

---

# **python-eduvpn-client Documentation**

***Release 0.0dev***

**Gijs Molenaar**

**Feb 21, 2022**



---

## Contents:

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Issues . . . . .	3
1.3	Contributing . . . . .	3
<b>2</b>	<b>API</b>	<b>5</b>
2.1	Remote calls . . . . .	5
2.2	eduVPN OAuth2 . . . . .	7
2.3	Crypto . . . . .	7
2.4	eduVPN IO . . . . .	8
2.5	OpenVPN . . . . .	8
2.6	Exceptions . . . . .	8
<b>3</b>	<b>Developer notes</b>	<b>9</b>
3.1	Notes about code . . . . .	9
3.2	Flow schema . . . . .	10
3.3	How to make a release . . . . .	10
<b>4</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Python Module Index</b>	<b>15</b>
	<b>Index</b>	<b>17</b>



# CHAPTER 1

---

## Introduction

---

This is the GNU/Linux desktop client and Python API for eduVPN. The Desktop client only works on Linux, but most parts of the API are usable on other platforms also. For the API Python 2.7, 3.4+ and pypy are supported.

## 1.1 Installation

It is recommended to use a package to install the eduVPN client, but you can also install using pip from py or directly from github. We distribute RPM packages for Fedora, and Deb packages for Debian and Ubuntu.

The eduVPN client has been tested with:

- Debian 9 (Stretch), Debian 10 (Buster)
- Ubuntu 18.04 LTS and 20.04 LTS
- CentOS 7 & 8
- Fedora 30 & 31

---

**Note:** If your target is not supported the client might still, potentially work with limited functionality. You need to have [Network Manager](#) and [OpenVPN 2.4.0+](#) installed.

---

### 1.1.1 Debian and Ubuntu

You can install the latest release on Debian or Ubuntu using the eduVPN packaging repository by running these commands as root or using sudo:

```
$ apt install apt-transport-https curl
$ curl -L https://repo.eduvpn.org/debian/eduvpn.key | apt-key add -
$ echo "deb https://repo.eduvpn.org/debian/ stretch main" > /etc/apt/sources.list.d/
  ↳eduvpn.list
```

(continues on next page)

(continued from previous page)

```
$ apt update
$ apt install eduvpn-client
```

This has been tested on Ubuntu 18.04 (Bionic) and Debian 9 (stretch). Unfortunately, Ubuntu 16.04 LTS (Xenial) is **not** supported. Ubuntu Xenial and older are bundled with an outdated and unsupported OpenVPN.

---

**Note:** For Debian Stretch you need to enable the backports repository and manually update network manager. *network-manager-openvpn* >= 1.2.10 is required for *tls-crypt* support. To enable the backports repository add this line to your */etc/apt/sources.list*:

```
deb http://deb.debian.org/debian stretch-backports main contrib non-free
```

And then update network-manager:

```
$ sudo apt-get update
$ sudo apt-get -t stretch-backports install network-manager-openvpn-gnome
```

---

### 1.1.2 Fedora

You can install the latest release of the eduVPN client on Fedora by running these commands as root or using sudo:

```
$ dnf install dnf-plugins-core
$ dnf copr enable gijzelaerr/eduvpn-client
$ dnf install eduvpn-client
```

More information is available at [fedora copr](#).

### 1.1.3 Centos

You can install the latest release of the eduVPN client on Centos 7 or 8 by running these commands as root or using sudo:

```
$ yum install yum-plugin-copr
$ yum copr enable gijzelaerr/eduvpn-client
$ yum install eduvpn-client
```

More information is available at [fedora copr](#).

