

---

# **pyunsplash Documentation**

***Release 1.0.0rc1***

**Salvatore Ventura <salvoventura@gmail.com>**

**Jun 02, 2020**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Dependencies</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
3.1	Creating an instance . . . . .	7
3.2	Authorization workflow . . . . .	7
3.3	Error handling . . . . .	7
3.4	Library logging . . . . .	7
<b>4</b>	<b>Examples</b>	<b>9</b>
<b>5</b>	<b>API: Class PyUnsplash</b>	<b>11</b>
5.1	<b>PyUnsplash(api_key)</b> . . . . .	11
5.2	Methods and properties . . . . .	11
5.2.1	<b>PyUnsplash.user(username, w, h)</b> . . . . .	12
5.2.2	<b>PyUnsplash.collections(type_, page, per_page)</b> . . . . .	12
5.2.3	<b>PyUnsplash.photos(type_, page, per_page, order_by)</b> . . . . .	13
5.2.3.1	<i>Generic Photos</i> . . . . .	13
5.2.3.2	<i>Random Photos</i> . . . . .	13
5.2.3.3	<i>Single Photo</i> . . . . .	14
5.2.4	<b>PyUnsplash.search(type_, type, page, per_page, query)</b> . . . . .	14
5.2.5	<b>PyUnsplash.stats()</b> . . . . .	15
<b>6</b>	<b>API: Class Collection</b>	<b>17</b>
6.1	Methods and properties . . . . .	17
6.1.1	<b>Collection.refresh()</b> . . . . .	17
6.1.2	<b>Collection.id</b> . . . . .	18
6.1.3	<b>Collection.title</b> . . . . .	18
6.1.4	<b>Collection.description</b> . . . . .	19
6.1.5	<b>Collection.user</b> . . . . .	19
6.1.6	<b>Collection.link_photos</b> . . . . .	19
6.1.7	<b>Collection.link_related</b> . . . . .	20
6.1.8	<b>Collection.photos(page, per_page, order_by)</b> . . . . .	20
6.1.9	<b>Collection.related()</b> . . . . .	21
<b>7</b>	<b>API: Class Collections</b>	<b>23</b>

7.1	Methods and properties	23
7.1.1	<b>Collections.entries</b>	23
<b>8</b>	<b>API: Class Photo</b>	<b>25</b>
8.1	Methods and properties	25
8.1.1	<b>Photo.refresh()</b>	25
8.1.2	<b>Photo.id</b>	26
8.1.3	<b>Photo.link_html</b>	26
8.1.4	<b>Photo.link_download</b>	27
8.1.5	<b>Photo.stats</b>	27
8.1.6	<b>Photo.get_attribution(format='txt')</b>	28
8.1.7	<b>Photo.body</b>	28
<b>9</b>	<b>API: Class Photos</b>	<b>29</b>
9.1	Methods and properties	29
9.1.1	<b>Photos.entries</b>	29
<b>10</b>	<b>API: Class Search</b>	<b>31</b>
10.1	Methods and properties	31
10.1.1	<b>Search.entries</b>	31
<b>11</b>	<b>API: Class Stats</b>	<b>33</b>
11.1	Methods and properties	33
11.1.1	<b>Stats.total</b>	33
11.1.2	<b>Stats.body</b>	34
<b>12</b>	<b>API: Class User</b>	<b>35</b>
12.1	Methods and properties	35
12.1.1	<b>User.refresh()</b>	35
12.1.2	<b>User.id</b>	36
12.1.3	<b>User.links</b>	36
12.1.4	<b>User.link_html</b>	36
12.1.5	<b>User.link_portfolio</b>	37
12.1.6	<b>User.link_followers</b>	37
12.1.7	<b>User.link_following</b>	38
12.1.8	<b>User.link_photos</b>	38
12.1.9	<b>User.photos(page, per_page, order_by)</b>	39
12.1.10	<b>User.followers()</b>	39
12.1.11	<b>User.following()</b>	40
12.1.12	<b>User.likes(page, per_page, order_by)</b>	40
12.1.13	<b>User.collections(page, per_page, order_by)</b>	41
12.1.14	<b>User.statistics(resolution, quantity)</b>	41
12.1.15	<b>User.body</b>	42
<b>13</b>	<b>API: Class Users</b>	<b>43</b>
13.1	Methods and properties	43
13.1.1	<b>Users.entries</b>	43
<b>14</b>	<b>API: Class UserStatistics</b>	<b>45</b>
14.1	Methods and properties	45
14.1.1	<b>UserStatistics.downloads</b>	45
14.1.2	<b>UserStatistics.views</b>	46
14.1.3	<b>UserStatistics.likes</b>	46
<b>15</b>	<b>Version</b>	<b>47</b>





An open source Python wrapper for the [Unsplash REST API](#).

