
SalesforcePy Documentation

Aaron Caffrey

Sep 07, 2023

1	Contributors	3
2	Requirements	5
2.1	Get Started with SalesforcePy	5
2.2	Usage Examples	6
2.3	Advanced Usage	11
2.4	FAQs	12
2.5	Contribute	13
2.6	SalesforcePy Package Reference	14
3	Indices and tables	23
	Python Module Index	25
	Index	27

An absurdly simple package for making Salesforce Rest API calls.

by Aaron Caffrey, Colin Cheevers, Jose Garcia, Tania Prince

This package enables you to produce:

1. A Salesforce client that is reusable, minimalistic, and pythonic.
2. Interfaces that are closely knit into the Salesforce Rest API service specification.
3. Gradual support for the Salesforce API extended family (i.e., Chatter, Analytics, Wave, Tooling, Bulk, Metadata, and so on).

CHAPTER 1

Contributors

Thanks goes to the people who have contributed code to this module, see the [GitHub Contributors](#) page.

Python 2 or 3

2.1 Get Started with SalesforcePy

2.1.1 Install

From Git

Install directly from Git using PyPi.

```
pip install git+ssh://git@github.com/forcedotcom/SalesforcePy.git
```

From Local Source

1. Download and extract, or clone this repo.
2. cd into the `/SalesforcePy` directory.
3. Run `pip install ..`

2.1.2 Create a Client and Log In

Getting started with SalesforcePy is a three-step process:

1. Import SalesforcePy
2. Create a client
3. Perform a login request

```
# Import SalesforcePy
import SalesforcePy as sfdc

# Create an instance of a Salesforce client. Replace the credentials here with valid_
↪ ones.
client = sfdc.client(
    username="jsoap@universalcontainers.com",
    password="p@ssword1",
    client_id="XXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    ↪XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    client_secret="123456789123456789",
    login_url="test.salesforce.com",
    version = "38.0", # optional, defaults to the latest version for your instance.
    timeout = "30", # optional, defines a connect/read timeout value, if not specified
                    # requests can hang for minutes or more.
)

# Log in
login_results = client.login()
```

In this example, `login_results[0]` is a dict with the response from the Salesforce OAuth resource. The only supported flow at present is username-password. For more information about the response, see [Understanding the Username-Password OAuth Authentication Flow](#).

You can also use the context manager which handles `login()` and `logout()` automatically.

```
client_args = {'username' : 'jsoap@universalcontainers.com',
               'password' : 'p@ssword1',
               'client_id': 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
               ↪XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
               'client_secret' : '123456789123456789'
               }
with sfdc.client(**client_args) as client:
    search_result = client.search('FIND {"test"} RETURNING Case(Id)')
```

2.2 Usage Examples

2.2.1 Query

Once the login call has been performed successfully, the client maintains the session, allowing you to make API calls. Here's how you can perform a query.

```
query_results = client.query('SELECT Id, Name FROM Account LIMIT 1')
```

In the example above `query_results[0]` is a dict with the response. For more information, See [Execute a SOQL Query](#).

2.2.2 Query More

While the client `query()` method helps with making requests for a small amount of data, it is not useful if the total size of records exceeds the standard batch size (2000).

In such scenarios, `query_more()` comes handy. As the example shows, you simply provide the query string in the same way you did previously with `query()`:

```
query_results = client.query_more('SELECT Id, Name FROM Account')
```

In this example, `query_results[0]` is a list of dicts, each of which is a single query result (batch). The behaviour of `query_more()` is to consume `nextRecordsUrl` of each query result recursively until it runs out. For more information, see [Retrieving the Remaining SOQL Query Results in Execute a SOQL Query](#).

2.2.3 Insert sObjects

Use the `sobjects` class from the client to perform DML statements and file i/o with Salesforce objects. Here's an example to demonstrate such insert operation.

```
create_result = client.sobjects(object_type='Account').insert({"Name" : "SalesforcePy  
→"})
```

In this example, `create_result[0]` is a dict with the response. For more information, see [Create a Record](#).

2.2.4 Update sObjects

Update works similarly to insert, with the main difference being that `id` is a required kwarg in `sobjects` so it is clear which record is to be updated.

```
update_result = client.sobjects(id='0010Y0000055YG7QAM', object_type='Account').update(  
→{"Name" : "SalesforcePy 2"})
```

In the example above `update_result[0]` will be `None`. This is because the HTTP method used under the hood is PATCH, for which the expected success code is 204. The success code can be found in `update_result[1].status`. For more information, see [Update a Record](#).