---

**Note:** Currently CentOS 7 ships with an outdated NetworkManager so some features might not work properly.

---

### 1.1.4 Pip

You can install the client API from pypi:

```
$ pip install eduvpn-client
```

Or if you want to try out the bleeding edge development version:

```
$ pip install git+https://github.com/eduvpn/python-eduvpn-client.git
```

You can install the dependencies for the user interface:

```
$ pip install -e ".[client]"
```

---

**Note:** the project depends on the `python-gi` package, which for now doesn't properly install in a virtualenv. If you do install `python-eduvpn-client` in a virtualenv it is recommended you create the virtualenv using the `--system-site-packages` flag and install the `python-gi` package using your operating system package manager. Read more about this on the [pygobject website](#).

---

If you use eduVPN this way you need to make sure all non-Python dependencies are installed. For Debian or Ubuntu:

```
$ apt install gir1.2-gtk-3.0 gir1.2-notify-0.7 libdbus-1-dev libnotify4 python3-  
↳dateutil \  
    python3-dbus python3-nacl python3-requests-oauthlib python3-gi network-manager-  
↳openvpn \  
    python3-pip git
```

For fedora:

```
$ dnf install -y gtk3 libnotify python3-dateutil python3-networkmanager python3-  
↳pydbus \  
    python3-pynacl python3-requests-oauthlib python3-gobject python3-pip \  
    python3-configparser python3-future git NetworkManager-openvpn NetworkManager-  
↳openvpn-gnome
```

## 1.2 Issues

If you experience any issues you could and should report them at our [issue tracker](#). Please don't forget to mention your OS, method of installation, eduVPN client version and instructions on how to reproduce the problem. If you have a problem enabling your VPN connection please also examine the `journalctl -u NetworkManager` logs.

### 1.2.1 Source code

Development of this project takes place on [github](#). You can find the source code and all releases there.

## 1.3 Contributing

Contributions are more than welcome! If you experience any problems let us know in the bug tracker. We accept patches in the form of github pull requests. Please make sure your code works with python 2 and python3, and is pep8 compatible. Also make sure the test suit actually passes all tests.





This is the documentation for the Python client API for eduVPN.

## 2.1 Remote calls

`eduvpn.remote.check_certificate(oauth, api_base_uri, common_name)`

`eduvpn.remote.create_config(oauth, api_base_uri, display_name, profile_id)`

Create a configuration for a given profile.

### Parameters

- **oauth** (`requests_oauthlib.OAuth2Session`) – oauth2 object
- **api\_base\_uri** (`str`) – the instance base URI
- **display\_name** (`str`) –
- **profile\_id** (`str`) –

`eduvpn.remote.create_keypair(oauth, api_base_uri)`

Create remote keypair and return results

### Parameters

- **oauth** (`requests_oauthlib.OAuth2Session`) – oauth2 object
- **api\_base\_uri** (`str`) – the instance base URI

**Returns** certificate and key

**Return type** tuple(str, str)

`eduvpn.remote.get_auth_url(oauth, code_verifier, auth_endpoint)`

” generate a authorization URL.

### Parameters

- **oauth** (`requests_oauthlib.OAuth2Session`) – oauth2 object

- **code\_verifier** (*str*) –

- **auth\_endpoint** (*str*) –

`eduvpn.remote.get_instance_info(instance_uri, verifier=None)`

Retrieve information from instance

**Parameters**

- **instance\_uri** (*str*) – the base URI for the instance

- **verifier** (*nacl.signing.VerifyKey*) – the verifykey used to verify the key

**Returns** `api_base_uri`, `authorization_endpoint`, `token_endpoint`

**Return type** `tuple(str, str, str)`

`eduvpn.remote.get_instances(discovery_uri, verifier=None)`

retrieve a list of instances.