The source code is available on GitHub at <https://github.com/salvoventura/pyunsplash>.





# CHAPTER 1

---

## Installation

---

PyUnsplash is available on [PyPI](#) and thus can be installed with `pip` on most platforms.

```
$ pip install pyunsplash
```



## CHAPTER 2

---

### Dependencies

---

This library depends on [Requests](#) to make - well - requests to the Unsplash API. This additional package should be automatically installed at installation time, or you can simply install it by:

```
$ pip install requests
```



### 3.1 Creating an instance

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash Application ID>')
```

API keys can be obtained from [Unsplash Developers](#).

---

### 3.2 Authorization workflow

TODO

---

### 3.3 Error handling

TODO

---

### 3.4 Library logging

The PyUnsplash library internal logging subsystem is driven by the application. The default logging level of the library is set to be **logging.ERROR**. If you want to access the library logging subsystem, you can fine-tune the logger with `PyUnsplash.logger_name` as per the following example:

```
from pyunsplash import PyUnsplash

# Initialize app logging
logger = logging.getLogger()
logging.basicConfig(filename='app.log', level=logging.DEBUG)

# pyunsplash logger defaults to level logging.ERROR
# If you need to change that, use getLogger/setLevel
# on the module logger, like this:
logging.getLogger(PyUnsplash.logger_name).setLevel(logging.DEBUG)

pu = PyUnsplash(api_key='<your Unsplash Application ID>')
# ... continue
```

## CHAPTER 4

---

### Examples

---

This example shows how the interaction with the paging functionality of the unsplash API is greatly abstracted and simplified. The code below will iterate through all collections, and retrieve each photo in there, and print their download link.

```
import logging
from pyunsplash import PyUnsplash
api_key = 'YOUR_APPLICATION_ID'

# instantiate PyUnsplash object
py_un = PyUnsplash(api_key=api_key)

# Start with the generic collection, maximize number of items
# note: this will run until all photos of all collections have
#       been visited, unless a connection error occurs.
#       Typically the API hourly limit gets hit during this
#
collections = py_un.collections(per_page=30)
while collections.has_next:
    for collection in collections.entries:
        photos = collection.photos()
        for photo in photos.entries:
            print collection.title, photo.link_download, photo.get_attribution()

# no need to specify per_page: will take from original object
collections = collections.get_next_page()
```





---

API: Class PyUnsplash

---

This is the main class used to interact with the `Unsplash` REST API.

## 5.1 PyUnsplash(api\_key)

Create an instance of class `PyUnsplash`. The `api_key` parameter is required. API keys can be obtained from [Unsplash Developers](#).

### Parameters

Argument	Type	Optional/Required	Notes
<code>api_key</code>	string	required	Your application ID

### Returns

<b>Object</b>	Instance of class <code>PyUnsplash</code>
---------------	---

### Example

```
import pyunplash
pu = pyunplash.PyUnsplash(api_key='<your Unsplash API key>')
```

---

## 5.2 Methods and properties

Methods and properties exposed by the `PyUnsplash` class.

### 5.2.1 PyUnsplash.user(username, w, h)

To interact with the `user` API, create an instance of class `User`.

#### Parameters

Argument	Type	Optional/Required	Notes
<b>username</b>	string	required	The user's username.
<b>w</b>	number	optional	Profile image width in pixels.
<b>h</b>	number	optional	Profile image height in pixels.

#### Returns

<b>Object</b>	Instance of class <code>User</code>
---------------	-------------------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
```

### 5.2.2 PyUnsplash.collections(type\_, page, per\_page)

To interact with the `collections` API, create an instance of class `Collections`.

