

---

# **Chump**

*Release 1.6.0*

**Dec 03, 2018**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Usage Examples . . . . .	3
1.2	Developer Interface . . . . .	4
1.3	History . . . . .	8
	<b>Python Module Index</b>	<b>11</b>



Chump is a fully featured API wrapper for Pushover:

```
>>> from chump import Application
>>> app = Application('vmXXhu6J04RCQPpaAIFUR6JOq6j11P1')
>>> app.is_authenticated
True
>>> user = app.get_user('KAGAw2ZMxDJVhW2HAUiSZEamwGebNa')
>>> user.is_authenticated, user.devices
(True, {'iPhone'})
>>> message = user.send_message("What's up, dog?")
>>> message.is_sent, message.id, str(message.sent_at)
(True, '7LjjD6bK8hgqdK6aJzZUblOPPH9cVpjZ', '2005-10-05 07:50:40+00:00')
```



Install chump just like everything else:

```
$ pip install chump
```

## 1.1 Usage Examples

Chump's meant to be easy to use. Sending a message is just as simple as in the example above, but there's more you can do.

### 1.1.1 Creating and sending a message yourself

If you'd like to send messages yourself, just swap out `send_message()` for `create_message()`:

```
>>> message = user.create_message("Happy birthday, chuck!")
>>> message.is_sent, message.id
(False, None)
>>> message.send()
True
>>> message.is_sent, message.id, str(message.sent_at)
(True, 'fZSrekCvxi2vnpVADWBNchAGr11Di4cZ', '1993-12-17 06:03:45+00:00')
```

### 1.1.2 Sending messages with additional parameters

Chump supports all the message parameters outlined in Pushover's [API Docs](#). Any of these parameters can be optionally supplied as `kwargs`:

```
>>> message = user.create_message(
...     title="No Crackers, Gromit!",
```

(continues on next page)

(continued from previous page)

```
...     message="<b>We've forgotten the crackers!</b>",
...     html=True,
...     sound='intermission'
... )
>>> (str(message), message.sound)
('(No Crackers, Gromit!) <b>We've forgotten the crackers!</b>', 'intermission')
```

And Chump will raise the appropriate exceptions if your kwargs violate the API restrictions:

```
>>> message = user.create_message(
...     "Gromit, we have a problem!"
...     sound='this is not a sound'
... )
ValueError: Bad sound: must be in ('alien', 'bike', 'bugle',
    'cashregister', 'classical', 'climb', 'cosmic', 'echo', 'falling',
    'gamelan', 'incoming', 'intermission', 'magic', 'mechanical', 'none',
    'persistent', 'pianobar', 'pushover', 'siren', 'spacealarm', 'tugboat',
    'updown'), was 'this is not a sound'
```

All parameters are exposed as attributes in the *Message*, so you can change them later.

### 1.1.3 Sending an emergency message

Pushover's emergency messages have a few additions over standard messages. They require dismissal from the user, and if not dismissed they'll keep popping up every `retry` seconds until `timeout` seconds from when they were sent. When the user acknowledges the message, `callback` will be pinged by Pushover's servers, but you can also check in on the message's status by calling `poll()`:

```
>>> message = user.send_message(
...     "Do something, Gromit!",
...     priority=chump.EMERGENCY
... )
>>> message.is_sent, message.id, message.is_acknowledged
(True, 'eChnqsE5nZyefIbTVMuS9cfDV77mMaN9', False)
>>> message.poll()
False
>>> str(message.acknowledged_at)
'1995-12-24 06:10:39+00:00'
```

`poll()` returns `True` whilst the message has not been acknowledged, so you can use it cleanly as a condition in while loops.

## 1.2 Developer Interface

### 1.2.1 Main Interface

**class** `chump.Application` (*token*)

The Pushover application in use.

**Parameters** `token` (*string*) – The application's API token.

**token** = `None`

A *string* of the application's API token.



