

First of all, thanks for your participation in the Simplenote API community. As of July 2012, there are almost 700,000 Simplenote users and growing. The robust 3rd party app support has been really important to the growth of the app so thank you for your contributions.

Background

The current Simplenote syncing is done through App Engine (simple-note.appspot.com), for the past year, we have been developing Simperium, an improved and generalized API that is based on the original Simplenote API, and transitioned all Simplenote users to use the new system a few months ago. That means the current API thru App Engine is serving as a proxy to Simperium. Writes to create or modify notes are first sent to Simperium then mirrored on App Engine. Read operations (index or retrieving specific notes) are handled by App Engine and not proxied to Simperium.

More details about the API can be found at the [Simperium HTTP API Reference](#).

Migration

Authentication

Security in Simperium is handled via access tokens. Every API call to modify data (calls to `api.simperium.com`) requires an access token. To obtain an access token, you'll need to authenticate the user via username and password. This is done via `auth.simperium.com` and calls to that server require an API key passed in the header.

An API key will be issued for each app so please contact us (fred@simperium.com) to get an API key for your app.

Getting a token

To get a token, you need to send an HTTP POST with a JSON payload containing the username and password of the user to:

```
https://auth.simperium.com/1/chalk-bump-f49/authorize/
```

You'll need to include the API key for your app (issued by Simperium) in a `X-Simperium-API-Key` header for this request:

```
$ curl -H 'X-Simperium-API-Key: <issued api key>' https://auth.simperium.com/1/chalk-bump-f49/authorize/ -d '{
  "username": "test@test.com", "password": "test"}'

{"username": "test@test.com", "access_token": "307cfe06115541a3833f3d79832ebd53", "userid": "e6e32fb2237a5c0c14913bb8a1398aee"}
```

The access token is returned in the JSON response. For all subsequent calls to `api.simperium.com` you'll need to include this token in the header `X-Simperium-Token`.

Notes

The major changes to note objects are new field names and now version/syncnum is handled separately from the data fields. The “version” field used to refer specifically to the content version, and “syncnum” to the overall object. Therefore it was possible for syncnum to change, but the note content/version would stay the same (for example, if only the tags field changed). In the new Simperium API, any change to any field will increment the Simperium version. The version is now returned in the index and in a header if you retrieve a note object individually.

Note object fields

Simplenote API2	Simperium	Value type
key	(Simperium id, no longer in object)	string
deleted	deleted	boolean: true, false
modifydate	modificationDate	float (epoch time in seconds)
createdate	creationDate	float (epoch time in seconds)
syncnum	(Simperium version, no longer in object)	n/a
version	(Simperium version, no longer in object)	n/a
minversion	(no longer used)	n/a
systemtags	systemTags	string array
tags	tags	string array
sharekey	shareURL	string
publishkey	publishURL	string
content	content	string

Modifying data in Simperium

Ids

Simperium relies on clients to generate their own ids, as such, UUIDs are suggested for object ids to minimize the chance of collision. The valid id characters are: `a-z0-9_.-%`

Versions

All objects in Simperium have a version number that is incremented anytime the data is changed. When modifying data, you should indicate which version you are modifying. If you've retrieved version 10 of an object for example, and you make local edits, when you send these edits back to Simperium, you should indicate that these changes were based on version 10. After a successful modification, the current version will be returned in the header `X-Simperium-Version`.

clientid and ccid

When making edits there are two identifiers that are always sent, `clientid` and `ccid`. `clientid` is like a session identifier and you can choose any string for your client. You should maintain and re-use the same string over the lifetime of a session. A `ccid` must also be sent, and should be a unique string that changes for every operation that you perform. For example, if you are sending a change for an object, you should keep using the same `ccid` until you get a response from the server (200 for success or 4xx) that indicates it has been processed. 5xx errors or no response means you should keep using the same `ccid`. The combination of `clientid` and `ccid` prevents duplicate operations from getting processed and allow better error tracking throughout the system.

Creating or modifying a note

Key differences when creating a note:

- The note id must be specified, we recommend generating a UUID for this, valid id characters are: `a-z0-9_.-%`
- All 8 fields must be sent. When modifying, you can just include the fields which have changed

HTTP POST with JSON payload to:

```
https://api.simperium.com/1/chalk-bump-f49/Note/i/(note id)/v/(note version)?clientid=(a session id)&ccid=(a unique operation id)
```

Create a note:

```
$ curl -iH "X-Simperium-Token: 307cfe06115541a3833f3d79832ebd53" https://api.simperium.com/1/chalk-bump-f49/Note/i/b5c87056be3e4950993f5140c1a28a5a/v/1?clientid=sample-f16a41bc3&ccid=fe600ca9b5a644cdaa6685c913569167 -d '{"tags": [], "deleted": false, "shareURL": "", "publishURL": "", "content": "new note!", "systemTags": [], "modificationDate": 1335390338.091436, "creationDate": 1335390338.091453}'

HTTP/1.1 200 OK
X-Simperium-Version: 1
```

