
simplenote.py Documentation

Release 2.1.0

Daniel Schauenberg

Nov 04, 2018

Contents

1	Indices and tables	3
1.1	Quickstart	3
1.2	API Reference	3
	Python Module Index	7

Contents:

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

1.1 Quickstart

`simplenote.py` is a simple wrapper around the `simplenote.com` web service. It provides the `Simplenote` object, with a set of convenience methods to interact with the service.

First import the module and create an object:

```
import simplenote
sn = simplenote.Simplenote(user, password)
```

This object then provides the following API methods:

```
sn.get_note_list(tags=<Optional list of tags to filter by>)
sn.get_note(note_id, version=<Optional integer version number of note to fetch>)
sn.add_note(note)
sn.update_note(note)
sn.trash_note(note_id)
sn.delete_note(note_id)
```

A `note` object is a dictionary with at least a `content` property, containing the note text. The `update_note` method needs a `note` object which also has a `key` property.

1.2 API Reference

If you are looking for information on a specific function, class or method, you can most likely find it here.

1.2.1 API

This chapter covers all API interfaces of the simplenote module.

Historical Simplenote API - Note Object

Prior to the Simperium API a complete note *dict* object contained the following fields:

```
{
  key          : (string, note identifier, created by server),
  deleted      : (bool, whether or not note is in trash),
  modifydate   : (last modified date, in seconds since epoch),
  createdate   : (note created date, in seconds since epoch),
  syncnum      : (integer, number set by server, track note changes),
  version      : (integer, number set by server, track note content changes),
  minversion   : (integer, number set by server, minimum version available for note),
  sharekey     : (string, shared note identifier),
  publishkey   : (string, published note identifier),
  systemtags   : [(Array of strings, some set by server)],
  tags         : [(Array of strings)],
  content      : (string, data content)
}
```

Simperium API - Note Object

Under Simperium some of the fields were renamed and some were removed. String data also seems to be UTF-8 by default. A Simperium note object looks like this:

```
{
  deleted          : (bool, whether or not note is in trash),
  modificationDate : (last modified date, in seconds since epoch),
  createDate       : (note created date, in seconds since epoch),
  version          : (integer, number set by server, track note content changes),
  shareURL         : (string, shared url),
  publishURL       : (string, published note url),
  systemTags       : [(Array of strings, some set by server)],
  tags             : [(Array of strings)],
  content          : (string, data content)
}
```

It no longer includes the “key” (actually now an “id”, but still not included in the note object).

However, Simplenote.py tries to work as a drop in replacement for code that expects the older fields and therefore you can still use the following:

```
{
  key          : (string, note identifier, created by server),
  deleted      : (bool, whether or not note is in trash),
  modifydate   : (last modified date, in seconds since epoch),
  createdate   : (note created date, in seconds since epoch),
  version      : (integer, number set by server, track note content changes),
  systemtags   : [(Array of strings, some set by server)],
  tags         : [(Array of strings)],
  content      : (string, data content)
}
```


And simplenote.py will handle conversion to/from the Simperium fields.

Simplenote main class

class simplenote.**Simplenote**(*username, password*)

Class for interacting with the simplenote web service

add_note(*note*)

Wrapper method to add a note

The method can be passed the note as a dict with the *content* property set, which is then directly send to the web service for creation. Alternatively, only the body as string can also be passed. In this case the parameter is used as *content* for the new note.

Arguments:

- *note* (dict or string): the note to add

Returns: A tuple (*note, status*)

- *note* (dict): the newly created note
- *status* (int): 0 on success and -1 otherwise

authenticate(*user, password*)

Method to get simplenote auth token

Arguments:

- *user* (string): simplenote email address
- *password* (string): simplenote password

Returns: Simplenote API token as string

delete_note(*note_id*)

Method to permanently delete a note

Arguments:

- *note_id* (string): key of the note to trash

Returns: A tuple (*note, status*)

- *note* (dict): an empty dict or an error message
- *status* (int): 0 on success and -1 otherwise

get_note(*noteid, version=None*)

Method to get a specific note

Arguments:

- *noteid* (string): ID of the note to get
- *version* (int): optional version of the note to get

Returns: A tuple (*note, status*)

- *note* (dict): note object
- *status* (int): 0 on success and -1 otherwise

get_note_list (*data=True, since=None, tags=[]*)

Method to get the note list

The method can be passed optional arguments to limit the list to notes containing a certain tag, or only updated since a certain Simperium cursor. If omitted a list of all notes is returned.

By default data objects are returned. If data is set to false only keys/ids and versions are returned. An empty data object is inserted for compatibility.

Arguments:

- tags=[] list of tags as string: return notes that have at least one of these tags
- since=cursor Simperium cursor as string: return only changes since this cursor
- data=True If false only return keys/ids and versions

Returns: A tuple (*notes, status*)

- notes (list): A list of note objects with all properties set except *content*.
- status (int): 0 on success and -1 otherwise

get_token ()

Method to retrieve an auth token.

The cached global token is looked up and returned if it exists. If it is *None* a new one is requested and returned.

Returns: Simplenote API token as string

trash_note (*note_id*)

Method to move a note to the trash

Arguments:

- note_id (string): key of the note to trash

Returns: A tuple (*note, status*)

- note (dict): the newly created note or an error message
- status (int): 0 on success and -1 otherwise

update_note (*note*)

Method to update a specific note object, if the note object does not have a “key” field, a new note is created

Arguments

- note (dict): note object to update

Returns: A tuple (*note, status*) - note (dict): note object - status (int): 0 on success and -1 otherwise

S

`simplenote`, 4

A

`add_note()` (`simplenote.Simplenote` method), [5](#)
`authenticate()` (`simplenote.Simplenote` method), [5](#)

D

`delete_note()` (`simplenote.Simplenote` method), [5](#)

G

`get_note()` (`simplenote.Simplenote` method), [5](#)
`get_note_list()` (`simplenote.Simplenote` method), [5](#)
`get_token()` (`simplenote.Simplenote` method), [6](#)

S

`Simplenote` (class in `simplenote`), [5](#)
`simplenote` (module), [4](#)

T

`trash_note()` (`simplenote.Simplenote` method), [6](#)

U

`update_note()` (`simplenote.Simplenote` method), [6](#)