**Parameters**

- **discovery\_uri** (*str*) – the URL to parse for instances discovery

- **verifier** (*nacl.signing.VerifyKey*) – used to verify the key

**Returns** `display_name`, `base_uri`, `logo_data`

**Return type** `generator`

`eduvpn.remote.get_profile_config(oauth, api_base_uri, profile_id)`

Return a profile configuration

**Parameters**

- **oauth** (*requests\_oauthlib.OAuth2Session*) – oauth2 object

- **api\_base\_uri** (*str*) – the instance base URI

- **profile\_id** (*str*) –

`eduvpn.remote.list_profiles(oauth, api_base_uri)`

List profiles on instance

**Parameters**

- **oauth** (*requests\_oauthlib.OAuth2Session*) – oauth2 object

- **api\_base\_uri** (*str*) – the instance base URI

**Returns** of available profiles on the instance (`display_name`, `profile_id`, `two_factor`)

**Return type** `list`

`eduvpn.remote.system_messages(oauth, api_base_uri)`

Return all system messages

**Parameters**

- **oauth** (*requests\_oauthlib.OAuth2Session*) – oauth2 object

- **api\_base\_uri** (*str*) – the instance base URI

`eduvpn.remote.translate_display_name(display_name)`

Translates a `display_name` in the current locale.

**Parameters** **display\_name** (*str or dict*) –

`eduvpn.remote.two_factor_enroll_totp(oauth, api_base_uri, secret, key)`

`eduvpn.remote.two_factor_enroll_yubi (oauth, api_base_uri, yubi_key_otp)`

`eduvpn.remote.user_info (oauth, api_base_uri)`  
returns the user information

**Parameters**

- **oauth** (*requests\_oauthlib.OAuth2Session*) – oauth2 object
- **api\_base\_uri** (*str*) – the instance base URI

`eduvpn.remote.user_messages (oauth, api_base_uri)`

These are messages specific to the user. It can contain a message about the user being blocked, or other personal messages from the VPN administrator.

**Parameters**

- **oauth** (*requests\_oauthlib.OAuth2Session*) – oauth2 object
- **api\_base\_uri** (*str*) – the instance base URI

**Returns** a list of dicts with `date_time`, `message`, `type` keys

**Return type** list

## 2.2 eduVPN OAuth2

## 2.3 Crypto

`eduvpn.crypto.common_name_from_cert (pem_data)`  
Extract common name from client certificate.

**Parameters** **pem\_data** (*str*) – PEM encoded certificate

**Returns** the common name of the client certificate.

**Return type** str

`eduvpn.crypto.gen_base32 (length=20)`  
Generate a base32 string.

`eduvpn.crypto.gen_code_challenge (code_verifier)`  
Transform the PKCE code verifier in a code challenge.

**Parameters** **code\_verifier** (*str*) – a string generated with `gen_code_verifier()`

`eduvpn.crypto.gen_code_verifier (length=128)`  
Generate a high entropy code verifier, used for PKCE.

**Parameters** **length** (*int*) – length of the code

**Returns**

**Return type** str

`eduvpn.crypto.make_verifier (key)`  
Create a NaCL verifier.

**Parameters** **key** (*str*) – A verification key

**Returns** a nacl verifykey object

**Return type** nacl.signing.VerifyKey

## 2.4 eduVPN IO

Helper functions related to local IO

`eduvpn.io.mkdir_p(path)`

Create a folder with all its parents, like `mkdir -p` :param path: path of directory to create :type path: str

`eduvpn.io.write_cert(content, type_, unique_name)`

Write a certificate to the filesystem :param content: content of certificate file :type content: str :param type: type of certificate file :type type: str :param unique\_name: description of file :type unique\_name: str

**Returns** full path to certificate file

**Return type** str

## 2.5 OpenVPN

`eduvpn.openvpn.format_like_ovpn(config, cert, key)`

create a OVPN format config text

**Parameters**

- **config** (str) – an existing ovpn config
- **cert** (str) –
- **key** (str) –

`eduvpn.openvpn.ovpn_to_nm(config, meta, display_name, username=None)`

Generate a NetworkManager style config dict from a parsed ovpn config dict

**Parameters**

- **config** (dict) – a parsed openvpn config, typically generated by `eduvpn.openvpn.parse_ovpn()`
- **meta** (dict) – configuration metadata
- **display\_name** (str) – the display name of the configuration
- **username** (str) – username to use for 2-factor authentication

`eduvpn.openvpn.parse_ovpn(configtext)`

Parse a ovpn like config file, return it in dict

configtext (str): content of a OpenVPN like config file

## 2.6 Exceptions

**exception** `eduvpn.exceptions.EduvpnAuthException`

Bases: `Exception`

eduVPN authentication exception

**exception** `eduvpn.exceptions.EduvpnException`

Bases: `Exception`

base eduVPN exception

### 3.1 Notes about code

Use `eduvpn.util.thread_helper(lambda: func(arg='arg'))` to schedule long running actions from the UI (main) thread.

Use `GLib.idle_add(lambda: func(arg='arg'))` to schedule UI updates back on the main thread.

Never call GTK functions directly from the background thread.

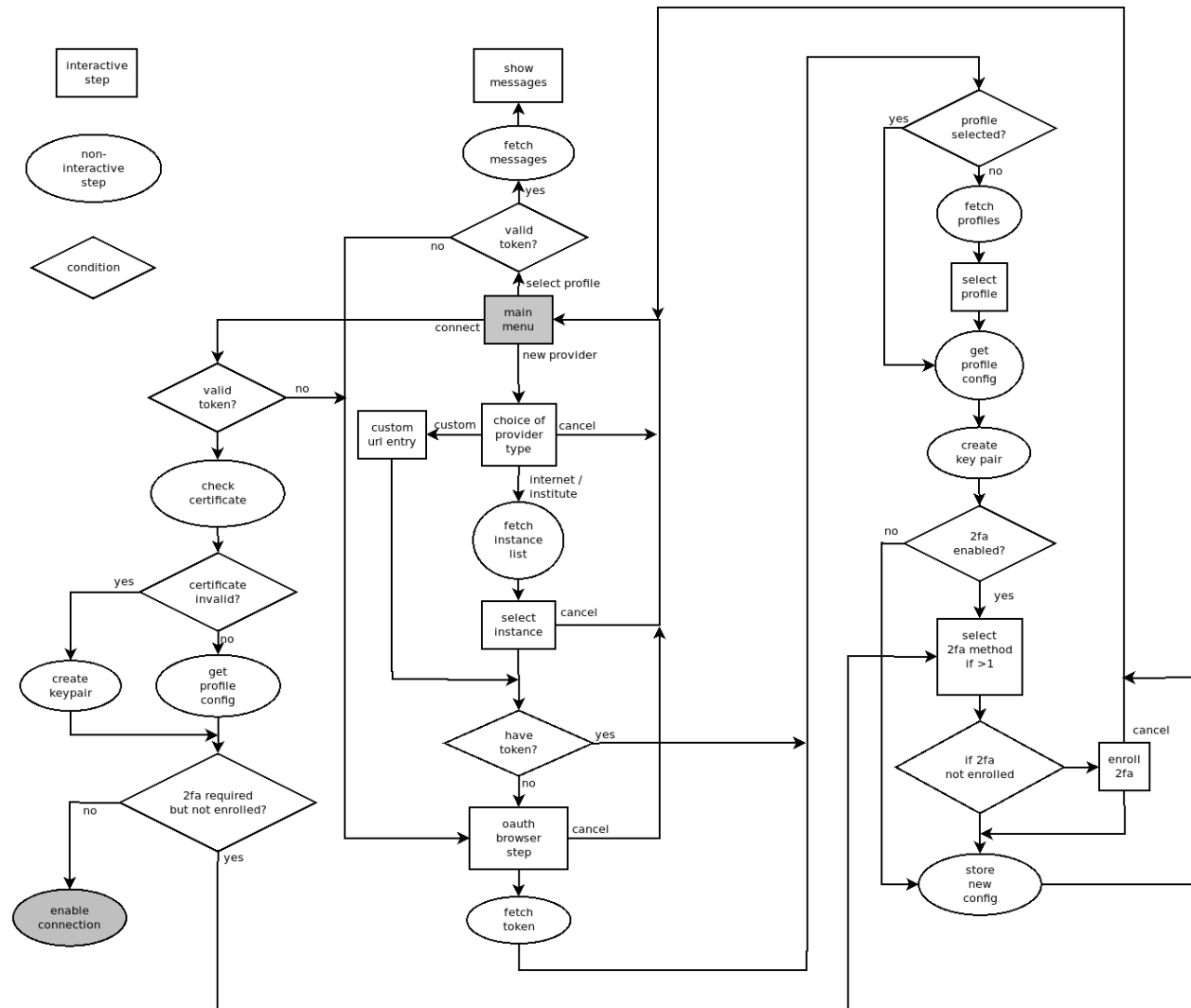
`eduvpn.actions` are the entrypoints to the application and are triggered from the main menu or a VPN status change.

