

---

# **deluge-webapi Documentation**

*Release 0.2.1*

**Igor 'idle sign' Starikov**

**Oct 28, 2018**



---

## Contents

---

<b>1</b>	<b>Description</b>	<b>3</b>
<b>2</b>	<b>Requirements</b>	<b>5</b>
<b>3</b>	<b>Installation</b>	<b>7</b>
<b>4</b>	<b>Table of Contents</b>	<b>9</b>
4.1	Quick Reference . . . . .	9
<b>5</b>	<b>Get involved into deluge-webapi</b>	<b>11</b>
<b>6</b>	<b>More things</b>	<b>13</b>



<https://github.com/idlesign/deluge-webapi>



# CHAPTER 1

---

## Description

---

*Plugin for Deluge WebUI providing sane JSON API.*

Exposes some sane JSON API through Deluge WebUI (missing out of the box or restyled) so that you can manipulate Deluge from your programs using HTTP requests.





## CHAPTER 2

---

### Requirements

---

1. Python 2.7+
2. Deluge 1.3.15+



## CHAPTER 3

---

### Installation

---

- Get plugin egg file, from `dist/` (<https://github.com/idlesign/deluge-webapi/tree/master/dist>) directory.
- Open Deluge Web UI, go to “Preferences -> Plugins -> Install plugin” and choose egg file.

---

**Note:** To build .egg file from source code yourself use `python setup.py bdist_egg` command in source code directory.

---

- Activate *WebAPI* plugin.
- You're ready.



### 4.1 Quick Reference

---

**Note:** First make sure you're using WebUI and *WebAPI* plugin is active under "Preferences -> Plugins".

---

#### 4.1.1 Hints

1. WebAPI exists alongside with Deluge built-in JSON API, so all HTTP requests should proceed to *http://localhost:8112/json* (or what ever you've got instead of *localhost*)
2. Use POST method for HTTP requests.
3. Make requests with valid JSON (including 'Accept: application/json' and 'Content-Type: application/json' HTTP headers).
4. JSON must include the following fields:
  - id* - message number for identity (could be any)
  - method* - API method name. E.g.: *auth.login*, *webapi.get\_torrents*
  - params* - API method parameters (depend on method)
5. Login beforehand.
  - Send {"id": 1, "method": "auth.login", "params": ["your\_password\_here"]} and receive cookies with auth information.
  - Use those cookies for every request.
6. API answers with an error with the following JSON:

```
{"error": "Some error description.", "id": 1, "result": False}
```

Check responses for *error* field.

7. Make sure Deluge WebUI is connected to Deluge daemon that makes actual torrent processing.
  - Send `{“id”: 1, “method”: “auth.check_session”, “params”: []}` and verify no error.
8. WebAPI method names start with *webapi.*. (E.g.: *webapi.add\_torrent* to call *add\_torrent* function).

### 4.1.2 API Methods

---

**Note:** WebAPI uses torrent hashes to identify torrents, so torrent ID is the same as hash.

---

#### Get torrents info

*get\_torrents(ids=None, params=None)*

Returns information about all or a definite torrent. Returned information can be filtered by supplying wanted parameter names.

```
{“id”: 1, “method”: “webapi.get_torrents”, “params”: [[“torrent_hash1”, “torrent_hash2”], [“name”, “comment”]]}
```

#### Add torrent

*add\_torrent(metainfo, options=None)*

Adds a torrent with the given options. *metainfo* could either be base64 torrent data or a magnet link.

```
{“id”: 1, “method”: “webapi.add_torrent”, “params”: [“base64_encoded_torrent_file_contents”, {“download_location”: “/home/idle/downloads/”}]}
```

#### Remove torrent

*remove\_torrent(torrent\_id, remove\_data=False)*

Removes a given torrent. Optionally can remove data.

```
{“id”: 1, “method”: “webapi.remove_torrent”, “params”: [“torrent_hash3”, True]}
```

#### Get API version

*get\_api\_version()*

Returns WebAPI plugin version.

```
{“id”: 1, “method”: “webapi.get_api_version”, “params”: []}
```

---

### Get involved into deluge-webapi

---

**Submit issues.** If you spotted something weird in application behavior or want to propose a feature you can do that at <https://github.com/idlesign/deluge-webapi/issues>

**Write code.** If you are eager to participate in application development, fork it at <https://github.com/idlesign/deluge-webapi>, write your code, whether it should be a bugfix or a feature implementation, and make a pull request right from the forked project page.

**Spread the word.** If you have some tips and tricks or any other words in mind that you think might be of interest for the others — publish it.





## CHAPTER 6

---

### More things

---

- You might be interested in considering other Deluge plugins at <http://dev.deluge-torrent.org/wiki/Plugins/>
- Or you might be interested in a PHP Deluge API wrapper: <https://github.com/kaysond/deluge-php>