposed is the `SFType` class, which is used internally by the `__getattr__()` method in the `Salesforce()` class and represents a specific SObject type. `SFType` requires `object_name` (i.e. `Contact`), `session_id` (an authentication ID), `sf_instance` (hostname of your Salesforce instance), and an optional `sf_version`

To add a `Contact` using the default version of the API you'd use:

```
from simple_salesforce import SFType
contact = SFType('Contact', 'sessionid', 'na1.salesforce.com')
contact.create({'LastName': 'Smith', 'Email': 'example@example.com'})
```

To use a proxy server between your client and the Salesforce endpoint, use the `proxies` argument when creating `SalesForce` object. The `proxy` argument is the same as what requests uses, a map of scheme to proxy URL:

```
proxies = {
    "http": "http://10.10.1.10:3128",
    "https": "http://10.10.1.10:1080",
}
SalesForce(instance='na1.salesforce.com', session_id='', proxies=proxies)
```

All results are returned as JSON converted OrderedDict to preserve order of keys from REST responses.

## 7.1 simple\_salesforce package

### 7.1.1 Subpackages

simple\_salesforce.tests package

Submodules

simple\_salesforce.tests.test\_api module

simple\_salesforce.tests.test\_login module

simple\_salesforce.tests.test\_util module

Tests for simple-salesforce utility functions

```
class simple_salesforce.tests.test_util.TestExceptionHandler (methodName='runTest')
    Bases: unittest.case.TestCase
        Test the exception router

    setUp ()
        Setup the exception router tests

    test_expired_session ()
        Test an expired session (401 code)

    test_generic_error_code ()
        Test an error code that is otherwise not caught

    test_malformed_request ()
        Test a malformed request (400 code)
```

```
test_multiple_records_returned()
    Test multiple records returned (a 300 code)
```

```
test_request_refused()
    Test a refused request (403 code)
```

```
test_resource_not_found()
    Test resource not found (404 code)
```

```
class simple_salesforce.tests.test_util.TestXMLParser (methodName='runTest')
    Bases: unittest.case.TestCase
```

Test the XML parser utility function

```
test_date_to_iso8601()
    Test date conversion
```

```
test_returns_valid_value()
    Test that when given the correct XML a valid response is returned
```

## Module contents

Simple-Salesforce Tests

## 7.1.2 Submodules

### simple\_salesforce.api module

Core classes and exceptions for Simple-Salesforce

```
class simple_salesforce.api.PerAppUsage (used, total, name)
    Bases: tuple
```

```
name
    Alias for field number 2
```

```
total
    Alias for field number 1
```

```
used
    Alias for field number 0
```

```
class simple_salesforce.api.SFType (object_name, session_id, sf_instance, sf_version='38.0',
                                     proxies=None, session=None)
```

Bases: object

An interface to a specific type of SObject

```
__init__ (object_name, session_id, sf_instance, sf_version='38.0', proxies=None, session=None)
    Initialize the instance with the given parameters.
```

Arguments:

- **object\_name** – the name of the type of SObject this represents, e.g. *Lead* or *Contact*
- **session\_id** – the session ID for authenticating to Salesforce
- **sf\_instance** – the domain of the instance of Salesforce to use
- **sf\_version** – the version of the Salesforce API to use
- **proxies** – the optional map of scheme to proxy server

- **session** – Custom requests session, created in calling code. This enables the use of requests Session features not otherwise exposed by simple\_salesforce.

**create** (*data*, *headers=None*)

Creates a new SObject using a POST to .../{*object\_name*}/.

Returns a dict decoded from the JSON payload returned by Salesforce.

Arguments:

- **data** – a dict of the data to create the SObject from. It will be JSON-encoded before being transmitted.
- **headers** – a dict with additional request headers.

**delete** (*record\_id*, *raw\_response=False*, *headers=None*)

Deletes an SObject using a DELETE to .../{*object\_name*}/{*record\_id*}.

If *raw\_response* is false (the default), returns the status code returned by Salesforce. Otherwise, return the *requests.Response* object.

Arguments:

- **record\_id** – the Id of the SObject to delete
- **raw\_response** – a boolean indicating whether to return the response directly, instead of the status code.
- **headers** – a dict with additional request headers.

