friendlypins

Release 2.0.0

Kevin S. Phillips

Mar 07, 2021

CONTENTS:

CHAPTER

ONE

FRIENDLYPINS

1.1 friendlypins package

1.1.1 Subpackages

friendlypins.scripts package

Submodules

friendlypins.scripts.fpins module

Command line interface for the friendly pins APIs

`friendlypins.scripts.fpins.`**`configure_logging`**`(`*verbosity*`)`

Configures the global logger for the application

Parameters **`verbosity`** (*`int`*) – numeric value for the verbosity level for the log the larger the number, the more verbose the output

`friendlypins.scripts.fpins.`**`get_args`**`(`*args*`)`

Helper method used to parse command line parameters

Parameters **`args`** (*`str`*) – optional command line arguments to be parsed if not provided, args will be parsed from the console

Returns parsed arguments

Return type argparse.ArgumentParser

`friendlypins.scripts.fpins.`**`main`**`(`*args=None*`)`

Entry point function

Parameters **`args`** (*`str`*) – sample command line parameters to use when launching the tool used for debug and testing purposes only When not provided, arguments will be parsed from the command line

Returns error code produced by completing the given operation

Return type int

friendlypins.scripts.webp2jpeg module

Command line tool for converting webp images into jpegs

WARNING This is a prototype, that needs to be more fully tested before it should be considered production ready.

`friendlypins.scripts.webp2jpeg.`**`configure_logging`**(*verbosity*)

Configures the global logger for the application

Parameters **`verbosity`** (*`int`*) – numeric value for the verbosity level for the log the larger the number, the more verbose the output

`friendlypins.scripts.webp2jpeg.`**`get_args`**(*args*)

Helper method used to parse command line parameters

Parameters **`args`** (*`str`*) – optional command line arguments to be parsed if not provided, args will be parsed from the console

Returns parsed arguments

Return type argparse.ArgumentParser

`friendlypins.scripts.webp2jpeg.`**`main`**(*args=None*)

Entry point function

Parameters **`args`** (*`str`*) – sample command line parameters to use when launching the tool used for debug and testing purposes only When not provided, arguments will be parsed from the command line

Returns error code produced by completing the given operation

Return type int

Module contents

Command line interfaces for interacting with project APIs

friendlypins.utils package

Submodules

friendlypins.utils.base_object module

Base class exposing common functionality to all Pinterest primitives

`class` `friendlypins.utils.base_object.`**`BaseObject`**(*url*, *rest_io*, *json_data=None*)

Bases: `object`

Common base class providing shared functionality for all Pinterest primitives

Parameters

- **`url`** (*`str`*) – URL for this object, relative to the API root
- **`rest_io`** (`RestIO`) – reference to the Pinterest REST API
- **`json_data`** (*`dict`*) – Optional JSON response data describing this object if not provided, the class will lazy load response data when needed

static default_fields()

Default implementation

static default_url(*unique_id*)

Default implementation

classmethod from_json(*json_data*, *rest_io*)

Factory method that instantiates an instance of this class from JSON response data loaded by the caller

Parameters

- **json_data** (*dict*) – pre-loaded response data describing the object
- **rest_io** (RestIO) – pre-initialized session object for communicating with the REST API

Returns instance of this derived class, pre-initialized with the provided response data

property json

returns raw json representation of this object

Type dict

refresh()

Updates cached response data describing the state of this pin

NOTE: This method simply clears the internal cache, and updated information will automatically be pulled on demand as additional queries are made through the API

friendlypins.utils.console_actions module

Worker methods used to perform actions performed by fpins console app

friendlypins.utils.console_actions.**check_rate_limit**(*api_token*)

Checks to see when the next rate limit renewal is to occur

Parameters **api_token** (*str*) – Authentication token for the user who owns the board

Returns 0 if the operation succeeded, otherwise an error code

Return type int

friendlypins.utils.console_actions.**create_board**(*api_token*, *board_name*)

Creates a new board

Parameters

- **api_token** (*str*) – Authentication token for the user who owns the board
- **board_name** (*str*) – Name of the board to create

Returns 0 if the board was created, otherwise an error code is returned

Return type int

friendlypins.utils.console_actions.**delete_board**(*api_token*, *board_name*)

Deletes a board owned by a specific user

Parameters

- **api_token** (*str*) – Authentication token for the user who owns the board
- **board_name** (*str*) – Name of the board to delete

Returns 0 if the board was deleted, otherwise an error code is returned

Return type int

`friendlypins.utils.console_actions.`**`download_thumbnails`**(*api_token*, *board_name*, *output_folder*)

Downloads thumbnails of all pins on a board

Parameters

- **`api_token`** (*`str`*) – Authentication token for accessing the Pinterest API
- **`board_name`** (*`str`*) – name of the board containing the pins to process
- **`output_folder`** (*`str`*) – path where the thumbnails are to be downloaded