2.2.5 Delete sObjects

```
delete_result = client.sobjects(id='0010Y0000055YG7QAM', object_type='Account').  
→delete()
```

In the example above `delete_result[0]` will be `None`. This is because the HTTP method used under the hood is DELETE, for which the expected success code is 204. The success code can be found in `delete_result[1].status`. For more information, see [Delete a Record](#).

2.2.6 Query SObject Row

If you know the ID of a record, you can easily retrieve the entire row using this method. This may be preferable to `query()` if you wish to get all fields without specifying them.

```
query_result = client.sobjects( object_type="Account", id="0010Y00000561jcQAA" ).  
→query()
```

2.2.7 Describe SObject

The describe method retrieves the individual metadata at all levels for the specified SObject.

```
describe_result = client.sobjects(object_type='Account').describe()
```

In the example above `describe_result[0]` will be a dict with the response. For more information, see [sObject Describe](#).

2.2.8 Describe Global

The describe global method lists the available objects and their metadata for the organization's data. In addition, it provides the organization encoding, as well as the maximum batch size permitted in queries.

```
describe_global_result = client.sobjects().describe_global()
```

In the example above `describe_global_result[0]` will be a dict with the response. For more information, see [Describe Global](#). The If-Modified-Since header cannot be used with this method.

2.2.9 Insert File

There are some special objects in Salesforce such as Attachment, Document, and ContentVersion which allow storage of files as blobs. The following is an example of how to insert a file.

```
# Create a file tuple ordered like so: ( filename, body, content_type )
file = ( "SalesforcePy.txt", "Hello world", "text/plain" )

insert_result = client.sobjects(object_type = "Attachment", binary_field="Body").
→insert({
    "Name": "SalesforcePy",
    "ParentId": "0010Y00000561jcQAA",
    "Description": "An excellent package"
}, binary=file )    # Pass your file through using the binary kwarg
```

2.2.10 Search

SOSL search statements can be made as follows:

```
search_result = client.search('FIND {SalesforcePy} RETURNING Account(Id, Name) LIMIT 5
→')
```

2.2.11 Execute Anonymous

Anonymous Apex can be executed in a Salesforce organisation like so:

```
ea_result = client.execute_anonymous('system.debug(\'Hello world.\');')
```

2.2.12 Approval Process

Approvals can be retrieved, submitted and approved/rejected

```
ap_result = client.approvals(requestBody)
```

Sample request body, submitting a record for approval

```
requestBody = {
  "requests": [{
    "actionType": "Submit",
    "contextId": "00161000011ueBV",
    "nextApproverIds": ["00561000000j3h2"],
    "comments": "this is a test",
    "contextActorId": "005610000027S1Y",
    "processDefinitionNameOrId": "test_account",
    "skipEntryCriteria": "true" }]
}
```

2.2.13 Chatter

Create a feed item (chatter post). It returns a 201 status code for a successful request. See the [Chatter REST API Developer Guide](#) for information on the expected body to create feed items.

```
# create chatter post
client.chatter.feed_item(body)

# create a comment on a chatter post
client.chatter.feed_comment('feed-elementid', body)
```

2.2.14 Wave

Retrieve a data set

Retrieve a wave data set using the `dataset()` function.

```
client.wave.dataset("opportunities")
```

Perform a query

Perform a SOQL query using the `wave query()` function.

```
query = {
  "query": ""q = load \"0Fb0N000000XuvBSAS/0Fc0N000001M5BMSA0\";\nq = filter q by
↪ 'Account.Industry' in
[\"Apparel\", \"Banking\", \"Biotechnology\"];\nq = group q by 'Account.Industry';\nq
↪= foreach q generate
'Account.Industry' as 'Account.Industry', count() as 'count';\nq = order q by
↪ 'Account.Industry' asc;\nq = limit q
2000;""
}

client.wave.query(query)
```

2.2.15 Bulk API 2.0

As a general rule, supported Bulk API 2.0 calls can be made from `client.jobs.ingest`. The samples here cover specific calls.

