
libgravatar Documentation

Release 0.2.3

Pablo SEMINARIO

Apr 04, 2018

Contents

1 Indices and tables	3
Python Module Index	5

Contents: A library that provides a Python 3 interface to the Gravatar APIs.

class libgravatar.**Gravatar** (*email*)

This class encapsulates all the unauthenticated methods from the API.

get_image (*size=80, default="", force_default=False, rating="", filetype_extension=False, use_ssl=False*)

Returns an URL to the user profile image.

```
>>> g = Gravatar('myemailaddress@example.com')
>>> g.get_image()
'http://www.gravatar.com/avatar/0bc83cb571cd1c50ba6f3e8a78ef1346'
```

With *size* you can request a specific image size, by default, images are presented at 80px by 80px. You may request image anywhere from 1px up to 2048px.

The *default* parameter is used to supply a default image when an email address has no match Gravatar image. *default* can be an URL or one of the built in options *404*, *mm*, *identicon*, *monsterid*, *wavatar*, *robohash*, *retro* or *blank*.

force_default force the default image to always load.

rating can be used to request images by audience type. Possible values are *g*, *pg*, *r* or *x*. By default, only *g* rated images are displayed.

filetype_extension add an optional *.jpg* extension.

use_ssl can be used to request images via SSL.

See more details on [Gravatar Image Requests](#).

get_profile (*data_format=""*)

Returns an URL to the profile information associated with the Gravatar account.

```
>>> g = Gravatar('myemailaddress@example.com')
>>> g.get_profile()
'http://www.gravatar.com/0bc83cb571cd1c50ba6f3e8a78ef1346'
```

See more details on [Gravatar Profile Requests](#).

class libgravatar.**GravatarXMLRPC** (*email, apikey="", password=""*)

This class encapsulates all the authenticated methods from the XML-RPC API.

API details: <http://en.gravatar.com/site/implement/xmlrpc>

addresses ()

Gets a list of addresses for this account.

exists (*hashes*)

Checks whether a hash has a gravatar.

test ()

Test the API.

userimages ()

Returns a dict of userimages for this account.

libgravatar.**default_url_is_valid** (*url*)

Gravatar conditions for valid default URLs.

```
>>> default_url_is_valid('http://example.com/images/avatar.jpg')
True
```

`libgravatar.md5_hash` (*string*)

Returns a md5 hash from a string.

```
>>> md5_hash('myemailaddress@example.com')
'0bc83cb571cd1c50ba6f3e8a78ef1346'
```

`libgravatar.sanitize_email` (*email*)

Returns an e-mail address in lower-case and strip leading and trailing whitespaces.

```
>>> sanitize_email(' MyEmailAddress@example.com ')
'myemailaddress@example.com'
```

CHAPTER 1

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

|

libgravatar, 1

A

addresses() (libgravatar.GravatarXMLRPC method), 1

D

default_url_is_valid() (in module libgravatar), 1

E

exists() (libgravatar.GravatarXMLRPC method), 1

G

get_image() (libgravatar.Gravatar method), 1

get_profile() (libgravatar.Gravatar method), 1

Gravatar (class in libgravatar), 1

GravatarXMLRPC (class in libgravatar), 1

L

libgravatar (module), 1

M

md5_hash() (in module libgravatar), 1

S

sanitize_email() (in module libgravatar), 2

T

test() (libgravatar.GravatarXMLRPC method), 1

U

userimages() (libgravatar.GravatarXMLRPC method), 1