**deleted** (*start*, *end*, *headers=None*)

Gets a list of deleted records

Use the SObject Get Deleted resource to get a list of deleted records for the specified object. .../deleted/?start=2013-05-05T00:00:00+00:00&end=2013-05-10T00:00:00+00:00

- **start** – start datetime object
- **end** – end datetime object
- **headers** – a dict with additional request headers.

**describe** (*headers=None*)

Returns the result of a GET to .../{*object\_name*}/describe as a dict decoded from the JSON payload returned by Salesforce.

Arguments:

- **headers** – a dict with additional request headers.

**describe\_layout** (*record\_id*, *headers=None*)

Returns the layout of the object

Returns the result of a GET to .../{*object\_name*}/describe/layouts/<*recordid*> as a dict decoded from the JSON payload returned by Salesforce.

Arguments:

- **record\_id** – the Id of the SObject to get
- **headers** – a dict with additional request headers.

**get** (*record\_id*, *headers=None*)

Returns the result of a GET to .../{*object\_name*}/{*record\_id*} as a dict decoded from the JSON payload returned by Salesforce.

Arguments:

- `record_id` – the Id of the SObject to get
- `headers` – a dict with additional request headers.

**get\_by\_custom\_id** (*custom\_id\_field, custom\_id, headers=None*)

Return an SFTYPE by custom ID

Returns the result of a GET to `.../{object_name}/{custom_id_field}/{custom_id}` as a dict decoded from the JSON payload returned by Salesforce.

Arguments:

- **custom\_id\_field** – the API name of a custom field that was defined as an External ID
- `custom_id` - the External ID value of the SObject to get
- `headers` – a dict with additional request headers.

**metadata** (*headers=None*)

Returns the result of a GET to `.../{object_name}/` as a dict decoded from the JSON payload returned by Salesforce.

Arguments:

- `headers` – a dict with additional request headers.

**request**

Deprecated access to `self.session` for backwards compatibility

**update** (*record\_id, data, raw\_response=False, headers=None*)

Updates an SObject using a PATCH to `.../{object_name}/{record_id}`.

If `raw_response` is false (the default), returns the status code returned by Salesforce. Otherwise, return the `requests.Response` object.

Arguments:

- `record_id` – the Id of the SObject to update
- **data** – a dict of the data to update the SObject from. It will be JSON-encoded before being transmitted.
- **raw\_response** – a boolean indicating whether to return the response directly, instead of the status code.
- `headers` – a dict with additional request headers.

**updated** (*start, end, headers=None*)

Gets a list of updated records

Use the SObject Get Updated resource to get a list of updated (modified or added) records for the specified object.

`.../updated/?start=2014-03-20T00:00:00+00:00&end=2014-03-22T00:00:00+00:00`

- `start` – start datetime object
- `end` – end datetime object
- `headers` – a dict with additional request headers.

**upsert** (*record\_id, data, raw\_response=False, headers=None*)

Creates or updates an SObject using a PATCH to `.../{object_name}/{record_id}`.

If `raw_response` is false (the default), returns the status code returned by Salesforce. Otherwise, return the `requests.Response` object.

Arguments:

- **record\_id** – an identifier for the SObject as described in the Salesforce documentation
- **data** – a dict of the data to create or update the SObject from. It will be JSON-encoded before being transmitted.
- **raw\_response** – a boolean indicating whether to return the response directly, instead of the status code.
- **headers** – a dict with additional request headers.

```
class simple_salesforce.api.Salesforce(username=None, password=None, security_token=None, session_id=None, instance=None, instance_url=None, organizationId=None, sandbox=False, version='38.0', proxies=None, session=None, client_id=None)
```

Bases: object

Salesforce Instance

An instance of Salesforce is a handy way to wrap a Salesforce session for easy use of the Salesforce REST API.

```
__init__(username=None, password=None, security_token=None, session_id=None, instance=None, instance_url=None, organizationId=None, sandbox=False, version='38.0', proxies=None, session=None, client_id=None)
```

Initialize the instance with the given parameters.

Available kwargs

Password Authentication:

- **username** – the Salesforce username to use for authentication
- **password** – the password for the username
- **security\_token** – the security token for the username
- **sandbox** – **True if you want to login to test.salesforce.com, False** if you want to login to *login.salesforce.com*.