**limit = None**

If a message has been sent, an `int` of the application's monthly message limit, otherwise `None`.

**remaining = None**

If a message has been sent, an `int` of the application's remaining message allotment, otherwise `None`.

**reset = None**

If a message has been sent, `datetime` of when the application's monthly message limit will reset, otherwise `None`.

**is\_authenticated**

A lazily loaded `bool` indicating whether the application is authenticated.

**sounds**

A lazily loaded `dict` of available notification sounds if authenticated, otherwise `None`.

**get\_user (token)**

Returns a `User` attached to the `Application` instance.

**Parameters** `token (string)` – User API token.

**Return type** A `User`.

**class chump.User (app, token)**

A Pushover user. The user is tied to a specific `Application`, which can be changed later by setting `app`.

**Parameters**

- **app** (`Application`) – The Pushover application to send messages with.
- **token** (`string`) – The user's API token.

**app = None**

The Pushover application to send messages with.

**token = None**

A `string` of the user's API token.

**is\_authenticated**

A lazily loaded `bool` indicating whether the user is authenticated.

**devices**

A lazily loaded a `set` of the user's devices if authenticated, otherwise `None`.

**create\_message (message, html=False, title=None, timestamp=None, url=None, url\_title=None, device=None, priority=0, callback=None, retry=30, expire=86400, sound=None)**

Creates a message to the User with `app`.

**Parameters**

- **message** (`string`) – Body for the message.
- **html** (`bool`) – Whether the message should be formatted as HTML. Defaults to `False`.
- **title** (`string`) – (optional) Title for the message. Defaults to `None`.
- **timestamp** (`datetime` or `int`) – (optional) Date and time to give the message. Defaults to the time the message was created.
- **url** (`string`) – (optional) URL to include in the message. Defaults to `None`.
- **device** (`string`) – (optional) device from `devices` to send to. Defaults to all of the user's devices.
- **priority** (`int`) – (optional) priority for the message. The constants `LOWEST`, `LOW`, `NORMAL`, `HIGH`, and `EMERGENCY` may be used for convenience. Defaults to `NORMAL`.

- **callback** (*string*) – (optional) If priority is *EMERGENCY*, the URL to ping when the message is acknowledged. Defaults to *None*.
- **retry** (*int*) – (optional) If priority is *EMERGENCY*, the number of seconds to wait between re-alerting the user. Must be greater than 30. Defaults to 30.
- **expire** (*int*) – (optional) If priority is *EMERGENCY*, the number of seconds to retry before giving up on alerting the user. Must be less than 86400. Defaults to 86400.
- **sound** (*string*) – (optional) The sound from `app.sounds` to play when the message is received. Defaults to the user's default sound.

**Returns** An unsent message.

**Return type** A *Message* or *EmergencyMessage*.

**send\_message** (*message*, *html=False*, *title=None*, *timestamp=None*, *url=None*, *url\_title=None*, *device=None*, *priority=0*, *callback=None*, *retry=30*, *expire=86400*, *sound=None*)  
Does the same as `create_message()`, but then sends the message with `app`.

**Returns** A sent message.

**Return type** A *Message* or *EmergencyMessage*.

## 1.2.2 Lower-Level Classes

**class** `chump.Message` (*user*, *message*, *html=False*, *title=None*, *timestamp=None*, *url=None*, *url\_title=None*, *device=None*, *priority=0*, *sound=None*)

A Pushover message. The message is tied to a specific *Application*, and *User*. All parameters are exposed as attributes on the message, for convenience.

**Parameters** **user** (*User*) – The user to send the message to.

All other arguments are the same as in `User.create_message()`.

**id** = *None*

A *string* of the id of the message if sent, otherwise *None*.

**is\_sent** = *None*

A *bool* indicating whether the message has been sent.

**sent\_at** = *None*

A *datetime* of when the message was sent, otherwise *None*.

**error** = *None*

An *APIError* if there was an error sending the message, otherwise *None*.

