pyshk Documentation

Release 1.1.0

Jeremy Low

December 20, 2015

Contents

1 Warnings **3**

2 Installation **5**

3 Authentication Tutorial **7**
3.1 Introduction . . . 7
3.2 Getting Authentication Tokens . . . 7
3.3 Starting the API Client . . . 8

4 Using the API Client **9**
4.1 Introduction . . . 9
4.2 Examples . . . 9

5 Models **13**
5.1 User . . . 13
5.2 Shake . . . 13
5.3 SharedFile . . . 14

6 Modules **17**
6.1 pyshk package . . . 17

7 Indices and tables **23**

Python Module Index **25**

Contents:

CHAPTER 1

Warnings

Just some things to be aware of prior to using this library:

Danger: If there are any problems or you find a bug, please create an issue on GitHub.

Danger: mlkshk does not currently provide an intermediate certificate for SSL validation, so SSL verification has been disabled for the time being.

CHAPTER 2

Installation

Until distribution by PyPi is setup, please clone the repository:

```
git clone git://github.com/jeremylow/pyshk.git
cd pyshk
```

Danger: This is pre-alpha software, so things **might change and break often**. Buyer beware. If there are any problems or you find a bug, please create an issue on GitHub.

Danger: mlkshk does not currently provide an intermediate certificate for SSL validation, so SSL verification has been disabled for the time being.

CHAPTER 3

Authentication Tutorial

3.1 Introduction

Mlkshk uses draft 12 of the OAuth2 specification. After we create an instance of the API, we authenticate against mlkshk.com. Thereafter, the API instance is authenticated. In order to access the mlkshk API, you need to create an application.

You should do that http://mlkshk.com/developers/new-api-application. Give your application a title and a description. Make sure to fill in the "Redirect URL" cause we'll use that later.

Warning: The redirect url has to be exactly the same on mlkshk.com and in your code or the server will return a 500 server error when you try to authenticate.

Once you click the big "Create It!" button, you'll be given a "Key" and a "Secret". Write those down. We'll need them later in this tutorial.

3.2 Getting Authentication Tokens

If this is your first run and you don't have access tokens to use the API, you should start a python interpreter and import the `api` moduled.:

```
from pyshk.api import Api
```

Next, we will create the API instance and get the access tokens. Here we'll use the consumer key, secret, and redirect URI from above, so have those handy. (Also, your default web browser will open after you enter the second line – just fair warning.)

```
a = Api(consumer_key=[Key], consumer_secret=[Secret])
a.get_auth(redirect_uri=[the redirect url])
```

If everything went well, your browser opened and asked you to allow access to your account. Allow the app access and you'll be redirected to whatever redirect URL you specified when creating the app.

The URL bar will have something like `http://[redirect_url]?code=[code]`. Copy and paste the `[code]` portion back into the python interpreter and hit enter. You'll then be given the access code and the access secret; you should write those down so you don't have to go through this again.

3.3 Starting the API Client

At this point, the API is authenticated and you can start making calls to the mlkshk server. If you went through this before and you already have an acces code & secret, you can pass those to the API when you instantiate it. Itl'll be something like:

```
api = Api(consumer_key='blah',
          consumer_secret='blahsecretblah',
          access_token_key='blah',
          access_token_secret='blahsupersecretblah')
```

So if everything above worked, you should be able to interact with the API at this point by starting it as above.

Checkout the 'Using the API' documentation for examples.

CHAPTER 4

Using the API Client

4.1 Introduction

If you haven't, please go through the Authentication Tutorial. You'll need an authenticated API client for the next section.

4.2 Examples

Let's start by creating an API client to use:

```
>>> from pyshk.api import Api

>>> client = Api(
...     consumer_key='key',
...     consumer_secret='secret',
...     access_token_key='key',
...     access_token_secret='supersecret')
```

4.2.1 Users

To get a User object for the currently authenticated user, we can call `client.get_user()`

```
>>> user = client.get_user()
```

This will return a User instance containing the user's Shakes and some information about the user: The User object has the following properties:

```
* about
* id
* name
* profile_image_url
* shakes
* website
```

Unfortunately, you cannot post to this resource, so those properties are unable to be changed. But, we can play around with the user's Shakes, so let's do that.