Direct Session and Instance Access:

- **session\_id** – Access token for this session

Then either \* **instance** – Domain of your Salesforce instance, i.e.

*na1.salesforce.com*

OR \* **instance\_url** – Full URL of your instance i.e.

*'https://na1.salesforce.com'*

Universal Kwargs: \* **version** – the version of the Salesforce API to use, for example

*29.0*

- **proxies** – the optional map of scheme to proxy server
- **session** – **Custom requests session, created in calling code. This** enables the use of requests Session features not otherwise exposed by simple\_salesforce.

```
apexexecute(action, method='GET', data=None, **kwargs)
```

Makes an HTTP request to an APEX REST endpoint

Arguments:

- **action** – The REST endpoint for the request.
- **method** – HTTP method for the request (default GET)
- **data** – A dict of parameters to send in a POST / PUT request
- **kwargs** – Additional kwargs to pass to *requests.request*

**describe** ()

Describes all available objects

**static parse\_api\_usage** (*sforce\_limit\_info*)

parse API usage and limits out of the Sforce-Limit-Info header

Arguments:

- **sforce\_limit\_info**: The value of response header ‘Sforce-Limit-Info’ Example 1: ‘api-usage=18/5000’ Example 2: ‘api-usage=25/5000; per-app-api-usage=17/250(appName=sample-connected-app)’

**query** (*query*, *\*\*kwargs*)

Return the result of a Salesforce SOQL query as a dict decoded from the Salesforce response JSON payload.

Arguments:

- **query** – the SOQL query to send to Salesforce, e.g. `SELECT Id FROM Lead WHERE Email = “waldo@somewhere.com”`

**query\_all** (*query*, *\*\*kwargs*)

Returns the full set of results for the *query*. This is a convenience wrapper around *query(...)* and *query\_more(...)*.

The returned dict is the decoded JSON payload from the final call to Salesforce, but with the *totalSize* field representing the full number of results retrieved and the *records* list representing the full list of records retrieved.

Arguments

- **query** – the SOQL query to send to Salesforce, e.g. `SELECT Id FROM Lead WHERE Email = “waldo@somewhere.com”`

**query\_more** (*next\_records\_identifier*, *identifier\_is\_url=False*, *\*\*kwargs*)

Retrieves more results from a query that returned more results than the batch maximum. Returns a dict decoded from the Salesforce response JSON payload.

Arguments:

- **next\_records\_identifier** – either the Id of the next Salesforce object in the result, or a URL to the next record in the result.
- **identifier\_is\_url** – True if *next\_records\_identifier* should be treated as a URL, False if *next\_records\_identifier* should be treated as an Id.

**quick\_search** (*search*)

Returns the result of a Salesforce search as a dict decoded from the Salesforce response JSON payload.

Arguments:

- **search** – the non-SOSL search string, e.g. *Waldo*. This search string will be wrapped to read `FIND {Waldo}` before being sent to Salesforce



**request**

Deprecated access to self.session for backwards compatibility

**restful** (*path, params, method='GET'*)

Allows you to make a direct REST call if you know the path

Arguments:

- **path:** The path of the request Example: subjects/User/ABC123/password'
- **params:** dict of parameters to pass to the path
- **method:** HTTP request method, default GET

**search** (*search*)

Returns the result of a Salesforce search as a dict decoded from the Salesforce response JSON payload.

Arguments:

- **search** – the fully formatted SOSL search string, e.g. *FIND {Waldo}*

**setPassword** (*user, password*)

Sets the password of a user

salesforce dev documentation link: [https://www.salesforce.com/us/developer/docs/api\\_rest/Content/dome\\_subject\\_user\\_password.htm](https://www.salesforce.com/us/developer/docs/api_rest/Content/dome_subject_user_password.htm)

Arguments:

- **user:** the userID of the user to set
- **password:** the new password

**set\_password** (*user, password*)

Sets the password of a user

salesforce dev documentation link: [https://www.salesforce.com/us/developer/docs/api\\_rest/Content/dome\\_subject\\_user\\_password.htm](https://www.salesforce.com/us/developer/docs/api_rest/Content/dome_subject_user_password.htm)

Arguments:

- **user:** the userID of the user to set
- **password:** the new password

**class** simple\_salesforce.api.**SalesforceAPI** (*username, password, security\_token, sandbox=False, sf\_version='27.0'*)

Bases: *simple\_salesforce.api.Salesforce*

Deprecated SalesforceAPI Instance

This class implements the Username/Password Authentication Mechanism using Arguments It has since been surpassed by the 'Salesforce' class, which relies on kwargs

**\_\_init\_\_** (*username, password, security\_token, sandbox=False, sf\_version='27.0'*)

Initialize the instance with the given parameters.