Create a Job

In this example, we create a job to insert accounts.

```
job_resource = {"object": "Account", "operation": "insert", "lineEnding": "CRLF"}
client.jobs.ingest.create(job_resource=job_resource)
```

For more information on the response for this request, see [Create a Job](#).

Upload Job Data

In this example, we create a job, then upload a csv file using the job ID.

```
job_resource = {"object": "Account", "operation": "insert", "lineEnding": "CRLF"}
create_result = client.jobs.ingest.create(job_resource=job_resource)

with open("/path/to/accounts.csv") as f:
    csv_file = f.read()
    job_id = create_result[0].get("id")
    batches_result = client.jobs.ingest.batches(job_id=job_id, csv_file=csv_file)
```

For more information on the response for this request, see [Upload Job Data](#).

Update a job state

In this example, we create a job, upload a csv file using its job ID, then update it with a state of UploadComplete.

```
job_resource = {"object": "Account", "operation": "insert", "lineEnding": "CRLF"}
create_result = client.jobs.ingest.create(job_resource=job_resource)
job_id = create_result[0].get("Id")

with open("/path/to/accounts.csv") as f:
    csv_file = f.read()
    batches_result = client.jobs.ingest.batches(job_id=job_id, csv_file=csv_file)

client.jobs.ingest.update(job_id=job_id, state="UploadComplete")
```

For more information on the response for this request, see [Close or Abort a Job](#).

Delete a job

In this example, we delete a job based on its ID. Assumed in this example that this value is stored in `job_id`.

```
delete_result = client.jobs.ingest.delete(job_id=job_id)
```

For more information on the response for this request, see [Delete a Job](#).

Get all jobs

In this example, we get a list of all jobs.

```
get_result = client.jobs.ingest.get()
```

For more information on the response for this request, see [Get All Jobs](#).

Get Job Info

In this example, we get information for a specific job based on its ID. Assumed in this example that this value is stored in `job_id`.

```
get_result = client.jobs.ingest.get(job_id=job_id)
```

For more information on the response for this request, see [Get Job Info](#).

Get Job Successes

In this example, we get a successes CSV for a given job based on its ID. Assumed in this example that this value is stored in `job_id`.

```
get_result = client.jobs.ingest.get(job_id=job_id, successes=True)
```

For more information on the response for this request, see [Get Job Successful Record Results](#).

Get Job Failures

In this example, we get a failures CSV for a given job based on its ID. Assumed in this example that this value is stored in `job_id`.

```
get_result = client.jobs.ingest.get(job_id=job_id, failures=True)
```

For more information on the response for this request, see [Get Job Failed Record Results](#).

Get Job Unprocessed Rows

In this example, we get an unprocessed rows CSV for a given job based on its ID. Assumed in this example that this value is stored in `job_id`.

```
get_result = client.jobs.ingest.get(job_id=job_id, unprocessed=True)
```

For more information on the response for this request, see [Get Job Unprocessed Record Results](#).

2.2.16 Logout

Expires the session by revoking the access token. It returns a 200 status code for a successful token revocation.

```
client.logout()
```

2.3 Advanced Usage

2.3.1 Using Keyword Arguments

Some of the parameters that are optionally defined at the client level can be defined at the function level as well. Function level arguments supersede the client arguments.

For example, you may want to define an overall timeout value of 30 for all requests but specify a higher value for query calls.

```
client = sfdc.client(
    username=username,
    password=password,
    client_id=client_id,
    client_secret=client_secret,
    timeout="30"
)
query_kwarg= {"timeout" : "60"}
client.query("Select Id FROM Account",**query_kwarg)
```

The following parameters support function level overriding:

- proxies
- timeout
- version

2.4 FAQs

2.4.1 How do I inspect my organisation schema?

1. Log in to [Workbench](#).
2. Go to **Info > Standard and Custom Objects**.
3. In the Object dropdown, choose the object you wish to inspect (eg. Case) then click Select.
4. Expand Fields. You should find what you're looking for here.

2.4.2 How do I test a query?

1. Log in to [Workbench](#).
2. Go to *Queries > SOQL Query*.
3. Enter your query or optionally use the form to help you build the query, then click *Query*.

