

---

# **pbwrap Documentation**

***Release 1.0.6***

**Michael Tsoukatos**

**Jun 07, 2021**



---

## Contents:

---

<b>1 Quickstart</b>	<b>3</b>
<b>Python Module Index</b>	<b>9</b>
<b>Index</b>	<b>11</b>



This wrapper is based on **Pastebin API** read their Documentation [here](#) for extra information and usage guide.



# CHAPTER 1

---

## Quickstart

---

Use **pip** to install **pbwrap**

```
pip install pbwrap
```

Import **pbwrap** to you script and create a *Pastebin* object.

```
from pbwrap import Pastebin  
  
pb = Pastebin(your_api_key)
```

You are now ready to request things from the Pastebin API

---

**Note:** It's recommended to authenticate as most endpoints require a user\_key API is limited without authenticating.

---

```
pb.authenticate(username, password)
```

---

**Note:** More documentation will be added in later releases! Read the Pastebin Class methods to learn how to use all the API endpoints.

---

## 1.1 Installation and Usage

Use **pip** to install **pbwrap**

```
pip install pbwrap
```

Import **pbwrap** to you script and create a *Pastebin* object.

```
from pbwrap import Pastebin  
  
pb = Pastebin(your_api_key)
```

You are now ready to request things from the Pastebin API

---

**Note:** It's recommended to authenticate as most endpoints require a user\_key API is limited without authenticating.

---

```
pb.authenticate(username, password)
```

## 1.2 Pastebin Class

**class** pbwrap.**Pastebin** (*api\_dev\_key=None*)

Pastebin class represents your communication with the Pastebin API through its functions you can use every API endpoint available.

Most functions require at least an `api_dev_key` parameter. Functions for manipulating your pastes through the API require an `api_user_key`.

Instantiate a Pastebin Object

**Parameters** `api_dev_key` (*string*) – Your API Pastebin key

**authenticate** (*username, password*)

Authenticate through the API login endpoint Your `api_user_key` attribute is set automatically

**Parameters**

- **username** (*string*) – Your username
- **password** (*string*) – Your password

**Returns** your `user_id` key

**Return type** `string`

**create\_paste** (*api\_paste\_code*, *api\_paste\_private=0*, *api\_paste\_name=None*,  
*api\_paste\_expire\_date=None*, *api\_paste\_format=None*)

Create a new paste if successful return its url.

**Parameters**

- **api\_paste\_code** (*string*) – your paste text
- **api\_paste\_private** (*int*) – valid values=0(public),1(unlisted),2(private)
- **api\_user\_name** – your paste name
- **api\_paste\_expire\_date** (*string*) – check documentation for valid values
- **api\_paste\_format** (*string*) – check documentation for valid values

**Returns** new paste url

**Return type** `string`

**create\_paste\_from\_file** (*filepath*, *api\_paste\_private=0*, *api\_paste\_name=None*,  
*api\_paste\_expire\_date=None*, *api\_paste\_format=None*)

Create a new paste from file if successful return its url.

**Parameters**



- **filepath** (*string*) – the path of the file
- **api\_paste\_private** (*int*) – valid values=0(public),1(unlisted),2(private)
- **api\_user\_name** – your paste name
- **api\_paste\_expire\_date** (*string*) – check documentation for valid values
- **api\_paste\_format** (*string*) – check documentation for valid values

**Returns** new paste url

**Return type** string

**delete\_user\_paste** (*api\_paste\_key*)

Deletes a paste created by the user.

**Parameters** **api\_paste\_key** (*string*) – the id key of the paste you want to delete

**Returns** api response

**Return type** string

**static get\_archive** ()

Return archive paste link list. Archive contains 25 most recent pastes.

**Returns** a list of url strings

**Return type** list

**static get\_raw\_paste** (*paste\_id*)

Return raw string of given paste\_id.

get\_raw\_paste(pasted\_id)

**Parameters** **paste\_id** (*string*) – The ID key of the paste

**Returns** the text of the paste

**Return type** string

**static get\_recent\_pastes** (*limit=50, lang=None*)

Return a list containing dictionaries of paste.

**Parameters**

- **limit** (*int*) – the limit of the items returned defaults to 50
- **lang** (*string*) – return only pastes from certain language defaults to None

**Returns** list of Paste objects.

**Return type** list(*Paste*)

**get\_trending** ()

Return a list of paste objects created from the most trending pastes

**Returns** a list of Paste objects

**Return type** list

**get\_user\_details** ()

Return user details in a dictionary. Can only be user after authenticating with get\_user\_id(username, password).

**Returns** dictionary containing user details

**Return type** dictionary

**get\_user\_pastes** (*api\_results\_limit=None*)

Return a list of Pastes created from the user

**Parameters** **api\_results\_limit** (*int*) – min=1, max=1000

**Returns** a list of Pastes created from the user

**Return type** list

**get\_user\_raw\_paste** (*api\_paste\_key*)

Return the raw data of a user paste(even private pastes!) as string.

**Parameters** **api\_paste\_key** (*string*) – the id key of the paste you want to fetch

**Returns** the text of the paste

**Return type** string

**static scrape\_paste\_metadata** (*paste\_key*)

Return a dictionary containing the metadata of the paste.

**Parameters** **paste\_key** (*string*) – the unique key of the paste you want to scrape

**Returns** dictionary containing the metadata of the paste

**Return type** dictionary

**static scrape\_raw\_paste** (*paste\_key*)

Return a string containing the text of the paste.

**Parameters** **paste\_key** (*string*) – the unique key of the paste you want to scrape

**Returns** raw string containing the text of the paste

**Return type** string

## 1.3 Formatter Module

Contains function that manipulate the API returns for easier data manipulation.

**pbwrap.formatter.archive\_url\_format** (*archive\_html*)

Return a list with recent pastes ids

**Parameters** **archive\_html** (*string*) – raw html of the archive url

**Returns** a list containing paste ids

**Return type** list

**pbwrap.formatter.paste\_list\_from\_xml** (*xml\_paste*)

Input an xml list and return a list of paste objects.

**Parameters** **xml\_paste** – An xml formatted response containing Paste information

**Type** string

**Returns** a list of Paste objects parsed from the input xml formatted string

**Return type** list(*Paste*)

**pbwrap.formatter.user\_from\_xml** (*user\_xml\_string*)

Return user dictionary from an xml format string

**Parameters** **user\_xml\_string** (*string*) – xml formatted string containing user information

**Returns** A dictionary containing the user info

Return type *User*

## 1.4 Models Classes

**class** pbwrap.models.**Paste** (*paste\_dict*)

Defines a Paste from Pastebin paste contains the following fields: key, date, title, size, expire\_date, private, format\_short, format\_long, url, hits.

**get\_raw\_text** ()

Fetch the text of a paste via the public API. :returns: the paste's text :rtype: string, None

**scrape\_raw\_text** ()

Fetch the ext of a paste via the Paid API. :returns: the paste's text :rtype: string, None

**class** pbwrap.models.**User** (*user\_dict*)

Defines a user contains the following fields: name format\_short expiration avatar\_url private website email location account\_type



### p

`pbwrap.formatter`, 6



## A

`archive_url_format()` (in module `pbwrap.formatter`), 6

`authenticate()` (*pbwrap.Pastebin method*), 4

## C

`create_paste()` (*pbwrap.Pastebin method*), 4

`create_paste_from_file()` (*pbwrap.Pastebin method*), 4

## D

`delete_user_paste()` (*pbwrap.Pastebin method*), 5

## G

`get_archive()` (*pbwrap.Pastebin static method*), 5

`get_raw_paste()` (*pbwrap.Pastebin static method*), 5

`get_raw_text()` (*pbwrap.models.Paste method*), 7

`get_recent_pastes()` (*pbwrap.Pastebin static method*), 5

`get_trending()` (*pbwrap.Pastebin method*), 5

`get_user_details()` (*pbwrap.Pastebin method*), 5

`get_user_pastes()` (*pbwrap.Pastebin method*), 5

`get_user_raw_paste()` (*pbwrap.Pastebin method*), 6

## P

`Paste` (*class in pbwrap.models*), 7

`paste_list_from_xml()` (in module `pbwrap.formatter`), 6

`Pastebin` (*class in pbwrap*), 4

`pbwrap.formatter` (*module*), 6

## S

`scrape_paste_metadata()` (*pbwrap.Pastebin static method*), 6

`scrape_raw_paste()` (*pbwrap.Pastebin static method*), 6

`scrape_raw_text()` (*pbwrap.models.Paste method*), 7

## U

`User` (*class in pbwrap.models*), 7

`user_from_xml()` (in module `pbwrap.formatter`), 6