#### Parameters

Argument	Type	Optional/Required	Notes
<b>type_</b>	string	optional (default: generic) (default: generic)	generic, featured (curated has been deprecated)
<b>page</b>	number	optional (default: 1)	Page number to retrieve.
<b>per_page</b>	number	optional (default: 10)	Number of items per page (max: 30)

#### Returns

<b>Object</b>	Instance of class <code>Collections</code> , <code>CuratedCollections</code> , or <code>FeaturedCollections</code> depending on the value of <code>type</code> <b>NOTE</b> curated endpoint has been deprecated by Unsplash
---------------	--

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')

# retrieve a page from the featured collections, with a maximum
# of 5 collections per-page
collections_page = pu.collections(type_='featured', per_page=5)
```

### 5.2.3 PyUnsplash.photos(type\_, page, per\_page, order\_by)

To interact with the `photos` API, create an instance of class `Photos`.

**NOTE** `curated` endpoint has been deprecated by Unsplash

#### 5.2.3.1 Generic Photos

##### Parameters

Argument	Type	Optional/Required	Notes
<b>type_</b>	string	optional (default: generic)	generic
<b>page</b>	number	optional (default: 1)	Page number to retrieve.
<b>per_page</b>	number	optional (default: 10)	Number of items per page (max: 30)
<b>order_by</b>	string	optional (default: latest)	latest, oldest, popular

##### Returns

<b>Ob- ject</b>	Instance of class <code>Photos</code> depending on the value of <code>type</code> <b>NOTE</b> <code>curated</code> endpoint has been deprecated by Unsplash
---------------------	--

##### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')

# retrieve a page from the generic photos, with a maximum
# of 15 photos per-page
collections_page = pu.photos(type_='generic', per_page=15)
```

#### 5.2.3.2 Random Photos

##### Parameters

Argument	Type	Optional/Required	Notes
<b>type_</b>	string	required	random
<b>collections</b>	string	optional	Public collection ID(s) to filter selection If multiple, comma-separated
<b>featured</b>	bool	optional	Limit selection to featured photos Completely remove if you don't want featured Setting to False doesn't work
<b>user-name</b>	string	optional	Limit selection to a single user
<b>query</b>	string	optional	Limit selection to photos matching a search term
<b>orientation</b>	string	optional	landscape, portrait, squarish
<b>count</b>	number	required	Number of items per page (max: 30)

**Note** You can't use the `collections` and `query` parameters in the same call

##### Returns

<b>Object</b>	Instance of class RandomPhotos
---------------	--------------------------------

### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')

# retrieve 4 random photos, which are featured, and tagged as "dog"
collections_page = pu.photos(type_='random', count=4, featured=True, query=
    ↪ "dog")
for photo in photos.entries:
    print(photo.id, photo.link_download)
```

---

### 5.2.3.3 Single Photo

#### Parameters

Argument	Type	Optional/Required	Notes
<b>type_</b>	string	required	single
<b>photo_id</b>	string	required	ID of the photo you want to load

#### Returns

<b>Object</b>	Instance of class SinglePhoto
---------------	-------------------------------

### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')

# retrieve specific photo by its ID, and print the attribution
photo = pu.photos(type_="single", photo_id='10_kVknP02g')
print(photo.entries.get_attribution(format='txt'))
print(photo.entries.get_attribution(format='html'))
```

---

### 5.2.4 PyUnsplash.search(type\_, type, page, per\_page, query)

To interact with the `search` API, create an instance of class `Search`.

#### Parameters

Argument	Type	Optional/Required	Notes
<b>type_</b>	string	required	photos, collections, users
<b>page</b>	number	optional (default: 1)	Page number to retrieve.
<b>per_page</b>	number	optional (default: 10)	Number of items per page (max: 30)
<b>query</b>	string	optional	Search terms

#### Returns

<b>Object</b>	Instance of class Search
---------------	--------------------------

### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')

#
#
search = pu.search(type_='photos', query='red,car')
for entry in search.entries:
    print entry.link_html
```

---

## 5.2.5 PyUnsplash.stats()

To interact with the `stats` API, create an instance of class `Stats`.

### Parameters

Argument	Type	Optional/Required	Notes
N/A			

### Returns

<b>Object</b>	Instance of class Stats
---------------	-------------------------

### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')

stats = pu.stats()
print stats.total    # this is json
```



---

API: Class Collection

---

This class is used to interact with individual `Collection` objects typically returned by `PyUnsplash` as part of `Collections` object entries.

## 6.1 Methods and properties

Methods and properties exposed by the `Collection` class.