2.4.3 Is it possible to debug requests being made by SalesforcePy?

Yes. Here's an example of how to do it, and what to expect.

```
import logging
import SalesforcePy as sfdc

username = "jsoap@universalcontainers.com"
password = "p@ssword1"
client_id = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.
↳XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
client_secret = "123456789123456789"

client = sfdc.client(
    username=username,
    password=password,
    client_id=client_id,
```

(continues on next page)

(continued from previous page)

```

    client_secret=client_secret
)
client.debug(level=logging.INFO)    # Tell the client to debug at an info level
client.login()    # Outputs "POST https://login.salesforce.com/services/oauth2/token"
→to logs

```

2.4.4 Can I specify a proxy to talk to Salesforce Org in the code?

Yes. Here's an example of how to do it.

```

import SalesforcePy as sfdc

username = "jsoap@universalcontainers.com"
password = "p@ssword1"
client_id = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX.
→XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
client_secret = "123456789123456789"

client = sfdc.client(
    username=username,
    password=password,
    client_id=client_id,
    client_secret=client_secret,
    proxies={"https": "localhost:8888/example/"})
    # `proxies` kwarg takes a dict as required by the `requests` module.
)

```

2.5 Contribute

2.5.1 Contribute to the Code

Git Workflow

1. Fork this repo.
2. Run `git clone -b <yourbranch> <yourfork>`.
3. From within the project root directory run `pip install`.
4. Develop!
5. Cover your code in `/tests`.
6. Create a pull request to the `forcedotcom:developer` branch.

How to Format Your Code

We recommend [pep8](#) Style Guide for Python the best you can. We also encourage using [autopep8](#) for automatic formatting.

2.5.2 Contribute to the Documentation

The tools used for documentation are:

- **Read The Docs:** An open source platform for automating, building, versioning, and hosting documentation. For more information see [Read the Docs: Documentation Simplified](#).
- **Sphinx:** Sphinx simplifies creating intelligent and beautiful documentation for Python projects (or other documents consisting of multiple reStructuredText sources). For more information on installation and usage, see [Getting Started with Sphinx](#).

Documentation file format is reStructuredText. Learn more with [this handy primer](#).

2.6 SalesforcePy Package Reference

2.6.1 SalesforcePy.chatter module

New in version 1.0.0.

```
class SalesforcePy.chatter.Chatter(client)  
    Bases: SalesforcePy.common.ApiNamespace
```

The Chatter namespace class from which all API calls to a Salesforce organisation are made.

New in version 1.0.0.

```
feed_comment (*args, **function_kwarg)
```

```
feed_item (*args, **function_kwarg)
```

```
class SalesforcePy.chatter.ChatterFeedComment(session_id, instance_url, _id, body,  
                                              **kwargs)  
    Bases: SalesforcePy.common.BaseRequest
```

Performs a request to `'/services/data/v<api_version>/chatter/feed-elements/<id>/capabilities/comments/items'`

New in version 1.0.0.

```
class SalesforcePy.chatter.ChatterFeedItem(session_id, instance_url, body, **kwargs)  
    Bases: SalesforcePy.common.BaseRequest
```

Performs a request to `'/services/data/v<api_version>/chatter/feed-elements'`

New in version 1.0.0.

2.6.2 SalesforcePy.common module

New in version 1.0.0.

```
class SalesforcePy.common.ApiNamespace(client)  
    Bases: object
```

Base class for API namespaces.

New in version 1.0.0.

```
class SalesforcePy.common.BaseRequest(session_id, instance_url, **kwargs)  
    Bases: object
```

Base class for all request objects, for convenience, new request types should inherit from this class.

New in version 1.0.0.

get_headers ()

Returns headers dict for the request.

Returns headers

Return type dict

get_request_url ()

Returns the request URL. (default: 'https://<instance_url><service>')

Returns request_url

Return type string

get_request_vars ()

Returns the variables required by *request()* and other functions.

Returns (headers, logger, request_object, response, service)

Return type (dict, logging.Logger, requests.Request|None, list|dict|None, string)

request ()

Makes request to Salesforce and returns serialised response. Catches any exceptions and appends them to *self.exceptions*.

return response: Salesforce response, if available

rtype list|dict|None

class SalesforcePy.common.s.OAuthRequest (*session_id, instance_url, **kwargs*)

Bases: [SalesforcePy.common.BaseRequest](#)

Base class for all OAuth request objects

New in version 1.0.0.

get_request_url ()

Returns the request URL. (default: 'https://<instance_url><service>')

Returns request_url

Return type string

request ()

Makes request to Salesforce and returns serialised response. Catches any exceptions and appends them to *self.exceptions*.

return response: Salesforce response, if available

rtype list|dict|None

exception SalesforcePy.common.SFDCRequestException

Bases: `exceptions.Exception`

This exception is raised when we fail to complete requests to the # noqa [SFDC REST API](#).# noqa

New in version 1.0.0.

