Help on package pylast:

NAME

pylast

DESCRIPTION

# pylast -

# A Python interface to Last.fm and Libre.fm

#

# Copyright 2008-2010 Amr Hassan

# Copyright 2013-2021 hugovk

#

# Licensed under the Apache License, Version 2.0 (the "License");

# you may not use this file except in compliance with the License.

# You may obtain a copy of the License at

#

# <https://www.apache.org/licenses/LICENSE-2.0>

#

# Unless required by applicable law or agreed to in writing, software

# distributed under the License is distributed on an "AS IS" BASIS,

# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

# See the License for the specific language governing permissions and

# limitations under the License.

#

# <https://github.com/pylast/pylast>

PACKAGE CONTENTS

CLASSES

builtins.Exception(builtins.BaseException)

PyLastError

MalformedResponseError

NetworkError

WSError

builtins.object

SessionKeyGenerator

builtins.tuple(builtins.object)

Image

ImageSizes

LibraryItem

LovedTrack

PlayedTrack

SimilarItem

TopItem

\_BaseObject(builtins.object)

Country

Library

\_Chartable(\_BaseObject)

Tag

User

AuthenticatedUser

\_Network(builtins.object)

LastFMNetwork

LibreFMNetwork

\_Opus(\_Taggable)

Album

Track

\_Search(\_BaseObject)

AlbumSearch

ArtistSearch

TrackSearch

\_Taggable(\_BaseObject)

Artist

class Album(\_Opus)

| Album(artist, title, network, username=None, info=None)

|

| An album.

|

| Method resolution order:

| Album

| \_Opus

| \_Taggable

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| \_\_init\_\_(self, artist, title, network, username=None, info=None)

| Create an opus instance.

| # Parameters:

| \* artist: An artist name or an Artist object.

| \* title: The album or track title.

| \* ws\_prefix: 'album' or 'track'

|

| get\_tracks(self)

| Returns the list of Tracks on this album.

|

| get\_url(self, domain\_name=0)

| Returns the URL of the album or track page on the network.

| # Parameters:

| \* domain\_name str: The network's language domain. Possible values:

| o DOMAIN\_ENGLISH

| o DOMAIN\_GERMAN

| o DOMAIN\_SPANISH

| o DOMAIN\_FRENCH

| o DOMAIN\_ITALIAN

| o DOMAIN\_POLISH

| o DOMAIN\_PORTUGUESE

| o DOMAIN\_SWEDISH

| o DOMAIN\_TURKISH

| o DOMAIN\_RUSSIAN

| o DOMAIN\_JAPANESE

| o DOMAIN\_CHINESE

|

| ----------------------------------------------------------------------

| Methods inherited from \_Opus:

|

| \_\_eq\_\_(self, other)

| Return self==value.

|

| \_\_ne\_\_(self, other)

| Return self!=value.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| \_\_str\_\_ = r(\*args)

|

| get\_artist(self)

| Returns the associated Artist object.

|

| get\_cover\_image(self, size=3)

| Returns a URI to the cover image

| size can be one of:

| SIZE\_EXTRA\_LARGE

| SIZE\_LARGE

| SIZE\_MEDIUM

| SIZE\_SMALL

|

| get\_listener\_count(self)

| Returns the number of listeners on the network

|

| get\_mbid(self)

| Returns the MusicBrainz ID of the album or track.

|

| get\_name(self, properly\_capitalized=False)

| Returns the album or track title (alias to get\_title()).

|

| get\_playcount(self)

| Returns the number of plays on the network

|

| get\_title(self, properly\_capitalized=False)

| Returns the artist or track title.

|

| get\_userplaycount(self)

| Returns the number of plays by a given username

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_Opus:

|

| artist = None

|

| title = None

|

| username = None

|

| ----------------------------------------------------------------------

| Methods inherited from \_Taggable:

|

| add\_tag(self, tag)

| Adds one tag.

| \* tag: a tag name or a Tag object.

|

| add\_tags(self, tags)

| Adds one or several tags.

| \* tags: A sequence of tag names or Tag objects.

|

| clear\_tags(self)

| Clears all the user-set tags.

|

| get\_tags(self)

| Returns a list of the tags set by the user to this object.

|

| get\_top\_tags(self, limit=None)

| Returns a list of the most frequently used Tags on this object.

|

| remove\_tag(self, tag)

| Remove a user's tag from this object.

|

| remove\_tags(self, tags)

| Removes one or several tags from this object.

| \* tags: a sequence of tag names or Tag objects.

|

| set\_tags(self, tags)

| Sets this object's tags to only those tags.

| \* tags: a sequence of tag names or Tag objects.