### 6.1.1 `Collection.refresh()`

Reload the full object from its REST API URI. Required if there has been an update to the object, or if you need to retrieve the full object starting from a simplified content copy coming from a list.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

None	
------	--

#### Example

```
import pyunplash
pu = pyunplash.PyUnsplash(api_key='<your Unsplash API key>')
collections_page = pu.collections(type_='featured', per_page=5)
for collection in collections_page.entries:
    collection.refresh()
    print collection.id
```

### 6.1.2 Collection.id

Unique identifier for this user resource.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	unique resource identifier
---------------	----------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
collections_page = pu.collections(type_='featured', per_page=5)
for collection in collections_page.entries:
    print collection.id
```

---

### 6.1.3 Collection.title

Title of this collection.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	title of this collection
---------------	--------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
collections_page = pu.collections(type_='featured', per_page=5)
for collection in collections_page.entries:
    print collection.title
```



### 6.1.4 Collection.description

Description for this collection.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	description for this collection
---------------	---------------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
collections_page = pu.collections(type_='featured', per_page=5)
for collection in collections_page.entries:
    print collection.description
```

---

### 6.1.5 Collection.user

User who created this collection.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	id of user who created this collection
---------------	--

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
collections_page = pu.collections(type_='featured', per_page=5)
for collection in collections_page.entries:
    print collection.user
```

---

### 6.1.6 Collection.link\_photos

API location of this collection's photos.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	url to the API location of this collection's photos
---------------	---

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
collections_page = pu.collections(type_='featured', per_page=5)
for collection in collections_page.entries:
    print collection.link_photos
```

---

### 6.1.7 Collection.link\_related

API location of this collection's related collections. (Non-curated collections only)

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	url to the API location of this collection's related collections
---------------	--

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
collections_page = pu.collections(type_='featured', per_page=5)
for collection in collections_page.entries:
    print collection.link_related
```

---

### 6.1.8 Collection.photos(page, per\_page, order\_by)

Allows easy access to each photo in this Collection. Returns Photos collection object from this collection's link\_photos url.

#### Parameters

Argument	Type	Optional/Required	Notes
<b>page</b>	number	optional (default: 1)	Page number to retrieve.
<b>per_page</b>	number	optional (default: 10)	Number of items per page (max: 30)
<b>order_by</b>	string	optional (default: latest)	latest, oldest, popular

**Returns**

<b>Object</b>	Instance of class Photos
---------------	--------------------------

**Example**

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
collections_page = pu.collections(type_='featured', per_page=5)
for collection in collections_page.entries:
    photos = collection.photos(order_by='popular', per_page=3)
    for photo in photos.entries:
        print photo.id, photo.link_download
```

---

### 6.1.9 Collection.related()

Allows easy access to Collection``s related to this ``Collection. Returns Collections object from this collection's link\_related url.

**Parameters**

Argument	Type	Optional/Required	Notes
N/A			

**Returns**

<b>Object</b>	Instance of class Collections
---------------	-------------------------------

**Example**

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
collections_page = pu.collections(type_='featured', per_page=5)
for collection in collections_page.entries:
    related_collections = collection.related
    for rel_collection in related_collections.entries:
        print rel_collection.title, rel_collection.description
```



---

## API: Class Collections

---

This class is used to interact with the `Unsplash /collections` REST API but also with lists of `Collection` objects as returned by certain `unsplash` REST APIs.

The constructor can be invoked through the main `PyUnsplash` class as `PyUnsplash.collections(type, page, per_page)`.

The following class methods/properties in `PyUnsplash` also return a `Collections` class object:

- `Collection.related()`
- `User.collections(page, per_page, order_by)`

### 7.1 Methods and properties

Methods and properties exposed by the `Collections` class.

#### 7.1.1 `Collections.entries`

Returns an iterator for the `Collection` objects contained in this `Collections` instance

##### Parameters

Argument	Type	Optional/Required	Notes
N/A			

##### Returns

iterator	each time an instance of class <code>Collection</code>
----------	--

##### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
collections = this_user.collections(page=1, per_page=5)
for collection in collections.entries:
    print collection.id, collection.title
```

This class is used to interact with individual `Photo` objects typically returned by `PyUnsplash` as part of `Photos` object entries.

## 8.1 Methods and properties

Methods and properties exposed by the `Photo` class.

### 8.1.1 `Photo.refresh()`

Reload the full object from its REST API URI. Required if there has been an update to the object, or if you need to retrieve the full object starting from a simplified content copy coming from a list.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

None	
------	--

#### Example