4.2.2 Shakes

The `user.shakes` property from above is a list of Shake objects. If the user has a pro account, they can have more than one Shake; if not, the only Shake will be their User Shake. To get the user's Shakes directly, there's another method, which will return a list of Shakes:

```
>>> shakes = client.get_user_shakes()
```

Getting the first Shake in that list, we can print some info about it:

```
>>> shake = shakes[0]
>>> shake.id
68435

>>> print(shake.created_at)
2015-04-27 17:22:54

# The time fields are actually stored as python datetime objects:

>>> shake.created_at
datetime.datetime(2015, 4, 27, 17, 22, 54)

# The Shake object contains a User object as well stored as 'owner':

>>> shake.owner.id
67136
>>> shake.owner.name
'jcbl'
```

4.2.3 Uploading Files

Now that we have a Shake object, we can start posting files to it. The client method for this is `client.post_shared_file()`

```
>>> uploaded_file = client.post_shared_file(
        image_file='scully.gif',
        title='scully.gif',
        description="this is an excellent gif")
```

What we'll get back is the name of the ShareFile and the sharekey.

```
{'name': 'scully.gif', 'share_key': '1645C'}
```

> **Warning:** While most methods return an object, this one doesn't – it just returns some info.

From there, we can get some info on the uploaded file and maybe post a comment or two.

4.2.4 SharedFiles

SharedFiles on mlkshk have a bunch of properties; we won't go into detail, but you can check out the modules page to see what's up. So now that we have the sharekey from above, we can get some more info on our awesome gif.

```
>>> awesome_gif = client.get_shared_file(sharekey='1645C')

# Let's see how many people liked this GIF:
```

```
>>> awesome_gif.likes
1

# This is a crime, but moving on, let's just see what all awesome_gif contains:
>>> vars(awesome_gif)
{'_posted_at': datetime.datetime(2015, 10, 9, 20, 34, 34),
 'comments': 0,
 'description': None,
 'height': 180,
 'image_url': None,
 'liked': False,
 'likes': 1,
 'name': 'tumblr_mo6ur4bPUm1rxfs8ro5_250.gif',
 'nsfw': False,
 'permalink': None,
 'saved': False,
 'saves': 0,
 'sharekey': '1645C',
 'source_url': None,
 'title': 'scully.gif',
 'user': {
    "id": 67136,
    "mlkshk_url": "https://mlkshk.com/user/jcbl",
    "name": "jcbl",
    "profile_image_url": "[...]",
    "shake_count": 0},
 'views': 0,
 'width': 245}
```

Warning: A couple of things to note before moving on: `awesome_gif.views` is wrong. At the time of this writing, it's off by 142 views. Everything else about the SharedFile seems to be correct, except that now `user.shake_count` is `0`. This is a function of the fact that the endpoint (`/api/sharedfile/:id`) returns information about the user, but it doesn't return information about the user's shakes.

4.2.5 Comments, Saves, & Likes

TKTK

CHAPTER 5

Models

5.1 User

class User ([*id=None*][, *name=None*][, *profile_image_url=None*][, *about=None*][, *website=None*][, *shakes=None*])

Represents a User on mlkshk and has the properties listed below. User information cannot be changed via the API as there is no endpoint for this functionality.

Parameters

- **id** – (int) ID of the user for mlkshk; unable to be changed via mlkshk.com or the API.
- **name** – (str): Username; unable to be changed via mlkshk.com or the API.
- **profile_image_url** – (str): Profile picture of the user. User can change this on the website.
- **about** – (str): Short description of the user.
- **website** – (str): User's website (i.e., not the user's page on mlshk.com).
- **shakes** – (list): List of *Shake* objects.

NewFromJson ([*data=None*])

Create a User object from a JSON string. Returns a User object.

AsDict ([*dt=True*])

Parameters **dt** – (bool) Return dates as `datetime` objects; false to return as ISO strings. Return the User object as a dictionary of values. NB: Dates will be returned as native, UTC-timezone (but not timezone aware) `datetime.datetime` objects.

AsJsonString ()

Return the User object as JSON string. NB: any dates will be converted to ISO strings as part of the serialization.