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class AlbumSearch(\_Search)

| AlbumSearch(album\_name, network)

|

| Search for an album by name.

|

| Method resolution order:

| AlbumSearch

| \_Search

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, album\_name, network)

| Initialize self. See help(type(self)) for accurate signature.

|

| get\_next\_page(self)

| Returns the next page of results as a sequence of Album objects.

|

| ----------------------------------------------------------------------

| Methods inherited from \_Search:

|

| get\_total\_result\_count(self)

| Returns the total count of all the results.

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class Artist(\_Taggable)

| Artist(name, network, username=None, info=None)

|

| An artist.

|

| Method resolution order:

| Artist

| \_Taggable

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_eq\_\_(self, other)

| Return self==value.

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| \_\_init\_\_(self, name, network, username=None, info=None)

| Create an artist object.

| # Parameters:

| \* name str: The artist's name.

|

| \_\_ne\_\_(self, other)

| Return self!=value.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| \_\_str\_\_ = r(\*args)

|

| \_\_unicode\_\_(self)

|

| get\_bio(self, section, language=None)

| Returns a section of the bio.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_bio\_content(self, language=None)

| Returns the content of the artist's biography.

|

| get\_bio\_published\_date(self)

| Returns the date on which the artist's biography was published.

|

| get\_bio\_summary(self, language=None)

| Returns the summary of the artist's biography.

|

| get\_correction(self)

| Returns the corrected artist name.

|

| get\_listener\_count(self)

| Returns the number of listeners on the network.

|

| get\_mbid(self)

| Returns the MusicBrainz ID of this artist.

|

| get\_name(self, properly\_capitalized=False)

| Returns the name of the artist.

| If properly\_capitalized was asserted then the name would be downloaded

| overwriting the given one.

|

| get\_playcount(self)

| Returns the number of plays on the network.

|

| get\_similar(self, limit=None)

| Returns the similar artists on the network.

|

| get\_top\_albums(self, limit=None, cacheable=True, stream=False)

| Returns a list of the top albums.

|

| get\_top\_tracks(self, limit=None, cacheable=True, stream=False)

| Returns a list of the most played Tracks by this artist.

|

| get\_url(self, domain\_name=0)

| Returns the URL of the artist page on the network.

| # Parameters:

| \* domain\_name: The network's language domain. Possible values:

| o DOMAIN\_ENGLISH

| o DOMAIN\_GERMAN

| o DOMAIN\_SPANISH

| o DOMAIN\_FRENCH

| o DOMAIN\_ITALIAN

| o DOMAIN\_POLISH

| o DOMAIN\_PORTUGUESE

| o DOMAIN\_SWEDISH

| o DOMAIN\_TURKISH

| o DOMAIN\_RUSSIAN

| o DOMAIN\_JAPANESE

| o DOMAIN\_CHINESE

|

| get\_userplaycount(self)

| Returns the number of plays by a given username

|

| is\_streamable(self)

| Returns True if the artist is streamable.

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| name = None

|

| username = None

|

| ----------------------------------------------------------------------

| Methods inherited from \_Taggable:

|

| add\_tag(self, tag)

| Adds one tag.

| \* tag: a tag name or a Tag object.

|

| add\_tags(self, tags)

| Adds one or several tags.

| \* tags: A sequence of tag names or Tag objects.

|

| clear\_tags(self)

| Clears all the user-set tags.

|

| get\_tags(self)

| Returns a list of the tags set by the user to this object.

|

| get\_top\_tags(self, limit=None)

| Returns a list of the most frequently used Tags on this object.

|

| remove\_tag(self, tag)

| Remove a user's tag from this object.

|

| remove\_tags(self, tags)

| Removes one or several tags from this object.

| \* tags: a sequence of tag names or Tag objects.

|

| set\_tags(self, tags)

| Sets this object's tags to only those tags.

| \* tags: a sequence of tag names or Tag objects.