Returns status code describing the result of the action. zero on success, non-zero on failure

Return type int

friendlypins.utils.rest_io module

Abstraction around the raw Pinterest REST API calls

`class` `friendlypins.utils.rest_io.`**`RestIO`**(*authentication_token*)

Bases: `object`

Interface for low level REST API interactions

Parameters **`authentication_token`** (*`str`*) – Personal API token for authenticating to REST API

`delete`(*path*)

Sends a delete request to a remote endpoint

Parameters **`path`** (*`str`*) – API endpoint to send delete request to

`get`(*path*, *properties=None*)

Gets API data from a given sub-path

Parameters

- **`path`** (*`str`*) – sub-path with in the REST API to query
- **`properties`** (*`dict`*) – optional set of request properties to append to the API call

Returns json data returned from the API endpoint

Return type dict

`get_pages`(*path*, *properties=None*)

Generator for iterating over paged results returned from API

Parameters

- **`path`** (*`str`*) – sub-path with in the REST API to query
- **`properties`** (*`dict`*) – optional set of request properties to append to the API call

Returns json data returned from the API endpoint

Return type dict

`property headers`

the HTTP headers from the most recent API operation

Type *Headers*

post (*path*, *data*, *properties=None*)
Posts API data to a given sub-path

Parameters

- **path** (*str*) – sub-path with in the REST API to send data to
- **data** (*str*) – form data to be posted to the API endpoint
- **properties** (*dict*) – optional set of request properties to append to the API call

Returns json data returned from the API endpoint

Return type dict

refresh_headers ()
Forces an update to the cached headers

property root_url
canonical url for the REST API

Type str

property token
authentication token

Type str

Module contents

Helper scripts used by various parts of the API

1.1.2 Submodules

friendlypins.api module

Primary entry point for the Friendly Pinterest library

class friendlypins.api.**API** (*personal_access_token*)
Bases: object

High level abstraction for the core Pinterest API

Parameters personal_access_token (*str*) – API authentication token used for secure access to a users' Pinterest data

get_board_by_id (*board_id*)
Locates a specific Pinterest board given it's internal identifier

NOTE: this API assumes that the ID provided references a valid board. If it does not, the object returned will be invalid and any attempts to access data from the board will result in an error.

Parameters board_id (*int*) – the unique identifier for the board

Returns reference to the Pinterest board

Return type *Board*

get_pin_by_id (*pin_id*)
Locates a specific Pinterest pin given it's internal identifier

NOTE: this API assumes that the ID provided references a valid pin. If it does not, the object returned will be invalid and any attempts to access data from the pin will result in an error.

Parameters **pin_id** (*int*) – the unique identifier for the board

Returns reference to the Pinterest pin

Return type *Pin*

property rate_limit_refresh

Gets the time when the next refresh for API queries takes effect

Type datetime.datetime

property transaction_limit

Gets the total number of transactions per hour we're allotted

Type int

property transaction_remaining

Gets the total number of transactions per hour we're allotted

Type int

property user

Gets all primitives associated with the authenticated user

Type *User*

friendlypins.board module

Primitives for interacting with Pinterest boards

class friendlypins.board.**Board**(*url*, *rest_io*, *json_data=None*)

Bases: *friendlypins.utils.base_object.BaseObject*

Abstraction around a Pinterest board

Parameters

- **url** (*str*) – URL for this object, relative to the API root
- **rest_io** (RestIO) – reference to the Pinterest REST API
- **json_data** (*dict*) – Optional JSON response data describing this object if not provided, the class will lazy load response data when needed

property creation_date

when this board was created

Type datetime.datetime

static default_fields()

list (str): list of fields we pre-populate when loading board data

static default_url(*unique_id*)

Generates a URL for the REST API endpoint for a board with a given identification number

Parameters **unique_id** (*int*) – unique ID for the board

Returns URL for the API endpoint

Return type str

delete()

Removes this board and all pins attached to it

property description

The descriptive text associated with this board

Type str

property name

The name of the board

Type str

property num_collaborators

number of people with edit permissions to this board

Type int

property num_followers

number of people following this board

Type int

property num_pins

The total number of pins linked to this board

Type int

property pins

pins linked to this board

Yields *Pin* – generator that lazy loads the definitions for every pin on this board

property privacy_setting

description of the restriction / privacy level of the board

Type str

property unique_id

The unique identifier associated with this board

Type int

property url

Web address for the UI associated with the dashboard

Type str

friendlypins.exceptions module

custom exceptions provided by our library

exception `friendlypins.exceptions.`**`RateLimitException`**`(`*response*`)`

Bases: `requests.exceptions.HTTPError`

Helper class for extrapolating details about a rate limit error

Initialize RequestException with *request* and *response* objects.