SalesforcePy.common.delete_request (*base_request*)

Performs DELETE request for the class provided.

Param base_request: Class with which to make request.

Type [BaseRequest](#)

Returns response

Return type requests.Response

`SalesforcePy.common.get_request (base_request)`

Performs GET request for the class provided.

Param base_request: Class with which to make request.

Type *BaseRequest*

Returns response

Return type requests.Response

`SalesforcePy.common.kwarg_adder (func)`

Decorator to add the kwargs from the client to the kwargs at the function level. If the same parameters are used in both, the function level kwarg will supersede the one at the client level.

Parameters **func** – client function to add client kwargs to

Returns the function with updated kwargs

`SalesforcePy.common.patch_request (base_request)`

Performs PATCH request for the class provided.

Param base_request: Class with which to make request.

Type *BaseRequest*

Returns response

Return type requests.Response

`SalesforcePy.common.post_request (base_request)`

Performs POST request for the class provided.

Param base_request: Class with which to make request.

Type *BaseRequest*

Returns response

Return type requests.Response

`SalesforcePy.common.put_request (base_request)`

Performs PUT request for the class provided.

Param base_request: Class with which to make request.

Type *BaseRequest*

Returns response

Return type requests.Response

2.6.3 SalesforcePy.jobs module

class `SalesforcePy.jobs.Batches (session_id, instance_url, api_version, job_id, csv_file, **kwargs)`

Bases: `SalesforcePy.common.BaseRequest`

Performs a PUT request to `/services/data/vX.XX/jobs/ingest/<job_id>/batches`

New in version 1.1.0.

class SalesforcePy.jobs.**CreateJob**(*session_id*, *instance_url*, *api_version*, *request_body*,
***kwargs*)

Bases: *SalesforcePy.common.BaseRequest*

Performs a POST request to *'/services/data/vX.XX/jobs/ingest'*

New in version 1.1.0.

class SalesforcePy.jobs.**DeleteJob**(*session_id*, *instance_url*, *api_version*, *job_id*, ***kwargs*)

Bases: *SalesforcePy.common.BaseRequest*

Performs a DELETE request to *'/services/data/vX.XX/jobs/ingest/<job_id>'*

New in version 1.1.0.

class SalesforcePy.jobs.**GetJob**(*session_id*, *instance_url*, *api_version*, ***kwargs*)

Bases: *SalesforcePy.common.BaseRequest*

Performs a GET request to *'/services/data/vX.XX/jobs/ingest'*

New in version 1.1.0.

class SalesforcePy.jobs.**Ingest**(*client*)

Bases: *SalesforcePy.common.ApiNamespace*

The Ingest namespace class from which all Bulk API calls to a Salesforce organisation are made.

New in version 1.1.0.

batches (**args*, ***function_kwarg*)

create (**args*, ***function_kwarg*)

delete (**args*, ***function_kwarg*)

get (**args*, ***function_kwarg*)

update (**args*, ***function_kwarg*)

class SalesforcePy.jobs.**Jobs**(*client*)

Bases: *SalesforcePy.common.ApiNamespace*

class SalesforcePy.jobs.**UpdateJob**(*session_id*, *instance_url*, *api_version*, *job_id*, *request_body*,
***kwargs*)

Bases: *SalesforcePy.common.BaseRequest*

Performs a PATCH request to *'/services/data/vX.XX/jobs/ingest/<job_id>'*

New in version 1.1.0.

2.6.4 SalesforcePy.sfdc module

class SalesforcePy.sfdc.**ApprovalProcess**(*session_id*, *instance_url*, ***kwargs*)

Bases: *SalesforcePy.common.BaseRequest*

Returns a list of all approval processes. Can also be used to submit a particular record # noqa if that entity supports an approval process and one has already been defined. It also supports # noqa specifying a collection of different Process Approvals requests to have them all executed in bulk.