`eduvpn.steps` contains all the various steps in the application flow.

`eduvpn.remote` contains all remote requests.

`eduvpn.other_nm` is a fork of the python NetworkManager wrapper.

## 3.2 Flow schema



## 3.3 How to make a release

- Determine version number (for example 1.0.2)
- Compose a list of changes (check issue tracker)
- Make sure the test suite runs with python2 and python3
- Set version number in `setup.py`, `setup_letsconnect.py` and `rpm/\*.spec`
- add changes to `CHANGES.md`
- Commit
- Press release button on github. List all changes here also
- Check if travis builds. If so, it will upload to pypi.
- If it doesn't build fix and do a manual upload using [twine](#)

- For now you need to manually create the Let's connect! wheel and upload using twine.
- Make a SRPM and upload to the [COPR repository](#)

---

**Note:** `$ make srpm` will use docker to build a srpm and put it in tmp/

---

- Update the debian package using the [eduVPN Debian meta files](#).





## CHAPTER 4

---

### Indices and tables

---

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



### **e**

- `eduvpn.crypto`, 7
- `eduvpn.exceptions`, 8
- `eduvpn.io`, 8
- `eduvpn.openvpn`, 8
- `eduvpn.remote`, 5



## C

`check_certificate()` (in module *eduvpn.remote*), 5  
`common_name_from_cert()` (in module *eduvpn.crypto*), 7  
`create_config()` (in module *eduvpn.remote*), 5  
`create_keypair()` (in module *eduvpn.remote*), 5

## E

`eduvpn.crypto` (module), 7  
`eduvpn.exceptions` (module), 8  
`eduvpn.io` (module), 8  
`eduvpn.openvpn` (module), 8  
`eduvpn.remote` (module), 5  
`EduvpnAuthException`, 8  
`EduvpnException`, 8

## F

`format_like_ovpn()` (in module *eduvpn.openvpn*), 8

## G

`gen_base32()` (in module *eduvpn.crypto*), 7  
`gen_code_challenge()` (in module *eduvpn.crypto*), 7  
`gen_code_verifier()` (in module *eduvpn.crypto*), 7  
`get_auth_url()` (in module *eduvpn.remote*), 5  
`get_instance_info()` (in module *eduvpn.remote*), 6  
`get_instances()` (in module *eduvpn.remote*), 6  
`get_profile_config()` (in module *eduvpn.remote*), 6

## L

`list_profiles()` (in module *eduvpn.remote*), 6

## M

`make_verifier()` (in module *eduvpn.crypto*), 7

`mkdir_p()` (in module *eduvpn.io*), 8

## O

`ovpn_to_nm()` (in module *eduvpn.openvpn*), 8

## P

`parse_ovpn()` (in module *eduvpn.openvpn*), 8

## S

`system_messages()` (in module *eduvpn.remote*), 6

## T

`translate_display_name()` (in module *eduvpn.remote*), 6  
`two_factor_enroll_totp()` (in module *eduvpn.remote*), 6  
`two_factor_enroll_yubi()` (in module *eduvpn.remote*), 6

## U

`user_info()` (in module *eduvpn.remote*), 7  
`user_messages()` (in module *eduvpn.remote*), 7

## W

`write_cert()` (in module *eduvpn.io*), 8