property friendly_error

user friendly representation of the error message

Type str

property pin_headers

reference to the pre-parsed HTTP headers returned from the API

Type *Headers*

property rate_renewal
time, in the current locale, when requests will be allowed once again

Type datetime.datetime

friendlypins.headers module

Primitives for operation in HTTP response headers provided by Pinterest

class `friendlypins.headers.`**`Headers`**(*data*)
Bases: `object`

Abstraction around the Pinterest API HTTP response header

Parameters **data** (*dict*) – Header data parsed from the HTTP response

property bytes
Gets the number of bytes contained in the response data

Type int

property date
Date/time when this header was last populated

Type datetime.datetime

property percent_rate_remaining
percentage representation of the number of API requests left

See *rate_remaining()* for more info

Type int

property rate_limit
number of API requests in total permitted for the authenticated user

Type int

property rate_remaining
Number of API requests the authenticated user is allowed to make

The remaining number of requests resets every hour, according to the API docs

Type int

property time_to_refresh
the time stamp of when the rate limiting threshold is renewed

Type datetime.datetime

friendlypins.pin module

Primitives for operating on Pinterest pins

class `friendlypins.pin.`**`Pin`**(*url*, *rest_io*, *json_data=None*)
Bases: *friendlypins.utils.base_object.BaseObject*

Abstraction around a Pinterest pin

Parameters

- **url** (*str*) – URL for this object, relative to the API root
- **rest_io** (`RestIO`) – reference to the Pinterest REST API

- **json_data** (*dict*) – Optional JSON response data describing this object if not provided, the class will lazy load response data when needed

static default_fields()

list (str): list of fields we pre-populate when loading pin data

static default_url(*unique_id*)

Generates a URL for the REST API endpoint for a pin with a given identification number

Parameters **unique_id** (*int*) – unique ID for the pin

Returns URL for the API endpoint

Return type str

delete()

Removes this pin from it's respective board

property link

Source URL containing the original data for the pin

Type str

property media_type

descriptor for the type of data stored in the pin's link

Returns None if the type of data associated with the Pin is unknown

Type str

property note

Descriptive text associated with pin

Type str

property thumbnail

the thumbnail image associated with this pin

Type *Thumbnail*

property unique_id

The unique identifier associated with this pin

Type int

property url

Web address for the UI associated with the pin

Type str

friendlypins.thumbnail module

Primitives for operating on Pinterest pins

class friendlypins.thumbnail.**Thumbnail**(*data*)

Bases: object

Abstraction around a Pin's thumbnail

Parameters **data** (*dict*) – Raw Pinterest API data describing a thumbnail

property height

The height of the thumbnail image, in pixels

Type int

property url

Source URL where the thumbnail image can be found

Type str

property width

The width of the thumbnail image, in pixels

Type int

friendlypins.user module

Interfaces for interacting with Pinterest users

class friendlypins.user.**User**(*url*, *rest_io*)

Bases: object

Abstraction around a Pinterest user and their associated data

Parameters

- **url** (*str*) – URL for this user, relative to the API root
- **rest_io** (RestIO) – reference to the Pinterest REST API

property account_type

type of Pinterest account (ie: individual / business)

Type str

property bio

description of who this user is

Type str

property boards

Generator for iterating over the boards owned by this user

Type *Board*

create_board(*name*, *description=None*)

Creates a new board for the currently authenticated user

Parameters

- **name** (*str*) – name for the new board
- **description** (*str*) – optional descriptive text for the board

Returns reference to the newly created board

Return type *Board*

property created

when this user's profile was created

Type datetime.datetime

static default_fields()

list (str): list of fields we pre-populate when loading user data

property first_name

the first name of the user

Type str

property last_name
the last name of the user

Type str

property name
the full name of the user

alias for first_name + last_name

Type str

property num_boards
the total number of boards owned by this user

Type int

property num_followers
number of people following this Pinterest user

Type int

property num_pins
the total number of pins owned by this user

Type int

refresh()
Updates cached response data describing the state of this user

NOTE: This method simply clears the internal cache, and updated information will automatically be pulled on demand as additional queries are made through the API

property unique_id
Gets the internal unique ID associated with the user

Type int

property url
the URL of the users profile

Type str

property username
display name, used for logging in to Pinterest

Type str

friendlypins.version module

1.1.3 Module contents

Package definition for the project

CHAPTER TWO

OVERVIEW

User friendly Python wrapper for the Pinterest developer APIs. This project is currently in very early development, with only a bare minimum of functionality, intended for some of my personal projects. I'm sharing it here in case someone finds the implementation helpful for their projects. Also, if anyone would like to contribute to this pet project feel free to fork it and send me pull requests for any changes you may like.

Comments, suggestions and bugs may be reported to the project maintainer

CHAPTER THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX