

---

# Identix One Python Documentation

*Release 0.1.6.3*

**Identix One**

**Mar 14, 2022**



## CONTENTS:

<b>1</b>	<b>identixone-python</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Getting Started . . . . .	2
1.3	Credits . . . . .	3
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Stable release . . . . .	5
2.2	From sources . . . . .	5
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>identixone</b>	<b>9</b>
4.1	identixone package . . . . .	9
<b>5</b>	<b>Contributing</b>	<b>13</b>
5.1	Types of Contributions . . . . .	13
5.2	Get Started! . . . . .	14
5.3	Pull Request Guidelines . . . . .	14
5.4	Tips . . . . .	15
5.5	Deploying . . . . .	15
<b>6</b>	<b>Credits</b>	<b>17</b>
6.1	Development Lead . . . . .	17
6.2	Contributors . . . . .	17
<b>7</b>	<b>History</b>	<b>19</b>
7.1	0.1.6.3 (2019-09-10) . . . . .	19
7.2	0.1.5 (2019-07-01) . . . . .	19
7.3	0.1.4 (2019-06-20) . . . . .	19
7.4	0.1.3 (2019-04-05) . . . . .	19
7.5	0.1.2 (2019-04-01) . . . . .	19
7.6	0.1.1 (2019-03-16) . . . . .	20
7.7	0.1.0 (2019-02-18) . . . . .	20
<b>8</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>
	<b>Index</b>	<b>25</b>



## IDENTIXONE-PYTHON

build passing

A Python package for interacting with the Identix.one API

- Free software: MIT license
- Package documentation: <https://identixone-python.readthedocs.io/>
- API documentation: <https://kb.identix.one/>
- API changelog: <https://kb.identix.one/#/apichangelog>
- Current supported most recent API version: **1.14.2**
- Current stable package version: **0.1.6.3**

### 1.1 Installation

Install from PyPi using `pip`, a package manager for Python.

```
pip install identixone
```

Don't have `pip` installed? Try installing it, by running this from the command line:

```
$ curl https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py | python
```

Or, you can [download the source code \(ZIP\)](#) for `identixone-python`, and then run:

```
python setup.py install
```

You may need to run the above commands with `sudo`.

#### 1.1.1 API Credentials

Get your free API token for development at <https://identix.one>

## 1.2 Getting Started

First of all, specify your API token and API version in *Client*:

```
from identixone.api import Client

version = 1
token = 'XXX'
client = Client(token, version)
```

You can also configure *Client* using environment variables with prefix *IDENTIXONE\_* and uppercase key (e.g. *TOKEN*, *VERSION*):

```
from identixone.api import Client

os.environ['IDENTIXONE_TOKEN'] = 'XXX'
os.environ['IDENTIXONE_VERSION'] = '1'
client = Client()
```

Now just make calls using *client* instance as if you were interacting with HTTP API.

For example, create source:

```
response = client.sources.create(name='source_name')
response.json()
# {"id": 1, "name": "source_name", "pps_timestamp": False, ... }
```

Or list some entries with filters:

```
import datetime

date_from = datetime.datetime(year=2019, month=1, day=13, hour=19,
                              minute=20, second=1)
date_to = datetime.datetime(year=2019, month=1, day=22, hour=19,
                             minute=20, second=1)
r = client.entries.list(date_from=date_from, date_to=date_to)
print(r.json())
# {"count": 1, "next": "url", "previous": "url", "results": [{ ... }]}
```

Or even compare two faces how similar they are:

```
from identixone.base.choices import Conf

response = client.utility.compare(
    photo1, photo2,
    liveness_photo1=False, liveness_photo2=False,
    conf=Conf.JUNK)
response.json()
# {"similar": True, "conf": "ha", "liveness_photo1": False, "liveness_photo2": True}
```

Full examples are inside *examples.py* file in the root of this repo.

To explore all of the API endpoints visit <https://kb.identix.one/>

## 1.3 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.





## INSTALLATION

### 2.1 Stable release

To install Identix One Python, run this command in your terminal:

```
$ pip install identixone
```

This is the preferred method to install Identix One Python, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

### 2.2 From sources

The sources for Identix One Python can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/identixone/identixone
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/identixone/identixone/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```



**USAGE**

Get your free API token for development at <https://identix.one>

To use Identix One Python in a project create instance of *Client*:

```
from identixone.api import Client

version = 1
token = 'XXX'
client = Client(token, version)
```

You can also configure *Client* using environment variables with prefix *IDENTIXONE\_* and uppercase key (e.g. TOKEN, VERSION):

```
from identixone.api import Client

os.environ['IDENTIXONE_TOKEN'] = 'XXX'
os.environ['IDENTIXONE_VERSION'] = '1'
client = Client()
```

Now just make calls using *client* instance as if you were interacting with HTTP API.

For example, create source:

```
response = client.sources.create(name='source_name')
response.json()
# {"id": 1, "name": "source_name", "pps_timestamp": False, ... }
```

Or list some entries with filters:

```
import datetime

date_from = datetime.datetime(year=2019, month=1, day=13, hour=19,
                              minute=20, second=1)
date_to = datetime.datetime(year=2019, month=1, day=22, hour=19,
                            minute=20, second=1)
r = client.entries.list(date_from=date_from, date_to=date_to)
print(r.json())
# {"count": 1, "next": "url", "previous": "url", "results": [{ ... }]}
```

Or even compare two faces how similar they are:

```
from identixone.base.choices import Conf

response = client.utility.compare(
```

(continues on next page)

(continued from previous page)

```
photo1, photo2,  
    liveness_photo1=False, liveness_photo2=False,  
    conf=Conf.JUNK)  
response.json()  
# {"similar": True, "conf": "ha", "liveness_photo1": False, "liveness_photo2": True}
```

Full examples are inside *examples.py* file in the root of this repo.

To explore all of the API endpoints visit <https://kb.identix.one/>

## IDENTIXONE

### 4.1 identixone package

#### 4.1.1 Subpackages

identixone.api package

Subpackages

identixone.api.users package

Subpackages

identixone.api.users.v1 package

Submodules

identixone.api.users.v1.users module

```
class identixone.api.users.v1.users.Users (http_client)
    Bases: object
    bulk_delete (permanent=None)
    change_password (old_password, password, password2, reset_tokens=None)
    delete_token (id_or_token)
    get_token (id_or_token)
    list_tokens (permanent=None)
    me ()
    update_token (id_or_token, **kwargs)
```

Module contents

Module contents

## Submodules

### identixone.api.client module

**class** `identixone.api.client.Client` (*token=None, version=None, http\_client=None, environment=None*)

Bases: `object`

High-level client to interact with the Identix.one API.

**property** `auth`

**dynamic\_import** (*module\_path, attribute*)

Imports any attribute from the module specified as string dotted path. Takes into account current supplied version to the Client instance.

**Parameters**

- **module\_path** – dotted path of the module from which to import from
- **attribute** – function, class or any other attr to be imported

**Returns** imported attribute

**property** `entries`

**env\_var** (*name*)

Wrapper around `env_var` utility function with supplied current environ.

**Parameters** **name** – name of the environment variable

**Returns** value of the environment variable or `None` if it doesn't exist

**property** `notifications`

**property** `persons`

**property** `persons_lists`

**property** `sources`

**property** `users`

**property** `utility`

## Module contents

### identixone.base package

#### Submodules

#### identixone.base.exceptions module

**exception** `identixone.base.exceptions.IdentixOneException`

Bases: `Exception`

Base for all errors produced by this library

**exception** `identixone.base.exceptions.ImproperlyConfigured`

Bases: `identixone.base.exceptions.IdentixOneException`

Errors that occur during configuration of Client

## Module contents

### identixone.http package

#### Submodules

#### identixone.http.client module

```
class identixone.http.client.IdentixOneHttpClient (auth_token, request_hooks=None,  
                                                timeout=None)
```

Bases: object

Http client used to make actual HTTP requests to Identix.one API. You can subclass it and provide new class as http\_client param into Client.

**property common\_headers**

**delete** (*url, params=None, data=None, files=None, headers=None, auth=None*)

**get** (*url, params=None, data=None, files=None, headers=None, auth=None*)

**patch** (*url, params=None, data=None, files=None, headers=None, auth=None*)

**post** (*url, params=None, data=None, files=None, headers=None, auth=None*)

**request** (*method, url, params=None, data=None, files=None, headers=None, auth=None*)

## Module contents

### identixone.utils package

#### Submodules

#### identixone.utils.environment module

```
identixone.utils.environment.env_var (environment, name)
```

## Module contents

### 4.1.2 Module contents

Top-level package for Identix One Python.





## CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

### 5.1 Types of Contributions

#### 5.1.1 Report Bugs

Report bugs at <https://github.com/identixone/identixone-python/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

#### 5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

#### 5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

#### 5.1.4 Write Documentation

Identix One Python could always use more documentation, whether as part of the official Identix One Python docs, in docstrings, or even on the web in blog posts, articles, and such.

#### 5.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/identixone/identixone-python/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 5.2 Get Started!

Ready to contribute? Here's how to set up *identixone* for local development.

1. Fork the *identixone* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/identixone-python.git
```

3. Install your local copy into a virtualenv. Assuming you have *virtualenvwrapper* installed, this is how you set up your fork for local development:

```
$ mkvirtualenv identixone
$ cd identixone/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass *flake8* and the tests, including testing other Python versions with *tox*:

```
$ flake8 identixone tests
$ python setup.py test or py.test
$ tox
```

To get *flake8* and *tox*, just *pip* install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in *README.rst*.
3. The pull request should work for Python 3.5 and 3.6. Check [https://travis-ci.org/identixone/identixone-python/pull\\_requests](https://travis-ci.org/identixone/identixone-python/pull_requests) and make sure that the tests pass for all supported Python versions.

