# totppy Documentation

***Release 0.1.1***

**Abdullah Selek**

**Nov 18, 2018**

# Contents

Contents:

**A Python library for generating TOTP and HOTP one-time passwords.**

Author: Abdullah Selek

CHAPTER 1

# Installation & Testing

## 1.1 Installation

**From PyPI**

```
$ pip install totppy
```

**From source**

Install dependencies using *pip*:

```
$ pip install -r requirements.txt
```

Download the latest *totppy* library from: https://github.com/abdullahselek/totppy

Extract the source distribution and run:

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

## 1.2 Running Tests

The test suite can be run against a single Python version which requires `pip install pytest` and optionally `pip install pytest-cov` (these are included if you have installed dependencies from `requirements.testing.txt`)

To run the unit tests with a single Python version:

```
$ py.test -v
```

to also run code coverage:

```
$ py.test -v --cov-report html --cov=totppy
```

To run the unit tests against a set of Python versions:

```
$ tox
```

## 1.3 Getting the code

The code is hosted at Github.

Check out the latest development version anonymously with:

```
$ git clone https://github.com/abdullahselek/totppy.git
$ cd totppy
```

CHAPTER 2

# Modules Documentation

## 2.1 totppy

**class** totppy.generator.**Factor**(*digits=6*, *period=30*)
Bases: object

The possible values are *digits* and *period*, with associated values for each.

**class** totppy.generator.**Generator**(*factor*, *algorithm*, *secret*)
Bases: object

**password**(*date*)
Returns a onetime use password. Args:

**date (datetime):** The target time, represented as a *datetime*.

**Returns:** The generated password.

**class** totppy.enums.**Algorithm**
Bases: enum.Enum

**SHA1 = 1**

**SHA256 = 2**

**SHA512 = 3**

**class** totppy.enums.**Error**
Bases: enum.Enum

**INVALID_DIGITS = 3**

**INVALID_PERIOD = 2**

**INVALID_TIME = 1**

totppy.crypto.**HMAC**(*algorithm*, *key*, *data*)
Return a new hmac object. Args:

**algorithm (enum):** Enum type from *Algorithm*.

**key (str):** Key for the keyed hash object.

**data (bytes):** Initial input for the hash, if provided.

totppy.base32helper.**decode**(*bytes_to_decode*)

Decodes given bytes with base32. Args:

**bytes_to_decode (bytes):** bytes to be decoded.

**Returns:** Base32 decoded str.

totppy.base32helper.**encode**(*str_to_encode*)

Encodes given str with base32. Args:

**str_to_encode (str):** str to be encoded.

**Returns:** Base32 encoded bytes.

CHAPTER 3

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

## t

# Index