5.2 Shake

class Shake ([*id=None*][, *name=None*][, *owner=None*][, *url=None*][, *thumbnail_url=None*][, *description=None*][, *type=None*][, *created_at=None*][, *updated_at=None*])

Object representing a Shake on mlkshk.com

Parameters

- **id** – (id): ID of the Shake
- **name** – (str): Shake's name
- **owner** – (User): User object representing the owner of the Shake
- **url** – (str): URL of the Shake (i.e., mlkshk.com/commute)
- **thumbnail_url** – (str): "Avatar" for the Shake
- **description** – (str): Description of the Shake
- **type** – (str): 'user' or 'group' denoting whether the Shake is has no contributors other than the owner (`user` shake) or many (`group` shake).
- **created_at** – (datetime): `datetime.datetime` representing the creation date & time of the Shake.
- **updated_at** – (datetime): Last modified `datetime.datetime` of the Shake.

NewFromJson([*data=None*])

Return a Shake object from a JSON string.

AsDict([*dt=True*])

Parameters **dt** – (bool) Return dates as `datetime` objects; false to return as ISO strings. Return the User object as a dictionary of values. NB: Dates will be returned as native, UTC-timezone (but not timezone aware) `datetime.datetime` objects.

AsJsonString()

Return a JSON string representing this Shake instance. NB: any dates will be converted to ISO strings as part of the serialization.

5.3 SharedFile

class SharedFile([*sharekey=None*][, *name=None*][, *user=None*][, *title=None*][, *description=None*][, *posted_at=None*][, *permalink=None*][, *width=None*][, *height=None*][, *views=None*][, *likes=None*][, *saves=None*][, *comments=None*][, *nsfw=None*][, *image_url=None*][, *source_url=None*][, *saved=None*][, *liked=None*])

Object representing a file or embedded video on mlkshk.com

Parameters

- **sharekey** – (str): Sharekey of the file.
- **name** – (str): Filename as uploaded to mlkshk.com
- **user** – (User): User object representing user who uploaded the file
- **title** – (str): Title of the SharedFile
- **description** – (str): Description of the SharedFile
- **posted_at** – (datetime): `datetime.datetime` object representing the time the SharedFile was uploaded
- **permalink** – (str): URL of the SharedFile. Example: "http://mlkshk.com/1645C"
- **width** – (int): Width of the image.
- **height** – (int): Height of the image
- **views** – (int): How many views the SharedFile has received.

- **likes** – (int): How many likes/favorites the SharedFile has received.
- **saves** – (int): How many saves the SharedFile has received.
- **comments** – (list): List of Comment objects posted to the SharedFile.
- **nsfw** – (bool): Whether the file is not safe for work (NB: folks in the community sometimes flag very large GIFs as NSFW).
- **image_url** – (str): URL for hotlinking to the image file.
- **source_url** – (str): If not an image, this can be a YouTube, Vine, etc. link.
- **saved** – (bool): If the currently authenticated user has saved the file.
- **liked** – (bool): If the currently authenticated user has liked the file.

NewFromJson([*data=None*])

Return a SharedFile object from a JSON string.

AsDict([*dt=True*])

Parameters **dt** – (bool) Return dates as `datetime` objects; false to return as ISO strings. Return the User object as a dictionary of values. NB: Dates will be returned as native, UTC-timezone (but not timezone aware) `datetime.datetime` objects.

Return a dictionary representing the SharedFile object.

AsJsonString()

Return a JSON string representing this SharedFile instance. NB: any dates will be converted to ISO strings as part of the serialization.

CHAPTER 6

Modules

6.1 pyshk package

6.1.1 Submodules

6.1.2 pyshk.api module

Library to provide access to MLKSHK API.

class `pyshk.api.Api`(*consumer_key=None*, *consumer_secret=None*, *access_token_key=None*, *access_token_secret=None*, *base_url=None*, *testing=False*)

Bases: `object`

get_auth(*redirect_uri=None*)

get_comments(*sharekey=None*)

Retrieve comments on a SharedFile

Parameters **sharekey** (*str*) – Sharekey for the file from which you want to return the set of comments.

Returns List of Comment objects.

get_favorites(*before=None*, *after=None*)

Get a list of the authenticated user's 10 most recent favorites (likes).