Arguments:

- **username** – the Salesforce username to use for authentication
- **password** – the password for the username
- **security\_token** – the security token for the username
- **sandbox** – **True** if you want to login to *test.salesforce.com*, **False** if you want to login to *login.salesforce.com*.

- **sf\_version** – the version of the Salesforce API to use, for example “27.0”

**class** `simple_salesforce.api.Usage` (*used, total*)

Bases: `tuple`

**total**

Alias for field number 1

**used**

Alias for field number 0

## simple\_salesforce.bulk module

Classes for interacting with Salesforce Bulk API

**class** `simple_salesforce.bulk.SFBulkHandler` (*session\_id, bulk\_url, proxies=None, session=None*)

Bases: `object`

Bulk API request handler Intermediate class which allows us to use commands,

such as ‘sf.bulk.Contacts.create(...)’

This is really just a middle layer, whose sole purpose is to allow the above syntax

**\_\_init\_\_** (*session\_id, bulk\_url, proxies=None, session=None*)

Initialize the instance with the given parameters.

Arguments:

- **session\_id** – the session ID for authenticating to Salesforce
- **bulk\_url** – API endpoint set in Salesforce instance
- **proxies** – the optional map of scheme to proxy server
- **session** – **Custom requests session, created in calling code. This** enables the use of requests Session features not otherwise exposed by simple\_salesforce.

**class** `simple_salesforce.bulk.SFBulkType` (*object\_name, bulk\_url, headers, session*)

Bases: `object`

Interface to Bulk/Async API functions

**\_\_init\_\_** (*object\_name, bulk\_url, headers, session*)

Initialize the instance with the given parameters.

Arguments:

- **object\_name** – the name of the type of SObject this represents, e.g. *Lead* or *Contact*
- **bulk\_url** – API endpoint set in Salesforce instance
- **headers** – bulk API headers
- **session** – **Custom requests session, created in calling code. This** enables the use of requests Session features not otherwise exposed by simple\_salesforce.

**delete** (*data*)

soft delete records

**hard\_delete** (*data*)

hard delete records

**insert** (*data*)  
insert records

**query** (*data*)  
bulk query

**update** (*data*)  
update records

**upsert** (*data*, *external\_id\_field*)  
upsert records based on a unique identifier

## simple\_salesforce.exceptions module

All exceptions for Simple Salesforce

**exception** `simple_salesforce.exceptions.SalesforceAuthenticationFailed` (*code*, *message*)

Bases: `simple_salesforce.exceptions.SalesforceError`

Thrown to indicate that authentication with Salesforce failed.

**\_\_init\_\_** (*code*, *message*)

**exception** `simple_salesforce.exceptions.SalesforceError` (*url*, *status*, *resource\_name*, *content*)

Bases: `exceptions.Exception`

Base Salesforce API exception

**\_\_init\_\_** (*url*, *status*, *resource\_name*, *content*)

Initialize the SalesforceError exception

SalesforceError is the base class of exceptions in simple-salesforce

**Args:** *url*: Salesforce URL that was called *status*: Status code of the error response *resource\_name*: Name of the Salesforce resource being queried *content*: content of the response

**message** = u'Unknown error occurred for {url}. Response content: {content}'

**exception** `simple_salesforce.exceptions.SalesforceExpiredSession` (*url*, *status*, *resource\_name*, *content*)

Bases: `simple_salesforce.exceptions.SalesforceError`

Error Code: 401 The session ID or OAuth token used has expired or is invalid. The response body contains the message and errorCode.

