
python-oauth10a Documentation

Release

Tim Sheerman-Chase et al.

Jan 25, 2018

Contents:

1	Indices and tables	5
	Python Module Index	7

The MIT License

Copyright (c) 2007-2010 Leah Culver, Joe Stump, Mark Paschal, Vic Fryzel

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

class `oauth10a.Client` (*consumer*, *token=None*, ***kwargs*)

OAuthClient is a worker to attempt to execute a request.

class `oauth10a.Consumer` (*key*, *secret*)

A consumer of OAuth-protected services.

The OAuth consumer is a “third-party” service that wants to access protected resources from an OAuth service provider on behalf of an end user. It’s kind of the OAuth client.

Usually a consumer must be registered with the service provider by the developer of the consumer software. As part of that process, the service provider gives the consumer a *key* and a *secret* with which the consumer software can identify itself to the service. The consumer will include its key in each request to identify itself, but will use its secret only when signing requests, to prove that the request is from that particular registered consumer.

Once registered, the consumer can then use its consumer credentials to ask the service provider for a request token, kicking off the OAuth authorization process.

exception `oauth10a.Error` (*message='OAuth error occurred.'*)

Generic exception class.

message

A hack to get around the deprecation errors in 2.6.

class `oauth10a.Request` (*method='GET'*, *url=None*, *parameters=None*, *body=""*, *is_form_encoded=False*)

The parameters and information for an HTTP request, suitable for authorizing with OAuth credentials.

When a consumer wants to access a service’s protected resources, it does so using a signed HTTP request identifying itself (the consumer) with its key, and providing an access token authorized by the end user to access those resources.

classmethod `from_request` (*http_method*, *http_url*, *headers=None*, *parameters=None*, *query_string=None*)

Combines multiple parameter sources.

get_nonoauth_parameters ()

Get any non-OAuth parameters.

get_normalized_parameters ()

Return a string that contains the parameters that must be signed.

classmethod `make_nonce` ()

Generate pseudorandom number.

classmethod **make_timestamp** ()
Get seconds since epoch (UTC).

sign_request (*signature_method*, *consumer*, *token*)
Set the signature parameter to the result of sign.

to_header (*realm*=")
Serialize as a header for an HTTPAuth request.

to_postdata ()
Serialize as post data for a POST request.

to_url ()
Serialize as a URL for a GET request.

class **oauth10a.Server** (*data_store=None*, *signature_methods=None*)
A skeletal implementation of a service provider, providing protected resources to requests from authorized consumers.

This class implements the logic to check requests for authorization. You can use it with your web server or web framework to protect certain resources with OAuth.

authorize_token (*token*, *user*)
Authorize a request token.

build_authenticate_header (*realm*=")
Optional support for the authenticate header.

fetch_access_token (*oauth_request*)
Processes an access_token request and returns the access token on success.

fetch_request_token (*oauth_request*)
Processes a request_token request and returns the request token on success.

get_callback (*oauth_request*)
Get the callback URL.

verify_request (*request*, *consumer*, *token*)
Verifies an api call and checks all the parameters.

class **oauth10a.SignatureMethod**
A way of signing requests.

The OAuth protocol lets consumers and service providers pick a way to sign requests. This interface shows the methods expected by the other *oauth* modules for signing requests. Subclass it and implement its methods to provide a new way to sign requests.

check (*request*, *consumer*, *token*, *signature*)
Returns whether the given signature is the correct signature for the given consumer and token signing the given request.

sign (*request*, *consumer*, *token*)
Returns the signature for the given request, based on the consumer and token also provided.

You should use your implementation of *signing_base*() to build the message to sign. Otherwise it may be less useful for debugging.

signing_base (*request*, *consumer*, *token*)
Calculates the string that needs to be signed.

This method returns a 2-tuple containing the starting key for the signing and the message to be signed. The latter may be used in error messages to help clients debug their software.

class `oauth10a.Token` (*key*, *secret*)

An OAuth credential used to request authorization or a protected resource.

Tokens in OAuth comprise a *key* and a *secret*. The key is included in requests to identify the token being used, but the secret is used only in the signature, to prove that the requester is who the server gave the token to.

When first negotiating the authorization, the consumer asks for a *request token* that the live user authorizes with the service provider. The consumer then exchanges the request token for an *access token* that can be used to access protected resources.

static from_string (*s*)

Deserializes a token from a string like one returned by `to_string()`.

to_string ()

Returns this token as a plain string, suitable for storage.

The resulting string includes the token's secret, so you should never send or store this string where a third party can read it.

`oauth10a.build_authenticate_header` (*realm=""*)

Optional WWW-Authenticate header (401 error)

`oauth10a.build_xoauth_string` (*url*, *consumer*, *token=None*)

Build an XOAUTH string for use in SMTP/IMPA authentication.

`oauth10a.escape` (*s*)

Escape a URL including any `/`.

`oauth10a.generate_nonce` (*length=8*)

Generate pseudo-random number.

`oauth10a.generate_timestamp` ()

Get seconds since epoch (UTC).

`oauth10a.generate_verifier` (*length=8*)

Generate pseudo-random number.

`oauth10a.to_unicode` (*s*)

Convert to unicode, raise exception with instructive error message if *s* is not unicode, ascii, or utf-8.

`oauth10a.to_unicode_optional_iterator` (*x*)

Raise `TypeError` if *x* is a str containing non-utf8 bytes or if *x* is an iterable which contains such a str.

`oauth10a.to_utf8_optional_iterator` (*x*)

Raise `TypeError` if *x* is a str or if *x* is an iterable which contains a str.

CHAPTER 1

Indices and tables

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

O

oauth10a, [1](#)

A

authorize_token() (oauth10a.Server method), 2

B

build_authenticate_header() (in module oauth10a), 3
build_authenticate_header() (oauth10a.Server method), 2
build_xoauth_string() (in module oauth10a), 3

C

check() (oauth10a.SignatureMethod method), 2
Client (class in oauth10a), 1
Consumer (class in oauth10a), 1

E

Error, 1
escape() (in module oauth10a), 3

F

fetch_access_token() (oauth10a.Server method), 2
fetch_request_token() (oauth10a.Server method), 2
from_request() (oauth10a.Request class method), 1
from_string() (oauth10a.Token static method), 3

G

generate_nonce() (in module oauth10a), 3
generate_timestamp() (in module oauth10a), 3
generate_verifier() (in module oauth10a), 3
get_callback() (oauth10a.Server method), 2
get_nonoauth_parameters() (oauth10a.Request method), 1
get_normalized_parameters() (oauth10a.Request method), 1

M

make_nonce() (oauth10a.Request class method), 1
make_timestamp() (oauth10a.Request class method), 1
message (oauth10a.Error attribute), 1

O

oauth10a (module), 1

R

Request (class in oauth10a), 1

S

Server (class in oauth10a), 2
sign() (oauth10a.SignatureMethod method), 2
sign_request() (oauth10a.Request method), 2
SignatureMethod (class in oauth10a), 2
signing_base() (oauth10a.SignatureMethod method), 2

T

to_header() (oauth10a.Request method), 2
to_postdata() (oauth10a.Request method), 2
to_string() (oauth10a.Token method), 3
to_unicode() (in module oauth10a), 3
to_unicode_optional_iterator() (in module oauth10a), 3
to_url() (oauth10a.Request method), 2
to_utf8_optional_iterator() (in module oauth10a), 3
Token (class in oauth10a), 2

V

verify_request() (oauth10a.Server method), 2