Parameters

- **before** (*str*) – get 10 SharedFile objects before (but not including) the SharedFile given by *before* for the authenticated user's set of Likes.
- **after** (*str*) – get 10 SharedFile objects after (but not including) the SharedFile give by 'after' for the authenticated user's set of Likes.

Returns List of SharedFile objects.

get_friends_shake(*before=None*, *after=None*)

Contrary to the endpoint naming, this resource is for a list of SharedFiles from your friends on mlkshk.

Returns List of SharedFiles.

get_incoming_shake(*before=None*, *after=None*)

Returns a list of the most recent SharedFiles on mlkshk.com

Parameters

- **before** (*str*) – get 10 SharedFile objects before (but not including) the SharedFile given by *before* for the Incoming Shake.
- **after** (*str*) – get 10 SharedFile objects after (but not including) the SharedFile give by 'after' for the Incoming Shake.

Returns List of SharedFile objects.

get_magic_shake (*before=None*, *after=None*)

From the API:

Returns the 10 most recent files accepted by the 'magic' file selection algorithm. Currently any files with 10 or more likes are magic.

Returns List of SharedFile objects

static get_nonce ()

get_shared_file (*sharekey=None*)

Returns a SharedFile object given by the sharekey.

Parameters **sharekey** (*str*) – Sharekey of the SharedFile you want to retrieve.

Returns SharedFile

get_shared_files_from_shake (*shake_id=None*, *before=None*, *after=None*)

Returns a list of SharedFile objects from a particular shake.

Parameters

- **shake_id** (*int*) – Shake from which to get a list of SharedFiles
- **before** (*str*) – get 10 SharedFile objects before (but not including) the SharedFile given by *before* for the given Shake.
- **after** (*str*) – get 10 SharedFile objects after (but not including) the SharedFile give by 'after' for the given Shake.

Returns List (list) of SharedFiles.

get_user (*user_id=None*, *user_name=None*)

Get a user object from the API. If no `user_id` or `user_name` is specified, it will return the User object for the currently authenticated user.

Parameters

- **user_id** (*int*) – User ID of the user for whom you want to get information. [Optional]
- **user_name** (*str*) – Username for the user for whom you want to get information. [Optional]

Returns A User object.

get_user_shakes ()

Get a list of Shake objects for the currently authenticated user.

Returns A list of Shake objects.

like_shared_file (*sharekey=None*)

'Like' a SharedFile. mlkshk doesn't allow you to unlike a sharedfile, so this is ~~permanent~~.

Parameters **sharekey** (*str*) – Sharekey for the file you want to 'like'.

Returns Either a SharedFile on success, or an exception on error.

post_comment (*sharekey=None*, *comment=None*)

Post a comment on behalf of the current user to the SharedFile with the given sharekey.

Parameters

- **sharekey** (*str*) – Sharekey of the SharedFile to which you'd like to post a comment.
- **comment** (*str*) – Text of the comment to post.

Returns Comment object.

post_shared_file(*image_file=None*, *source_link=None*, *shake_id=None*, *title=None*, *description=None*)

Upload an image.

TODO: Don't have a pro account to test (or even write) code to upload a shared filed to a particular shake.

Parameters

- **image_file** (*str*) – path to an image (jpg/gif) on your computer.
- **source_link** (*str*) – URL of a source (youtube/vine/etc.)
- **shake_id** (*int*) – shake to which to upload the file or source_link [optional]
- **title** (*str*) – title of the SharedFile [optional]
- **description** (*str*) – description of the SharedFile

Returns SharedFile key.

save_shared_file(*sharekey=None*)

Save a SharedFile to your Shake.

Parameters **sharekey** (*str*) – Sharekey for the file to save.

Returns SharedFile saved to your shake.

update_shared_file(*sharekey=None*, *title=None*, *description=None*)

Update the editable details (just the title and description) of a SharedFile.

Parameters

- **sharekey** (*str*) – Sharekey of the SharedFile to update.
- **title** (*Optional[str]*) – Title of the SharedFile.
- **description** (*Optional[str]*) – Description of the SharedFile

Returns SharedFile on success, 404 on Sharekey not found, 403 on unauthorized.