**message** = u'Expired session for {url}. Response content: {content}'

**exception** `simple_salesforce.exceptions.SalesforceGeneralError` (*url*, *status*, *resource\_name*, *content*)

Bases: `simple_salesforce.exceptions.SalesforceError`

A non-specific Salesforce error.

**message** = u'Error Code {status}. Response content: {content}'

**exception** `simple_salesforce.exceptions.SalesforceMalformedRequest` (*url*, *status*, *resource\_name*, *content*)

Bases: `simple_salesforce.exceptions.SalesforceError`

Error Code: 400 The request couldn't be understood, usually because the JSON or XML body contains an error.

**message = u'Malformed request {url}. Response content: {content}'**

**exception** `simple_salesforce.exceptions.SalesforceMoreThanOneRecord` (*url*, *status*, *resource\_name*, *content*)

Bases: `simple_salesforce.exceptions.SalesforceError`

Error Code: 300 The value returned when an external ID exists in more than one record. The response body contains the list of matching records.

**message = u'More than one record for {url}. Response content: {content}'**

**exception** `simple_salesforce.exceptions.SalesforceRefusedRequest` (*url*, *status*, *resource\_name*, *content*)

Bases: `simple_salesforce.exceptions.SalesforceError`

Error Code: 403 The request has been refused. Verify that the logged-in user has appropriate permissions.

**message = u'Request refused for {url}. Response content: {content}'**

**exception** `simple_salesforce.exceptions.SalesforceResourceNotFound` (*url*, *status*, *resource\_name*, *content*)

Bases: `simple_salesforce.exceptions.SalesforceError`

Error Code: 404 The requested resource couldn't be found. Check the URI for errors, and verify that there are no sharing issues.

**message = u'Resource {name} Not Found. Response content: {content}'**

## simple\_salesforce.login module

Login classes and functions for Simple-Salesforce

Heavily Modified from RestForce 1.0.0

`simple_salesforce.login.SalesforceLogin` (*username=None*, *password=None*, *security\_token=None*, *organizationId=None*, *sandbox=False*, *sf\_version='38.0'*, *proxies=None*, *session=None*, *client\_id=None*)

Return a tuple of (*session\_id*, *sf\_instance*) where *session\_id* is the session ID to use for authentication to Salesforce and *sf\_instance* is the domain of the instance of Salesforce to use for the session.

Arguments:

- *username* – the Salesforce username to use for authentication
- *password* – the password for the username
- *security\_token* – the security token for the username
- **organizationId** – the ID of your organization NOTE: *security\_token* and *organizationId* are mutually exclusive
- **sandbox** – True if you want to login to *test.salesforce.com*, False if you want to login to *login.salesforce.com*.
- **sf\_version** – the version of the Salesforce API to use, for example "27.0"
- *proxies* – the optional map of scheme to proxy server

- **session** – Custom requests session, created in calling code. This enables the use of requests Session features not otherwise exposed by simple\_salesforce.
- **client\_id** – the ID of this client

### simple\_salesforce.util module

Utility functions for simple-salesforce

`simple_salesforce.util.call_salesforce(url, method, session, headers, **kwargs)`  
Utility method for performing HTTP call to Salesforce.

Returns a *requests.result* object.

`simple_salesforce.util.date_to_iso8601(date)`  
Returns an ISO8601 string from a date

`simple_salesforce.util.exception_handler(result, name='')`  
Exception router. Determines which error to raise for bad results

`simple_salesforce.util.getUniqueElementValueFromXmlString(xmlString, element-Name)`

Extracts an element value from an XML string.

For example, invoking `getUniqueElementValueFromXmlString(`

`'<?xml version="1.0" encoding="UTF-8"?><foo>bar</foo>', 'foo')`

should return the value `'bar'`.

## 7.1.3 Module contents

Simple-Salesforce Package



## 8.1 v0.72

### 8.1.1 Bugs

- [134] query\_all changed to be non-recursive due to recursion limit being hit for large result sets.

## 8.2 v0.71

### 8.2.1 Features

- [131] Added the ability to pass custom headers through to requests

## 8.3 v0.70

### 8.3.1 Features

- [98] Requests session objects are now shared between SFTypes instances and the parent Salesforce object

## 8.4 v0.69

### 8.4.1 Features

- [103] Require requests[secure] to allow for system openssl, which will allow for TLSv1.2 in Python < 2.7.9 assuming system openssl is newer than Python's.