When modifying a note, you should include the version that you are modifying in the URL by appending `/v/(version number)`. In the body, you can include only the fields that have changed. In this example, we update `content` and `modificationDate`.

```
$ curl -iH "X-Simperium-Token: 307cfe06115541a3833f3d79832ebd53" https://api.simperium.com/1/chalk-bump-f49/Note/i/b5c87056be3e4950993f5140c1a28a5a/v/1?clientid=sample-f16a41bc3&ccid=78422d1ad0224c5f8a67832cd8ed0bfb -d '{"modificationDate":1341563318.0, "content": "updated note with new modification time"}'

HTTP/1.1 200 OK
X-Simperium-Version: 2
```

Retrieving a note

To retrieve a note, just send an HTTP GET to the same URL:

```
$ curl -iH "X-Simperium-Token: 307cfe06115541a3833f3d79832ebd53" https://api.simperium.com/1/chalk-bump-f49/Note/i/b5c87056be3e4950993f5140c1a28a5a/v/2?clientid=sample-f16a41bc3&ccid=5b2dcfd8d18d49e09250c01271da6c62 -d '{"tags": [], "deleted": false, "shareURL": "", "systemTags": [], "content": "updated note with new modification time", "publishURL": "", "modificationDate": 1341563318.0, "creationDate": 1335390338.091453}'

HTTP/1.1 200 OK
X-Simperium-Version: 2
Content-Length: 205
Content-Type: application/json; charset=UTF-8
```

We can also retrieve a specific version by appending the version number:

```
$ curl -iH "X-Simperium-Token: 307cfe06115541a3833f3d79832ebd53" https://api.simperium.com/1/chalk-bump-f49/Note/i/b5c87056be3e4950993f5140c1a28a5a/v/1?clientid=sample-f16a41bc3&ccid=db7d26460f7e443da91dd5f2351a3b06

HTTP/1.1 200 OK
X-Simperium-Version: 1
Content-Length: 180
Content-Type: application/json; charset=UTF-8
```

```
{"tags": [], "deleted": false, "shareURL": "", "systemTags": [], "content": "new note!", "publishURL": "", "modificationDate": 1335390338.091436, "creationDate": 1335390338.091453}
```

Sending a note to the Trash:

Sending a note to the Trash is just updating it's `deleted` property to be `true`. Restoring a note from the Trash would mean updating `deleted` to `false`.

```
$ curl -iH "X-Simperium-Token: 307cfe06115541a3833f3d79832ebd53" https://api.simperium.com/1/chalk-bump-f49/Note/i/b5c87056be3e4950993f5140c1a28a5a/v/2?clientid=sample-f16a41bc3&ccid=5b2dcfd8d18d49e09250c01271da6c62 -d '{"deleted":true}'

HTTP/1.1 200 OK
X-Simperium-Version: 3
```

Permanently delete a note

To permanently delete a note, send an HTTP DELETE to the note URL. After this, the note will no longer show up in the index and cannot be retrieved.

```
$ curl -iH "X-Simperium-Token: 307cfe06115541a3833f3d79832ebd53" -X DELETE https://api.simperium.com/1/chalk-bump-f49/Note/i/b5c87056be3e4950993f5140c1a28a5a/v/3?clientid=sample-f16a41bc3&ccid=db7d26460f7e443da91dd5f2351a3b06

HTTP/1.1 200 OK
X-Simperium-Version: 4
```

Index

To retrieve the index, send an HTTP GET request to:

```
https://api.simperium.com/1/chalk-bump-f49/Note/index?mark=<optional mark>&limit=<optional max number of note objects to return>
```

This shows two notes, one at version 2 and the other at version 1:

```
$ curl -iH "X-Simperium-Token: 307cfe06115541a3833f3d79832ebd53" https://api.simperium.com/1/chalk-bump-f49/Note/index

HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 154
```

```
{
  "current": "4ff76c0cb829e9e5b0000544",
  "index":
    [
      {
        "id": "c98641c0c3944ab19f7b79ef93f2c3ee", "v": 2},
      {
        "id": "b93a9cba98404aad852bbd78b65dde8d", "v": 1}
    ]
}
```

The `index` field is an array of objects which will each have the `id` and the version, `v` of the object. If there are more than 1000 objects in the page, a `mark` parameter will be returned which can be used to query subsequent pages. The `current` field is a marker used to refer to the most recent change, if this value stays the same across subsequent queries, then no change has been made to any object.

By setting `limit` to 1 we can show how to use `mark`:

```
$ curl -iH "X-Simperium-Token: 307cfe06115541a3833f3d79832ebd53" https://api.simperium.com/1/chalk-bump-f49/Note/index?limit=1

HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 138
```

```
{
  "current": "4ff76c0cb829e9e5b0000544",
  "index":
    [
      {
        "id": "c98641c0c3944ab19f7b79ef93f2c3ee", "v": 2}
    ],
  "mark": "4ff76c0cb829e9e5b0000544"
}
```

```
$ curl -iH "X-Simperium-Token: 307cfe06115541a3833f3d79832ebd53" https://api.simperium.com/1/chalk-bump-f49/Note/index?mark=4ff76c0cb829e9e5b0000544

HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 102
```

```
{
  "current": "4ff76c0cb829e9e5b0000544",
  "index":
    [
      {
        "id": "b93a9cba98404aad852bbd78b65dde8d", "v": 1}
    ]
}
```