Brickfront Documentation

Release 0.0.4

Callum Bartlett

Sep 15, 2017

Contents:

CHAPTER 1

Intro

Brickfront is a basic interface for working through Brickset's API. It should be pretty simple to use, though I do have the code reference if there's anyhing too difficult.

Installation:

```
pip install brickfront
```

CHAPTER 2

Getting Started

There's quite basic usage. For most things you don't need an API key, but for others you may need to get one.

First you need to make a *Client* object.

```
>>> import brickfront
>>> client = brickfront.Client(API_KEY)
```

From there, you can make requests through your client to be able to get different sets.

```
>>> setList = client.getSets(query='star wars')
>>> len(setList)
20
>>> build = setList[18]
>>> build.name
"Jabba's Palace"
>>> build.setID
'2141'
>>> build.pieces
231
>>> build.priceUK
'27.99'
```

Most code is fully internally documented, so it will autofill and properly interface with Python's *help* function.

CHAPTER 3

Code Reference

Client

class `brickfront.client.`**`Client`**(*apiKey: str*)

A frontend authenticator for Brickset.com's API. All endpoints require an API key.

Parameters **`apiKey`** (`str`) – The API key you got from Brickset.

Raises ***`brickfront.errors.InvalidKey`*** – If the key provided is invalid.

`checkKey`() → bool

Checks that an API key is valid.

Returns Whether the key is valid or not.

Return type *bool*

`getAdditionalImages`(*setID: str*) → list

Gets a list of URLs containing images of the set.

Parameters **`setID`** (`str`) – The ID of the set you want to grab the images for.

Returns A list of URL strings.

Return type List[*str*]

Warning: An empty list will be returned if there are no additional images, or if the set ID is invalid.

`getInstructions`(*setID: str*) → list

Get the instructions for a set.

Parameters **`setID`** (`str`) – The ID for the set you want to get the instructions of.

Returns A list of URLs to instructions.

Return type List[*str*]

> **Warning:** An empty list will be returned if there are no instructions, or if the set ID is invalid.

getRecentlyUpdatedSets(*minutesAgo: int*) → list

Gets the information of recently updated sets.

Parameters **minutesAgo** (*int*) – The amount of time ago that the set was updated.

Returns A list of sets that were updated within the given time.

Return type List[*brickfront.build.Build*]

getReviews(*setID: str*) → list

Get the reviews for a set.

Parameters **setID** (*str*) – The ID of the set you want to get the reviews of.

Returns A list of reviews.

Return type List[*brickfront.review.Review*]

> **Warning:** An empty list will be returned if there are no reviews, or if the set ID is invalid.

getSet(*setID: str*) → brickfront.build.Build

Gets the information of one build, using its Brickset set ID.

Parameters **setID** (*str*) – The ID of the build from Brickset.

Returns A single LEGO set object.

Return type *brickfront.build.Build*

Raises ***brickfront.errors.InvalidSetID*** – If no sets exist by that ID.

getSets(***kwargs*) → list

A way to get different sets from a query. All parameters are optional, but you should *probably* use some.

Parameters

- **query** (*str*) – The thing you're searching for.
- **theme** (*str*) – The theme of the set.
- **subtheme** (*str*) – The subtheme of the set.
- **setNumber** (*str*) – The LEGO set number.
- **year** (*str*) – The year in which the set came out.
- **owned** (*str*) – Whether or not you own the set. Only works when logged in with `login()`. Set to *1* to make true.
- **wanted** (*str*) – Whether or not you want the set. Only works when logged in with `login()`. Set to *1* to make true.
- **orderBy** (*str*) – How you want the set ordered. Accepts 'Number', 'YearFrom', 'Pieces', 'Minifigs', 'Rating', 'UKRetailPrice', 'USRetailPrice', 'CARetailPrice', 'EURetailPrice', 'Theme', 'Subtheme', 'Name', 'Random'. Add 'DESC' to the end to sort descending, e.g. NameDESC. Case insensitive. Defaults to 'Number'.
- **pageSize** (*str*) – How many results are on a page. Defaults to 20.
- **pageNumber** (*str*) – The number of the page you're looking at. Defaults to 1.

- **userName** (*str*) – The name of a user whose sets you want to search.

Returns A list of LEGO sets.

Return type List[*brickfront.build.Build*]

login (*username: str*, *password: str*) → bool

Logs into Brickset as a user, returning a userhash, which can be used in other methods. The userhash is stored inside the client.

Parameters

- **username** (*str*) – Your Brickset username.
- **password** (*str*) – Your Brickset password.

Returns A boolean value of whether or not the login request was done properly.

Return type *bool*

Raises ***brickfront.errors.InvalidLogin*** – If your login details are incorrect.

Build

class brickfront.build.**Build** (*data*, *userHash*, *client*)

A class holding the information of a LEGO set. Some attributes may be *None*. Check them before using them.

Called automatically by other functions. Do not manually call.

Variables

- **setID** – The *int* ID of the set on Brickset.
- **number** – The number of the set on Brickset, held as a *str*.
- **variant** – The *int* variant of the LEGO set.
- **year** – The *str* year in which the set came out.
- **theme** – *str* The theme of the set.
- **themeGroup** – *str* The type of the theme.
- **subtheme** – *str* Any other theme that the set may belong to.
- **pieces** – *int* The number of pieces in the set.
- **minifigs** – *int* The number of minifigs that the set may come with.
- **imageURL** – *str* The URL of the image of the set.
- **bricksetURL** – *str* The link to the Brickset page of the set.
- **released** – *bool* Whether or not the set has been released.
- **priceUK** – *str* The UK price of the set.
- **priceUS** – *str* The US price of the set.
- **priceCA** – *str* The CA price of the set.
- **priceEU** – *str* The EU price of the set.
- **rating** – *float* The Brickset rating of the LEGO set.
- **additionalImages** – List[*str*]

- **`reviews`** – List[*brickfront.review.Review*]
- **`instructions`** – List[*str*]

Review

class `brickfront.review.`**`Review`**(*data*)

A class holding data for a review.

Variables

- **`author`** – The *str* name of the person who wrote the review.
- **`datePosted`** – The *str* date that the review was posted.
- **`overallRating`** – The *int* rating that was given.
- **`parts`** – *int*
- **`buildingExperience`** – *int*
- **`playability`** – *int*
- **`valueForMoney`** – *int*
- **`title`** – *str*
- **`review`** – *str*
- **`HTML`** – *bool*

Exceptions

exception `brickfront.errors.`**`InvalidKey`**

The API key provided was invalid and cannot be used.

exception `brickfront.errors.`**`InvalidLogin`**

The login credentials used were invalid.

exception `brickfront.errors.`**`InvalidRequest`**

The request that was sent to the server was invalid.

exception `brickfront.errors.`**`InvalidSetID`**

The set ID that was passed is invalid - eg it doesn't exist.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index