### 8.4.2 Other

- Replaced httpretty with responses for unit tests due to SSL-related errors in httpretty.

## 8.5 v0.68.2

### 8.5.1 Other

- [85] Added tox support, updated travis-ci config
- Added CHANGES file
- [97] `_call_salesforce` is now used consistently across API



## CHAPTER 9

---

### Authors & License

---

This package is released under an open source Apache 2.0 license. Simple-Salesforce was originally written by [Nick Catalano](#) but most newer features and bugfixes come from [community contributors](#). Pull requests submitted to the [GitHub Repo](#) are highly encouraged!

Authentication mechanisms were adapted from Dave Wingate's [RestForce](#) and licensed under a MIT license

The latest build status can be found at [Travis CI](#)



## CHAPTER 10

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### S

- `simple_salesforce`, [25](#)
- `simple_salesforce.api`, [16](#)
- `simple_salesforce.bulk`, [22](#)
- `simple_salesforce.exceptions`, [23](#)
- `simple_salesforce.login`, [24](#)
- `simple_salesforce.tests`, [16](#)
- `simple_salesforce.tests.test_util`, [15](#)
- `simple_salesforce.util`, [25](#)



## Symbols

[\\_\\_init\\_\\_\(\)](#) (simple\_salesforce.api.SFType method), 16  
[\\_\\_init\\_\\_\(\)](#) (simple\_salesforce.api.Salesforce method), 19  
[\\_\\_init\\_\\_\(\)](#) (simple\_salesforce.api.SalesforceAPI method), 21  
[\\_\\_init\\_\\_\(\)](#) (simple\_salesforce.bulk.SFBulkHandler method), 22  
[\\_\\_init\\_\\_\(\)](#) (simple\_salesforce.bulk.SFBulkType method), 22  
[\\_\\_init\\_\\_\(\)](#) (simple\_salesforce.exceptions.SalesforceAuthenticationFailed method), 23  
[\\_\\_init\\_\\_\(\)](#) (simple\_salesforce.exceptions.SalesforceError method), 23

## A

[apexexecute\(\)](#) (simple\_salesforce.api.Salesforce method), 19

## C

[call\\_salesforce\(\)](#) (in module simple\_salesforce.util), 25  
[create\(\)](#) (simple\_salesforce.api.SFType method), 17

## D

[date\\_to\\_iso8601\(\)](#) (in module simple\_salesforce.util), 25  
[delete\(\)](#) (simple\_salesforce.api.SFType method), 17  
[delete\(\)](#) (simple\_salesforce.bulk.SFBulkType method), 22  
[deleted\(\)](#) (simple\_salesforce.api.SFType method), 17  
[describe\(\)](#) (simple\_salesforce.api.Salesforce method), 20  
[describe\(\)](#) (simple\_salesforce.api.SFType method), 17  
[describe\\_layout\(\)](#) (simple\_salesforce.api.SFType method), 17

## E

[exception\\_handler\(\)](#) (in module simple\_salesforce.util), 25

## G

[get\(\)](#) (simple\_salesforce.api.SFType method), 17

[get\\_by\\_custom\\_id\(\)](#) (simple\_salesforce.api.SFType method), 18  
[getUniqueElementValueFromXmlString\(\)](#) (in module simple\_salesforce.util), 25

## H

[hard\\_delete\(\)](#) (simple\_salesforce.bulk.SFBulkType method), 22

## I

[insert\(\)](#) (simple\_salesforce.bulk.SFBulkType method), 22

## M

[message](#) (simple\_salesforce.exceptions.SalesforceError attribute), 23  
[message](#) (simple\_salesforce.exceptions.SalesforceExpiredSession attribute), 23  
[message](#) (simple\_salesforce.exceptions.SalesforceGeneralError attribute), 23  
[message](#) (simple\_salesforce.exceptions.SalesforceMalformedRequest attribute), 24  
[message](#) (simple\_salesforce.exceptions.SalesforceMoreThanOneRecord attribute), 24  
[message](#) (simple\_salesforce.exceptions.SalesforceRefusedRequest attribute), 24  
[message](#) (simple\_salesforce.exceptions.SalesforceResourceNotFound attribute), 24  
[metadata\(\)](#) (simple\_salesforce.api.SFType method), 18