New in version 1.0.0.

class SalesforcePy.sfdc.**Client**(**args*, ***kwargs*)

Bases: *object*

The client class from which all API calls to a Salesforce organisation are made.

New in version 1.0.0.

approvals (*args, **function_kwarg)

debug (*args, **function_kwarg)

execute_anonymous (*args, **function_kwarg)

login (*args, **function_kwarg)

logout (*args, **function_kwarg)

query (*args, **function_kwarg)

query_more (*args, **function_kwarg)

search (*args, **function_kwarg)

set_api_version (*args, **function_kwarg)

set_instance_url (url)

Strips the protocol from *url* and assigns the value to *self.instance_url*

Parameters **url** (*string*) – Instance URL used to make requests (eg. 'https://eu11.salesforce.com')

subjects (*args, **function_kwarg)

class SalesforcePy.sfdc.**ExecuteAnonymous** (session_id, instance_url, ab, **kwargs)

Bases: *SalesforcePy.common.BaseRequest*

Performs a request to */services/data/vX.XX/tooling/executeAnonymous/*

New in version 1.0.0.

class SalesforcePy.sfdc.**Login** (username, password, client_id, client_secret, **kwargs)

Bases: *SalesforcePy.common.OAuthRequest*

Performs a request to *'/services/oauth2/token'*

New in version 1.0.0.

get_payload ()

Returns the payload dict to be used in the request.

Returns OAuth2 request body required to obtain access token.

Return type dict

get_session_id ()

Returns the session ID obtained if the login request was successful

Returns Session ID

Return type string

request ()

Gets the result of *super* for this method, then assigns the *access_token* to *session_id*. Returns request response.

return Response dict

rtype dict

exception SalesforcePy.sfdc.**LoginException**

Bases: *exceptions.Exception*

Exception thrown during due to login failure.

New in version 1.0.0.

class SalesforcePy.sfdc.**Logout** (*session_id, instance_url, **kwargs*)
 Bases: *SalesforcePy.commonsoauth2.RevokeRequest*

Performs a request to *'/services/oauth2/revoke'*

New in version 1.0.0.

class SalesforcePy.sfdc.**Query** (*session_id, instance_url, query_string, **kwargs*)
 Bases: *SalesforcePy.commonsoauth2.BaseRequest*

Performs a request to *'/services/data/vX.XX/query/'*

New in version 1.0.0.

class SalesforcePy.sfdc.**QueryMore** (*session_id, instance_url, query_string, **kwargs*)
 Bases: *SalesforcePy.commonsoauth2.BaseRequest*

Performs recursive requests to *'/services/data/vX.XX/query/'* when there are multiple batches to process.

New in version 1.0.0.

request (**args*)

Makes a *Query* request for the initial query string, then calls itself recursively to request all remaining batches, if there are any. This method will break the recursion and return all when the last batch processed contains a *done* value equal to *True*.

param results: All queried batch results obtained in prior recursions of this method.

type results: [dict]

return A list of dicts where each dict is a batch of query results

rtype [dict]

class SalesforcePy.sfdc.**SObjectBlob** (*_client, service, http_method*)
 Bases: *SalesforcePy.commonsoauth2.BaseRequest*

Perform a request to *'/services/data/vX.XX/sobjects'* where file i/o is necessary.

New in version 1.0.0.

request ()

Returns the request response.

Returns response

Return type dict

set_request_body (***kwargs*)

Creates binary request body by merging *entity, json_body, file_content_type, field, filename* and *content* from ***kwargs* into a binary body template. Sets *request_body* instance variable with the result.

Note: *content* can be either a *file* or a raw value.

param ***kwargs*:

type ***kwargs*: string

class SalesforcePy.sfdc.**SObjectController** (*_client, object_type, _id, binary_field, api_version, external_id*)

Bases: object

A special class that controls insert/update/delete/query/describe of SObject resources.