6.1.3 pyshk.errors module

exception pyshk.errors.**ApiInstanceUnauthorized**

Bases: exceptions.Exception

exception pyshk.errors.**ApiResponseUnauthorized**(*response*)

Bases: exceptions.Exception

Handle errors with authenticating the API instance

exception pyshk.errors.**NotFound404**(*response*)

Bases: exceptions.Exception

Handle errors with 404 Not Found Status Codes

6.1.4 pyshk.models module

class `pyshk.models.`**`Comment`**(***kwargs*)

Bases: `object`

A class representing a Comment on mlkshk.

Exposes the following properties of a Comment: comment.body comment.posted_at comment.user

`AsDict`(*dt=True*)

A dict representation of this Comment instance.

The return value uses the same key names as the JSON representation.

Parameters

- **`dt`** (*bool*) – If True, return dates as python datetime objects. If
- **`return dates as ISO strings.`** (*False,*) –

Returns A dict representing this Comment instance

`AsJsonString`()

A JSON string representation of this Comment instance.

Returns A JSON string representation of this Comment instance

static **`NewFromJSON`**(*data*)

Create a new Comment instance from a JSON dict.

Parameters **`data`** (*dict*) – JSON dictionary representing a Comment.

Returns A Comment instance.

`__repr__`()

String representation of this Comment instance.

class `pyshk.models.`**`Shake`**(***kwargs*)

Bases: `object`

A class representing a Shake on mlkshk.

Exposes the following properties of a Shake: shake.id shake.name shake.owner shake.url shake.thumbnail_url shake.description shake.type shake.created_at shake.updated_at

`AsDict`(*dt=True*)

A dict representation of this Shake instance.

The return value uses the same key names as the JSON representation.

Returns A dict representing this Shake instance

`AsJsonString`()

A JSON string representation of this Shake instance.

Returns A JSON string representation of this Shake instance

static **`NewFromJSON`**(*data*)

Create a new Shake instance from a JSON dict.

Parameters **`data`** (*dict*) – JSON dictionary representing a Shake.

Returns A Shake instance.

`__repr__`()

String representation of this Shake instance.

created_at

created_at_iso

updated_at

updated_at_iso

class pyshk.models.**SharedFile**(**args*, ***kwargs*)

Bases: object

A class representing a file shared on MLKSHK.

Exposes the following properties of a sharedfile: sharedfile.sharekey sharedfile.name sharedfile.user sharedfile.title sharedfile.description sharedfile.posted_at sharedfile.permalink sharedfile.width sharedfile.height sharedfile.views sharedfile.likes sharedfile.saves sharedfile.comments sharedfile.nsfw sharedfile.image_url sharedfile.source_url sharedfile.saved sharedfile.liked

Parameters

- **sharedfile.sharekey** –
- **sharedfile.name** –
- **sharedfile.user** –
- **sharedfile.title** –
- **sharedfile.description** –
- **sharedfile.posted_at** –
- **sharedfile.permalink** –
- **sharedfile.width** –
- **sharedfile.height** –
- **sharedfile.views** –
- **sharedfile.likes** –
- **sharedfile.saves** –
- **sharedfile.comments** –
- **sharedfile.nsfw** –
- **sharedfile.image_url** –
- **sharedfile.source_url** –
- **sharedfile.saved** –
- **sharedfile.liked** –

AsDict(*dt=True*)

A dict representation of this Shake instance.

The return value uses the same key names as the JSON representation.

Parameters **dt** (*bool*) – If True, return dates as python datetime objects. If False, return dates as ISO strings.

Returns A dict representing this Shake instance

AsJsonString()

A JSON string representation of this SharedFile instance.

Returns A JSON string representation of this SharedFile instance

static NewFromJSON(*data*)
Create a new SharedFile instance from a JSON dict.

Parameters **data** (*dict*) – JSON dictionary representing a SharedFile.

Returns A SharedFile instance.

__eq__(*other*)
Compare two SharedFiles on all attributes **except** saved status and liked status.

posted_at

posted_at_iso

class pyshk.models.**User**(**kwargs)
Bases: object

A class representing a MLKSHK user.

Exposes the following properties of a user: user.id user.name user.profile_image_url user.about user.website user.shakes

AsDict(*dt=True*)
A dict representation of this User instance.