## N

[name](#) (simple\_salesforce.api.PerAppUsage attribute), 16

## P

[parse\\_api\\_usage\(\)](#) (simple\_salesforce.api.Salesforce static method), 20  
[PerAppUsage](#) (class in simple\_salesforce.api), 16

## Q

[query\(\)](#) (simple\_salesforce.api.Salesforce method), 20

query() (simple\_salesforce.bulk.SFBulkType method), 23  
 query\_all() (simple\_salesforce.api.Salesforce method), 20  
 query\_more() (simple\_salesforce.api.Salesforce method), 20  
 quick\_search() (simple\_salesforce.api.Salesforce method), 20

## R

request (simple\_salesforce.api.Salesforce attribute), 20  
 request (simple\_salesforce.api.SFType attribute), 18  
 restful() (simple\_salesforce.api.Salesforce method), 21

## S

Salesforce (class in simple\_salesforce.api), 19  
 SalesforceAPI (class in simple\_salesforce.api), 21  
 SalesforceAuthenticationFailed, 23  
 SalesforceError, 23  
 SalesforceExpiredSession, 23  
 SalesforceGeneralError, 23  
 SalesforceLogin() (in module simple\_salesforce.login), 24  
 SalesforceMalformedRequest, 23  
 SalesforceMoreThanOneRecord, 24  
 SalesforceRefusedRequest, 24  
 SalesforceResourceNotFound, 24  
 search() (simple\_salesforce.api.Salesforce method), 21  
 set\_password() (simple\_salesforce.api.Salesforce method), 21  
 setPassword() (simple\_salesforce.api.Salesforce method), 21  
 setUp() (simple\_salesforce.tests.test\_util.TestExceptionHandler method), 15  
 SFBulkHandler (class in simple\_salesforce.bulk), 22  
 SFBulkType (class in simple\_salesforce.bulk), 22  
 SFType (class in simple\_salesforce.api), 16  
 simple\_salesforce (module), 25  
 simple\_salesforce.api (module), 16  
 simple\_salesforce.bulk (module), 22  
 simple\_salesforce.exceptions (module), 23  
 simple\_salesforce.login (module), 24  
 simple\_salesforce.tests (module), 16  
 simple\_salesforce.tests.test\_util (module), 15  
 simple\_salesforce.util (module), 25

## T

test\_date\_to\_iso8601() (simple\_salesforce.tests.test\_util.TestXMLParser method), 16  
 test\_expired\_session() (simple\_salesforce.tests.test\_util.TestExceptionHandler method), 15

test\_generic\_error\_code() (simple\_salesforce.tests.test\_util.TestExceptionHandler method), 15  
 test\_malformed\_request() (simple\_salesforce.tests.test\_util.TestExceptionHandler method), 15  
 test\_multiple\_records\_returned() (simple\_salesforce.tests.test\_util.TestExceptionHandler method), 15  
 test\_request\_refused() (simple\_salesforce.tests.test\_util.TestExceptionHandler method), 16  
 test\_resource\_not\_found() (simple\_salesforce.tests.test\_util.TestExceptionHandler method), 16  
 test\_returns\_valid\_value() (simple\_salesforce.tests.test\_util.TestXMLParser method), 16  
 TestExceptionHandler (class in simple\_salesforce.tests.test\_util), 15  
 TestXMLParser (class in simple\_salesforce.tests.test\_util), 16  
 total (simple\_salesforce.api.PerAppUsage attribute), 16  
 total (simple\_salesforce.api.Usage attribute), 22

## U

update() (simple\_salesforce.api.SFType method), 18  
 update() (simple\_salesforce.bulk.SFBulkType method), 23  
 updated() (simple\_salesforce.api.SFType method), 18  
 upsert() (simple\_salesforce.api.SFType method), 18  
 upsert() (simple\_salesforce.bulk.SFBulkType method), 23  
 Usage (class in simple\_salesforce.api), 22  
 used (simple\_salesforce.api.PerAppUsage attribute), 16  
 used (simple\_salesforce.api.Usage attribute), 22