## 5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_identixone
```

## 5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.



## CREDITS

### 6.1 Development Lead

- Lev Rubel <l@datacorp.ee>

### 6.2 Contributors

- Yaroslav Ulanovich <yul@datacorp.ee>
- Gleb Abroskin <gab@datacorp.ee>



## HISTORY

### 7.1 0.1.6.3 (2019-09-10)

- Added persons lists module

### 7.2 0.1.5 (2019-07-01)

- Returned entries module
- Removed records module

### 7.3 0.1.4 (2019-06-20)

- Removed Python 3.4 support
- Library updated to the new (1.12) API version
- Removed entries module
- Fixed places where default values were different than default API values.
- Fixed several API calls where default values should not be provided.

### 7.4 0.1.3 (2019-04-05)

- Fixed several API calls where default values should not be provided.
- Fixed places where default values were different than default API values.

### 7.5 0.1.2 (2019-04-01)

- API Changelog is now constantly updated here: <https://kb.identix.one/#/apichangelog>
- Updated documentation to show how to configure client with env variables
- Records endpoints are now deprecated
- Added new Entries and Entries Stats endpoints for RESTful manipulation with data (meant to replace and enhance records functionality)

- Added new Person Entries endpoint: create new person by providing id of NM entry
- Added examples of newly added endpoints

## 7.6 0.1.1 (2019-03-16)

- Updated docstrings for main functions
- New type of exception `ImproperlyConfigured` that replaces more general error in several places
- Added conf choices where applicable
- Added missing methods to bulk delete tokens with filtration (permanent/temporary/both)
- Added new source option `store_images_for_confs`, introduced in 1.9.0 API
- Added choices `NotificationHTTPMethod` of notifications `http_method` parameter for convenience
- Utility compare function now has default conf which equals HA. It reflects now default API behaviour
- Removed CHANGES.md because it is redundant. All changes are going to be reflected here, there's no need to duplicate info.
- Fixed a bug with env variables (fixed one typo & inability to override vars by setting env variables instead of providing them as parameters to `init` of `Client`)
- Fixed: previously you could provide your own `http_client` to the `Client` instance, but it required instance with already supplied token (so you basically needed to provide token in two places). Now you provide only class in `http_client` and initialization in `Client` will create instance with provided token for you.

## 7.7 0.1.0 (2019-02-18)

- First release on PyPI.



## INDICES AND TABLES

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



## PYTHON MODULE INDEX

### i

- `identixone`, [11](#)
- `identixone.api`, [10](#)
- `identixone.api.client`, [10](#)
- `identixone.api.users`, [9](#)
- `identixone.api.users.v1`, [9](#)
- `identixone.api.users.v1.users`, [9](#)
- `identixone.base`, [11](#)
- `identixone.base.exceptions`, [10](#)
- `identixone.http`, [11](#)
- `identixone.http.client`, [11](#)
- `identixone.utils`, [11](#)
- `identixone.utils.environment`, [11](#)



## INDEX

### A

`auth()` (*identixone.api.client.Client* property), 10

### B

`bulk_delete()` (*identixone.api.users.v1.users.Users* method), 9

### C

`change_password()` (*identixone.api.users.v1.users.Users* method), 9

`Client` (class in *identixone.api.client*), 10

`common_headers()` (*identixone.http.client.IdentixOneHttpClient* property), 11

### D

`delete()` (*identixone.http.client.IdentixOneHttpClient* method), 11

`delete_token()` (*identixone.api.users.v1.users.Users* method), 9

`dynamic_import()` (*identixone.api.client.Client* method), 10

### E

`entries()` (*identixone.api.client.Client* property), 10

`env_var()` (*identixone.api.client.Client* method), 10

`env_var()` (in module *identixone.utils.environment*), 11

### G

`get()` (*identixone.http.client.IdentixOneHttpClient* method), 11

`get_token()` (*identixone.api.users.v1.users.Users* method), 9

### I

*identixone* (module), 11

*identixone.api* (module), 10

*identixone.api.client* (module), 10

*identixone.api.users* (module), 9

*identixone.api.users.v1* (module), 9

*identixone.api.users.v1.users* (module), 9

*identixone.base* (module), 11

*identixone.base.exceptions* (module), 10

*identixone.http* (module), 11

*identixone.http.client* (module), 11

*identixone.utils* (module), 11

*identixone.utils.environment* (module), 11

*IdentixOneException*, 10

*IdentixOneHttpClient* (class in *identixone.http.client*), 11

*ImproperlyConfigured*, 10

### L

`list_tokens()` (*identixone.api.users.v1.users.Users* method), 9

### M

`me()` (*identixone.api.users.v1.users.Users* method), 9

### N

`notifications()` (*identixone.api.client.Client* property), 10

### P

`patch()` (*identixone.http.client.IdentixOneHttpClient* method), 11

`persons()` (*identixone.api.client.Client* property), 10

`persons_lists()` (*identixone.api.client.Client* property), 10

`post()` (*identixone.http.client.IdentixOneHttpClient* method), 11

### R

`request()` (*identixone.http.client.IdentixOneHttpClient* method), 11

### S

`sources()` (*identixone.api.client.Client* property), 10

## U

`update_token()` (*identixone.api.users.v1.users.Users* *method*), [9](#)

`Users` (*class in identixone.api.users.v1.users*), [9](#)

`users()` (*identixone.api.client.Client* *property*), [10](#)

`utility()` (*identixone.api.client.Client* *property*), [10](#)