```
import pyunplash
pu = pyunplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
photos = this_user.photos() # photos is an instance of class Photos
for photo in photos.entries:
    photo.refresh()
    print photo.id, photo.link_download
```

### 8.1.2 Photo.id

Unique identifier for this user resource.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	unique resource identifier
---------------	----------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
photos = this_user.photos()    # photos is an instance of class Photos
for photo in photos.entries:
    photo.refresh()
    print photo.id, photo.link_download
```

---

### 8.1.3 Photo.link\_html

Link to the html representation of the resource.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	url to the html representation of the resource
---------------	--

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
photos = this_user.photos()    # photos is an instance of class Photos
for photo in photos.entries:
    print photo.id, photo.link_html
```



### 8.1.4 Photo.link\_download

Download location of this resource.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	url to the download location of the resource
---------------	--

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
photos = this_user.photos() # photos is an instance of class Photos
for photo in photos.entries:
    print photo.id, photo.link_download
```

---

### 8.1.5 Photo.stats

Retrieve a single photo's stats.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>json</b>	JSON encoded stats information about this photo
-------------	---

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
photos = this_user.photos() # photos is an instance of class Photos
for photo in photos.entries:
    print photo.stats
```

### 8.1.6 Photo.get\_attribution(format='txt')

Generate and return a standard attribution string according to 'format' parameter.

**Note** format parameter value 'str' has been deprecated in favor of 'txt'.

#### Parameters

Argument	Type	Optional/Required	Notes
format	string	optional	Valid values: 'txt', 'html'

#### Returns

<b>string</b>	Text or HTML standard attribution string.
---------------	---

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
photos = this_user.photos() # photos is an instance of class Photos
for photo in photos.entries:
    print photo.get_attribution() # Photo by salvatore_
    ↪ventura on Unsplash
    print photo.get_attribution(format='txt') # Photo by salvatore_
    ↪ventura on Unsplash
    print photo.get_attribution(format='html') # <span>Photo by <a href=
    ↪"https://unsplash.com/@salvoventura">salvatore ventura</a> on <a href=
    ↪"https://unsplash.com/@salvoventura">Unsplash</a></span>
```

---

### 8.1.7 Photo.body

This is the full object returned by the API in JSON format.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	Photo object returned by API, in JSON format
---------------	--

## API: Class Photos

This class is used to interact with the Unsplash /photos REST API but also with lists of Photo objects as returned by certain unsplash REST APIs.

The constructor can be invoked through the main PyUnsplash class as `PyUnsplash.photos(type, page, per_page, order_by)`.

The following class methods/properties in `PyUnsplash` also return a `Photos` class object:

- `User.likes(page, per_page, order_by)`
- `User.photos(page, per_page, order_by)`

### 9.1 Methods and properties

Methods and properties exposed by the `Photos` class.

#### 9.1.1 Photos.entries

Returns an iterator for the `Photo` objects contained in this `Photos` instance

##### Parameters

Argument	Type	Optional/Required	Notes
N/A			

##### Returns

iterator	each time an instance of class <code>Photo</code>
----------	---

##### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
photos = this_user.photos()      # photos is an instance of class Photos
for photo in photos.entries:
    print photo.id, photo.link_download
```

This class is used to interact with the Unsplash /search/ REST API.

The constructor is invoked through the main PyUnsplash class as `PyUnsplash.search(type, page, per_page, query)`

## 10.1 Methods and properties

Methods and properties exposed by the `Search` class.

### 10.1.1 Search.entries

Returns an iterator for the returned objects contained in this `Search` instance. Each entry will be an instance of class `Collection`, `Photo` or `User` according to the search type.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

iterator	each time an instance of class <code>User</code>
----------	--

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
search = pu.search(type_='photos', query='red,car')
for photo in search.entries:
    print photo.id, photo.link_download
```



This class is used to interact with the Unsplash `/stats/total` REST API.

The constructor is invoked through the main `PyUnsplash` class as `PyUnsplash.stats()`

## 11.1 Methods and properties

Methods and properties exposed by the `Stats` class.