**send** ()

Sends the message. If called after the message has been sent, resends it.

**Returns** A *bool* indicating if the message was successfully sent.

**Return type** A *bool*.

**class** `chump.EmergencyMessage` (*user*, *message*, *html=False*, *title=None*, *timestamp=None*, *url=None*, *url\_title=None*, *device=None*, *sound=None*, *callback=None*, *retry=30*, *expire=86400*)

Bases: `chump.Message`

An emergency Pushover message, (that is, a message with the priority of *EMERGENCY*).

All arguments are the same as in `Message`, with the additions of `callback`, `retry`, and `timeout`, which are all, too, as defined in `User.create_message()`.

**receipt = None**

A *string* of the receipt returned by the endpoint, for polling.

**last\_polled\_at = None**

A *datetime* of when the message was last polled.

**last\_delivered\_at = None**

A *datetime* of when the message was last delivered.

**is\_acknowledged = None**

A *bool* indicating whether the message has been acknowledged.

**acknowledged\_at = None**

A *datetime* of when the message was acknowledged, otherwise *None*.

**acknowledged\_by = None**

A *User* of the first user to have acknowledged the notification, otherwise *None*.

**is\_expired = None**

A *bool* indicating whether the message has expired.

**expires\_at = None**

A *datetime* of when the message expires.

**is\_called\_back = None**

A *bool* indicating whether the message has been called back.

**called\_back\_at = None**

A *datetime* of when the message was called back, otherwise *None*.

**send()**

Sends the message. If called after the message has been sent, resends it.

**Returns** A *bool* indicating if the message was successfully sent.

**Return type** A *bool*.

**poll()**

Polls for the results of the sent message. If the message has not been sent, does so.

**Returns** A *bool* indicating if the message has not expired, called back nor been acknowledged, or *None* if the message has no receipt with which to poll.

**Return type** A *bool* or *None*.

**cancel()**

Cancels the request for acknowledgment of a sent message.

**Returns** A *bool* indicating if the message was successfully cancelled.

**Return type** A *bool*.

### 1.2.3 Exceptions

**exception** `chump.APIError(url, request, response, timestamp)`

Pushover errors eponysterically end up here.

**Parameters**

- **url** (*string*) – The URL of the original request.
- **request** (*dict*) – The original request payload.
- **response** (*dict*) – The *json* response from the endpoint.

- **timestamp** (*datetime*) – When this error was raised.

**url** = **None**

A *string* of the URL of the original request.

**request** = **None**

A *dict* of the original request payload.

**response** = **None**

A *dict* of the *json* response from the endpoint.

**timestamp** = **None**

A *datetime* of when this error was raised.

**id** = **None**

A *string* of the request's id.

**status** = **None**

An *int* of the status code.

**errors** = **None**

A *list* of human readable error messages as *strings*.

**bad\_inputs** = **None**

A *dict* of the request's original arguments that the endpoint didn't like as *strings* and why, also as *strings*.

**receipt** = **None**

A *string* of the message's receipt if it was an emergency message, otherwise *None*.

### 1.2.4 Constants

`chump.LOWEST` = `-2`

Message priority: No sound, no vibration, no banner.

`chump.LOW` = `-1`

Message priority: No sound, no vibration, banner.

`chump.NORMAL` = `0`

Message priority: Sound, vibration, and banner if outside of user's quiet hours.

`chump.HIGH` = `1`

Message priority: Sound, vibration, and banner regardless of user's quiet hours.

`chump.EMERGENCY` = `2`

Message priority: Sound, vibration, and banner regardless of user's quiet hours, and re-alerts until acknowledged.

## 1.3 History

### 1.3.1 1.6.0 (11/04/2018)

- Reduce requests required to instantiate applications and users.
- Add connection pooling to improve network performance.
- Lazy evaluation of app and user authentication.
- Lazy evaluation of `Application.sounds` and `User.devices`.

- Improve compatibility with Python 2.7 through 3.7.
- Bugfixes.