New in version 1.0.0.

delete (*args, **function_kwarg)

describe (*args, **function_kwarg)

describe_global (*args, **function_kwarg)

get_service ()

Returns the correct sobject service depending on whether the countrroller contains an *id* instance variable

Returns service

Return type string

insert (*args, **function_kwarg)

query (*args, **function_kwarg)

update (*args, **function_kwarg)

upsert (*args, **function_kwarg)

class SalesforcePy.sfdc.**SObjects** (_client, **kwargs)

Bases: *SalesforcePy.common.BaseRequest*

Perform a request to *'/services/data/vX.XX/sobjects'*

New in version 1.0.0.

request ()

Makes the appropriate request depending on the *http_method*. Supported now are: *'GET'*, *'POST'*, *'PATCH'*, and *'DELETE'*. Returns request response.

Note: As successful *'PATCH'* and *'DELETE'* responses return *NO CONTENT*, this method will return *None*. It may be advisable to check the *status* of the *SObject* instance returned as an additional factor in determining whether the request succeeded.

return response dict

rtype dict|None

class SalesforcePy.sfdc.**Search** (session_id, instance_url, search_string, **kwargs)

Bases: *SalesforcePy.common.BaseRequest*

Performs a request to *'/services/data/vX.XX/search/'*

New in version 1.0.0.

SalesforcePy.sfdc.**client** (username, password, client_id, client_secret, **kwargs)

Builds a *Client* and returns it.

New in version 1.0.0.

Note: if any of the required parameters are missing, a *ValueError* will be raised.

Parameters

***username** (*string*) Salesforce username.

***password** (*string*) Salesforce password.

***client_id** (*string*) Salesforce client ID.

***client_secret** (*string*) Salesforce client secret.

****kwargs** kwargs (see below)

Keyword Arguments

***login_url (string)** Salesforce login URL without protocol

Default: `'login.salesforce.com'`

***protocol (string)** Protocol (future use)

***proxies (dict)** A dict containing proxies to be used by *requests* module.

Example: `{“https”: “example.org:443”}`

Default: *None*

***timeout (‘string’)** Tell Requests to stop waiting for a response after a given number of seconds

returns client

rtype Client

raises ValueError

2.6.5 SalesforcePy.wave module

New in version 1.0.0.

class SalesforcePy.wave.**Wave** (*client*)

Bases: *SalesforcePy.common.ApiNamespace*

The Wave namespace class from which all API calls to a Salesforce organisation are made.

New in version 1.0.0.

dataset (*args, **function_kwarg)

query (*args, **function_kwarg)

class SalesforcePy.wave.**WaveDataSet** (*session_id, instance_url, api_name, **kwargs*)

Bases: *SalesforcePy.common.BaseRequest*

Performs a request to `‘/services/data/<api_version>/wave/datasets/<api_name>’`

New in version 1.0.0.

class SalesforcePy.wave.**WaveQuery** (*session_id, instance_url, query, **kwargs*)

Bases: *SalesforcePy.common.BaseRequest*

Performs a request to `‘/services/data/<api_version>/wave/query’`

New in version 1.0.0.

CHAPTER 3

Indices and tables

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

C

`chatter`, [14](#)
`commons`, [14](#)

j

`jobs`, [16](#)

S

`SalesforcePy.chatter`, [14](#)
`SalesforcePy.commons`, [14](#)
`SalesforcePy.jobs`, [16](#)
`SalesforcePy.sfdc`, [17](#)
`SalesforcePy.wave`, [21](#)
`sfdc`, [17](#)

W

`wave`, [21](#)

A

ApiNamespace (class in SalesforcePy.common), 14
 ApprovalProcess (class in SalesforcePy.sfdc), 17
 approvals () (SalesforcePy.sfdc.Client method), 18

B

BaseRequest (class in SalesforcePy.common), 14
 Batches (class in SalesforcePy.jobs), 16
 batches () (SalesforcePy.jobs.Ingest method), 17

C

Chatter (class in SalesforcePy.chatter), 14
 chatter (module), 14
 ChatterFeedComment (class in SalesforcePy.chatter), 14
 ChatterFeedItem (class in SalesforcePy.chatter), 14
 Client (class in SalesforcePy.sfdc), 17
 client () (in module SalesforcePy.sfdc), 20
 commons (module), 14
 create () (SalesforcePy.jobs.Ingest method), 17
 CreateJob (class in SalesforcePy.jobs), 16