### 11.1.1 `Stats.total`

Description

Parameters

Argument	Type	Optional/Required	Notes
N/A			

Returns

<b>json</b>	JSON encoded stats information about Unsplash.com
-------------	---

Example

```
import pyunplash
pu = pyunplash.PyUnsplash(api_key='<your Unsplash API key>')
stats = pu.stats()
print stats.total
```

### 11.1.2 Stats.body

This is the full object returned by the API in JSON format.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	Stats object returned by API, in JSON format
---------------	--



This class is used to interact with the Unsplash `/users/:username` REST API.

The constructor is invoked through the main `PyUnsplash` class as `PyUnsplash.user(username, w, h)`

## 12.1 Methods and properties

Methods and properties exposed by the `User` class.

### 12.1.1 `User.refresh()`

Reload the full object from its REST API URI. Required if there has been an update to the object, or if you need to retrieve the full object starting from a simplified content copy coming from a list.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

None	
------	--

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
this_user.refresh()
```

### 12.1.2 User.id

Unique identifier for this user resource.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	unique resource identifier
---------------	----------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
print this_user.id
```

---

### 12.1.3 User.links

Dictionary of the `links` section in the resource body.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>dict</b>	links section from the resource body
-------------	--------------------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
print this_user.links
```

---

### 12.1.4 User.link\_html

Link to the html representation of the resource.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

**Returns**

<b>string</b>	url to the html representation of the resource
---------------	--

**Example**

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
print this_user.link_html
```

---

### 12.1.5 User.link\_portfolio

Link to the user specified portfolio page.

**Parameters**

Argument	Type	Optional/Required	Notes
N/A			

**Returns**

<b>string</b>	url to the user portfolio
---------------	---------------------------

**Example**

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
print this_user.link_portfolio
```

---

### 12.1.6 User.link\_followers

API location of this user's followers.

**Parameters**

Argument	Type	Optional/Required	Notes
N/A			

**Returns**

<b>string</b>	url to the API location of this user's followers
---------------	--

**Example**

```
import pyunplash
pu = pyunplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
print this_user.link_followers
```

---

### 12.1.7 User.link\_following

API location of users this user is following.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	url to the API location of users this user is following
---------------	---

#### Example

```
import pyunplash
pu = pyunplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
print this_user.link_following
```

---

### 12.1.8 User.link\_photos

API location of this user's photos.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	url to the API location of this user's photos
---------------	---

#### Example

```
import pyunplash
pu = pyunplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
print this_user.link_photos
```

---

### 12.1.9 User.photos(page, per\_page, order\_by)

Allows easy access to each photo from this user. Returns `Photos` collection object from this user's `link_photos` url.

#### Parameters

Argument	Type	Optional/Required	Notes
<b>page</b>	number	optional (default: 1)	Page number to retrieve.
<b>per_page</b>	number	optional (default: 10)	Number of items per page (max: 30)
<b>order_by</b>	string	optional (default: latest)	latest, oldest, popular

#### Returns

<b>Object</b>	Instance of class <code>Photos</code>
---------------	---------------------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
photos = this_user.photos(per_page=5)
for photo in photos.entries:
    print photo.id, photo.link_download
```

---

### 12.1.10 User.followers()

Allows easy access to each user that is a follower of this user. Returns `Users` collection object from this user's `link_followers` url.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>Object</b>	Instance of class <code>Users</code>
---------------	--------------------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
followers = this_user.followers()
for user in followers.entries:
    print user.id, user.body.get('first_name'), user.body.get('last_name')
```

### 12.1.11 User.following()

Allows easy access to each user that is followed by this user. Returns `Users` collection object from this user's `link_following` url.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>Object</b>	Instance of class <code>Users</code>
---------------	--------------------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
following = this_user.following()
for user in following.entries:
    print user.id, user.body.get('first_name'), user.body.get('last_name')
```

---

### 12.1.12 User.likes(page, per\_page, order\_by)

Allows easy access to each photo liked by this user. Returns `Photos` collection object.