### **1.3.2 1.5.2 (10/15/2016)**

- Exception logging bugfixes.
- Dropped support for Python 2.6.
- General code formatting improvements.

### **1.3.3 1.5.1 (03/05/2015)**

- Regression fixes.

### **1.3.4 1.5.0 (03/05/2015)**

- Use unicode exclusively.
- Update length limits on message strings.
- Add support for HTML messages.
- Improve timezone support when pytz is unavailable.
- Remove requests dependency.
- Improve documentation.

### **1.3.5 1.4.0 (05/31/2014)**

- Add new Pushover priority: LOWEST.
- Add Application.{limit,remaining,reset} to track message allotment.
- Fix title and message length validation.

### **1.3.6 1.3.2 (05/16/2014)**

- Fix dependency issues when installing.

### **1.3.7 1.3.1 (09/30/2013)**

- Updated authentication checks for new API responses.

### **1.3.8 1.3.0 (09/01/2013)**

- Added Python 2.6 support.

**1.3.9 1.2.1 (09/01/2013)**

- Bugfixes.

**1.3.10 1.2.0 (08/30/2013)**

- Renamed classes.

**1.3.11 1.1.0 (08/30/2013)**

- Added Python 3 support.

**1.3.12 1.0.0 (08/30/2013)**

- API overhaul and “completion”.

**1.3.13 0.1.0 (08/28/2013)**

- Initial release.

**C**

chump, 8





**A**

acknowledged\_at (chump.EmergencyMessage attribute),  
7  
acknowledged\_by (chump.EmergencyMessage attribute),  
7  
APIError, 7  
app (chump.User attribute), 5  
Application (class in chump), 4

**B**

bad\_inputs (chump.APIError attribute), 8

**C**

called\_back\_at (chump.EmergencyMessage attribute), 7  
cancel() (chump.EmergencyMessage method), 7  
chump (module), 4, 8  
create\_message() (chump.User method), 5

**D**

devices (chump.User attribute), 5

**E**

EMERGENCY (in module chump), 8  
EmergencyMessage (class in chump), 6  
error (chump.Message attribute), 6  
errors (chump.APIError attribute), 8  
expires\_at (chump.EmergencyMessage attribute), 7

**G**

get\_user() (chump.Application method), 5

**H**

HIGH (in module chump), 8

**I**

id (chump.APIError attribute), 8  
id (chump.Message attribute), 6  
is\_acknowledged (chump.EmergencyMessage attribute),  
7

is\_authenticated (chump.Application attribute), 5  
is\_authenticated (chump.User attribute), 5  
is\_called\_back (chump.EmergencyMessage attribute), 7  
is\_expired (chump.EmergencyMessage attribute), 7  
is\_sent (chump.Message attribute), 6

**L**

last\_delivered\_at (chump.EmergencyMessage attribute),  
7  
last\_polled\_at (chump.EmergencyMessage attribute), 7  
limit (chump.Application attribute), 4  
LOW (in module chump), 8  
LOWEST (in module chump), 8

**M**

Message (class in chump), 6

**N**

NORMAL (in module chump), 8

**P**

poll() (chump.EmergencyMessage method), 7

**R**

receipt (chump.APIError attribute), 8  
receipt (chump.EmergencyMessage attribute), 6  
remaining (chump.Application attribute), 5  
request (chump.APIError attribute), 8  
reset (chump.Application attribute), 5  
response (chump.APIError attribute), 8

**S**

send() (chump.EmergencyMessage method), 7  
send() (chump.Message method), 6  
send\_message() (chump.User method), 6  
sent\_at (chump.Message attribute), 6  
sounds (chump.Application attribute), 5  
status (chump.APIError attribute), 8

## T

timestamp (chump.APIError attribute), 8  
token (chump.Application attribute), 4  
token (chump.User attribute), 5

## U

url (chump.APIError attribute), 8  
User (class in chump), 5