The return value uses the same key names as the JSON representation.

Parameters

- **dt** (*bool*) – If True, return dates as python datetime objects. If
- **return dates as ISO strings.** (*False,*) –

Returns A dict representing this User instance

AsJsonString()
A JSON string representation of this User instance.

Returns A JSON string representation of this User instance

static NewFromJSON(*data*)
Create a new User instance from a JSON dict.

Parameters **data** (*dict*) – JSON dictionary representing a user.

Returns A User instance.

__eq__(*other*)
Compare two user objects against one another.

Parameters **other** (User) – another User object against which to compare the current user.

__repr__()
String representation of this User instance.

mlkshk_url

shake_count

pyshk.models.**convert_time**(*dt*)
2015-10-09T15:58:11Z -> datetime.datetime(2015, 10, 9, 15, 58, 11)

6.1.5 Module contents

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

p

pyshk, 22
pyshk.api, 17
pyshk.errors, 19
pyshk.models, 20

Symbols

__eq__() (pyshk.models.SharedFile method), 22
__eq__() (pyshk.models.User method), 22
__repr__() (pyshk.models.Comment method), 20
__repr__() (pyshk.models.Shake method), 20
__repr__() (pyshk.models.User method), 22

A

Api (class in pyshk.api), 17
ApiInstanceUnauthorized, 19
ApiResponseUnauthorized, 19
AsDict() (pyshk.models.Comment method), 20
AsDict() (pyshk.models.Shake method), 20
AsDict() (pyshk.models.SharedFile method), 21
AsDict() (pyshk.models.User method), 22
AsDict() (Shake method), 14
AsDict() (SharedFile method), 15
AsDict() (User method), 13
AsJsonString() (pyshk.models.Comment method), 20
AsJsonString() (pyshk.models.Shake method), 20
AsJsonString() (pyshk.models.SharedFile method), 21
AsJsonString() (pyshk.models.User method), 22
AsJsonString() (Shake method), 14
AsJsonString() (SharedFile method), 15
AsJsonString() (User method), 13

C

Comment (class in pyshk.models), 20
convert_time() (in module pyshk.models), 22
created_at (pyshk.models.Shake attribute), 20
created_at_iso (pyshk.models.Shake attribute), 21

G

get_auth() (pyshk.api.Api method), 17
get_comments() (pyshk.api.Api method), 17
get_favorites() (pyshk.api.Api method), 17
get_friends_shake() (pyshk.api.Api method), 17
get_incoming_shake() (pyshk.api.Api method), 17
get_magic_shake() (pyshk.api.Api method), 18
get_nonce() (pyshk.api.Api static method), 18
get_shared_file() (pyshk.api.Api method), 18
get_shared_files_from_shake() (pyshk.api.Api method), 18
get_user() (pyshk.api.Api method), 18
get_user_shakes() (pyshk.api.Api method), 18

L

like_shared_file() (pyshk.api.Api method), 18

M

mlkshk_url (pyshk.models.User attribute), 22

N

NewFromJSON() (pyshk.models.Comment static method), 20
NewFromJSON() (pyshk.models.Shake static method), 20
NewFromJSON() (pyshk.models.SharedFile static method), 22
NewFromJSON() (pyshk.models.User static method), 22
NewFromJson() (Shake method), 14
NewFromJson() (SharedFile method), 15
NewFromJson() (User method), 13
NotFound404, 19

P

post_comment() (pyshk.api.Api method), 18
post_shared_file() (pyshk.api.Api method), 19
posted_at (pyshk.models.SharedFile attribute), 22
posted_at_iso (pyshk.models.SharedFile attribute), 22
pyshk (module), 22
pyshk.api (module), 17
pyshk.errors (module), 19
pyshk.models (module), 20

S

save_shared_file() (pyshk.api.Api method), 19
Shake (built-in class), 13
Shake (class in pyshk.models), 20
shake_count (pyshk.models.User attribute), 22

SharedFile (built-in class), 14
SharedFile (class in pyshk.models), 21

U

update_shared_file() (pyshk.api.Api method), 19
updated_at (pyshk.models.Shake attribute), 21
updated_at_iso (pyshk.models.Shake attribute), 21
User (built-in class), 13
User (class in pyshk.models), 22