D

dataset () (SalesforcePy.wave.Wave method), 21
 debug () (SalesforcePy.sfdc.Client method), 18
 delete () (SalesforcePy.jobs.Ingest method), 17
 delete () (SalesforcePy.sfdc.SObjectController method), 19
 delete_request () (in module SalesforcePy.common), 15
 DeleteJob (class in SalesforcePy.jobs), 17
 describe () (SalesforcePy.sfdc.SObjectController method), 20
 describe_global () (SalesforcePy.sfdc.SObjectController method), 20

E

execute_anonymous () (SalesforcePy.sfdc.Client method), 18

ExecuteAnonymous (class in SalesforcePy.sfdc), 18

F

feed_comment () (SalesforcePy.chatter.Chatter method), 14
 feed_item () (SalesforcePy.chatter.Chatter method), 14

G

get () (SalesforcePy.jobs.Ingest method), 17
 get_headers () (SalesforcePy.common.BaseRequest method), 15
 get_payload () (SalesforcePy.sfdc.Login method), 18
 get_request () (in module SalesforcePy.common), 16
 get_request_url () (SalesforcePy.common.BaseRequest method), 15
 get_request_url () (SalesforcePy.common.OAuthRequest method), 15
 get_request_vars () (SalesforcePy.common.BaseRequest method), 15
 get_service () (SalesforcePy.sfdc.SObjectController method), 20
 get_session_id () (SalesforcePy.sfdc.Login method), 18
 GetJob (class in SalesforcePy.jobs), 17

I

Ingest (class in SalesforcePy.jobs), 17
 insert () (SalesforcePy.sfdc.SObjectController method), 20

J

Jobs (class in SalesforcePy.jobs), 17
 jobs (module), 16

K

kwargs_adder () (in module SalesforcePy.common), 16

L

Login (*class in SalesforcePy.sfdc*), 18
login () (*SalesforcePy.sfdc.Client method*), 18
LoginException, 18
Logout (*class in SalesforcePy.sfdc*), 19
logout () (*SalesforcePy.sfdc.Client method*), 18

O

OAuthRequest (*class in SalesforcePy.commons*), 15

P

patch_request () (*in module SalesforcePy.commons*), 16
post_request () (*in module SalesforcePy.commons*), 16
put_request () (*in module SalesforcePy.commons*), 16

Q

Query (*class in SalesforcePy.sfdc*), 19
query () (*SalesforcePy.sfdc.Client method*), 18
query () (*SalesforcePy.sfdc.SObjectController method*), 20
query () (*SalesforcePy.wave.Wave method*), 21
query_more () (*SalesforcePy.sfdc.Client method*), 18
QueryMore (*class in SalesforcePy.sfdc*), 19

R

request () (*SalesforcePy.commons.BaseRequest method*), 15
request () (*SalesforcePy.commons.OAuthRequest method*), 15
request () (*SalesforcePy.sfdc.Login method*), 18
request () (*SalesforcePy.sfdc.QueryMore method*), 19
request () (*SalesforcePy.sfdc.SObjectBlob method*), 19
request () (*SalesforcePy.sfdc.SObjects method*), 20

S

SalesforcePy.chatter (*module*), 14
SalesforcePy.commons (*module*), 14
SalesforcePy.jobs (*module*), 16
SalesforcePy.sfdc (*module*), 17
SalesforcePy.wave (*module*), 21
Search (*class in SalesforcePy.sfdc*), 20
search () (*SalesforcePy.sfdc.Client method*), 18
set_api_version () (*SalesforcePy.sfdc.Client method*), 18
set_instance_url () (*SalesforcePy.sfdc.Client method*), 18
set_request_body () (*SalesforcePy.sfdc.SObjectBlob method*), 19
sfdc (*module*), 17

SFDCRequestException, 15

SObjectBlob (*class in SalesforcePy.sfdc*), 19
SObjectController (*class in SalesforcePy.sfdc*), 19
SObjects (*class in SalesforcePy.sfdc*), 20
subjects () (*SalesforcePy.sfdc.Client method*), 18

U

update () (*SalesforcePy.jobs.Ingest method*), 17
update () (*SalesforcePy.sfdc.SObjectController method*), 20
UpdateJob (*class in SalesforcePy.jobs*), 17
upsert () (*SalesforcePy.sfdc.SObjectController method*), 20

W

Wave (*class in SalesforcePy.wave*), 21
wave (*module*), 21
WaveDataSet (*class in SalesforcePy.wave*), 21
WaveQuery (*class in SalesforcePy.wave*), 21