#### Parameters

Argument	Type	Optional/Required	Notes
<b>page</b>	number	optional (default: 1)	Page number to retrieve.
<b>per_page</b>	number	optional (default: 10)	Number of items per page (max: 30)
<b>order_by</b>	string	optional (default: latest)	latest, oldest, popular

#### Returns

<b>Object</b>	Instance of class <code>Photos</code>
---------------	---------------------------------------

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
photos = this_user.likes(per_page=5)
for photo in photos.entries:
    print photo.id, photo.link_download
```

### 12.1.13 User.collections(page, per\_page, order\_by)

Allow easy access to the collections created by this user. Returns `Collections` object.

#### Parameters

Argument	Type	Optional/Required	Notes
<b>page</b>	number	optional (default: 1)	Page number to retrieve.
<b>per_page</b>	number	optional (default: 10)	Number of items per page (max: 30)

#### Returns

<b>Object</b>	Instance of class <code>Collections</code>
---------------	--

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
collections = this_user.collections(page=1, per_page=5)
for collection in collections.entries:
    print collection.id, collection.title
```

---

### 12.1.14 User.statistics(resolution, quantity)

Retrieve the consolidated number of downloads, views and likes of this user's photos, as well as the historical breakdown and average of these stats in a specific timeframe (default is 30 days).

Returns `UserStatistics` object.

#### Parameters

Argument	Type	Optional/Required	Notes
<b>resolution</b>	number	optional (default: <code>days</code> )	The frequency of the stats.
<b>quantity</b>	number	optional (default: 30)	The amount of for each stat.

**NOTE** Currently, the only resolution param supported is `days` and the quantity param can be any number between 1 and 30.

#### Returns

<b>Object</b>	Instance of class <code>UserStatistics</code>
---------------	---

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
user_statistics = this_user.statistics()
```

### 12.1.15 User.body

This is the full object returned by the API in JSON format.

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

<b>string</b>	User object returned by API, in JSON format
---------------	---



This class is used to interact with lists of `User` objects as returned by certain `unsplash` REST APIs.

The following class methods/properties in `PyUnsplash` return a `Users` class object:

- `User.followers()`
- `User.following()`

## 13.1 Methods and properties

Methods and properties exposed by the `Users` class.

### 13.1.1 `Users.entries`

Returns an iterator for the `User` objects contained in this `Users` instance

#### Parameters

Argument	Type	Optional/Required	Notes
N/A			

#### Returns

iterator	each time an instance of class <code>User</code>
----------	--

#### Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura', w=100, h=100)
```

(continues on next page)

(continued from previous page)

```
followers = this_user.followers()    # followers is an instance of class_
↳ Users
for user in followers.entries:
    print user.id, user.body.get('first_name'), user.body.get('last_name')
```

---

API: Class UserStatistics

---

This class is used to interact with the Unsplash `/users/:username/statistics` REST API.

The constructor is invoked through the `PyUnsplash.User` class as `PyUnsplash.User.statistics()`

## 14.1 Methods and properties

Methods and properties exposed by the `UserStatistics` class.

### 14.1.1 UserStatistics.downloads

Description

Parameters

Argument	Type	Optional/Required	Notes
N/A			

Returns

<b>dict</b>	Dictionary containing statistic data from <i>downloads</i>
-------------	--

Example

```
import pyunplash
pu = pyunplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura')
user_statistics = this_user.statistics()
print (user_statistics.downloads.get('total'))
print (user_statistics.downloads.get('historical', {}).get('change', None))
```

### 14.1.2 UserStatistics.views

Description

Parameters

Argument	Type	Optional/Required	Notes
N/A			

Returns

<b>dict</b>	Dictionary containing statistic data from <i>views</i>
-------------	--

Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura')
user_statistics = this_user.statistics()
print (user_statistics.views.get('total'))
print (user_statistics.views.get('historical', {}).get('change', None))
```

---

### 14.1.3 UserStatistics.likes

Description

Parameters

Argument	Type	Optional/Required	Notes
N/A			

Returns

<b>dict</b>	Dictionary containing statistic data from <i>likes</i>
-------------	--

Example

```
import pyunsplash
pu = pyunsplash.PyUnsplash(api_key='<your Unsplash API key>')
this_user = pu.user('salvoventura')
user_statistics = this_user.statistics()
print (user_statistics.likes.get('total'))
print (user_statistics.likes.get('historical', {}).get('change', None))
```

## CHAPTER 15

---

### Version

---

#### **PyUnsplash v1.0.0rc2 (release candidate, v2)**

This is the second release candidate for production ready 1.0.0 of PyUnsplash. Thanks to all the early adopters and bug reporters: thanks to you there have been a number of improvements and bugfixes and we are progressing the rc process.

This release brings mainly one bugfix:

- #13 `search.get_next_page()` does not work

**Todo** - Authorization workflow & Current user - Write operations



## CHAPTER 16

---

### License

---

PyUnsplash is released under the [MIT License](#).