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class ArtistSearch(\_Search)

| ArtistSearch(artist\_name, network)

|

| Search for an artist by artist name.

|

| Method resolution order:

| ArtistSearch

| \_Search

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, artist\_name, network)

| Initialize self. See help(type(self)) for accurate signature.

|

| get\_next\_page(self)

| Returns the next page of results as a sequence of Artist objects.

|

| ----------------------------------------------------------------------

| Methods inherited from \_Search:

|

| get\_total\_result\_count(self)

| Returns the total count of all the results.

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class AuthenticatedUser(User)

| AuthenticatedUser(network)

|

| A Last.fm user.

|

| Method resolution order:

| AuthenticatedUser

| User

| \_Chartable

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, network)

| Initialize self. See help(type(self)) for accurate signature.

|

| get\_name(self, properly\_capitalized=False)

| Returns the name of the authenticated user.

|

| ----------------------------------------------------------------------

| Methods inherited from User:

|

| \_\_eq\_\_(self, other)

| Return self==value.

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| \_\_ne\_\_(self, other)

| Return self!=value.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| \_\_str\_\_ = r(\*args)

|

| get\_country(self)

| Returns the name of the country of the user.

|

| get\_friends(self, limit=50, cacheable=False, stream=False)

| Returns a list of the user's friends.

|

| get\_image(self, size=3)

| Returns the user's avatar

| size can be one of:

| SIZE\_EXTRA\_LARGE

| SIZE\_LARGE

| SIZE\_MEDIUM

| SIZE\_SMALL

|

| get\_library(self)

| Returns the associated Library object.

|

| get\_loved\_tracks(self, limit=50, cacheable=True, stream=False)

| Returns this user's loved track as a sequence of LovedTrack objects in

| reverse order of their timestamp, all the way back to the first track.

|

| If limit==None, it will try to pull all the available data.

| If stream=True, it will yield tracks as soon as a page has been retrieved.

|

| This method uses caching. Enable caching only if you're pulling a

| large amount of data.

|

| get\_now\_playing(self)

| Returns the currently playing track, or None if nothing is playing.

|

| get\_playcount(self)

| Returns the user's playcount so far.

|

| get\_recent\_tracks(self, limit=10, cacheable=True, time\_from=None, time\_to=None, stream=False, now\_playing=False)

| Returns this user's played track as a sequence of PlayedTrack objects

| in reverse order of playtime, all the way back to the first track.

|

| Parameters:

| limit : If None, it will try to pull all the available data.

| from (Optional) : Beginning timestamp of a range - only display

| scrobbles after this time, in UNIX timestamp format (integer

| number of seconds since 00:00:00, January 1st 1970 UTC). This

| must be in the UTC time zone.

| to (Optional) : End timestamp of a range - only display scrobbles

| before this time, in UNIX timestamp format (integer number of

| seconds since 00:00:00, January 1st 1970 UTC). This must be in

| the UTC time zone.

| stream: If True, it will yield tracks as soon as a page has been retrieved.

|

| This method uses caching. Enable caching only if you're pulling a

| large amount of data.

|

| get\_registered(self)

| Returns the user's registration date.

|

| get\_tagged\_albums(self, tag, limit=None, cacheable=True)

| Returns the albums tagged by a user.

|

| get\_tagged\_artists(self, tag, limit=None)

| Returns the artists tagged by a user.

|

| get\_tagged\_tracks(self, tag, limit=None, cacheable=True)

| Returns the tracks tagged by a user.

|

| get\_top\_albums(self, period='overall', limit=None, cacheable=True)

| Returns the top albums played by a user.

| \* period: The period of time. Possible values:

| o PERIOD\_OVERALL

| o PERIOD\_7DAYS

| o PERIOD\_1MONTH

| o PERIOD\_3MONTHS

| o PERIOD\_6MONTHS

| o PERIOD\_12MONTHS

|

| get\_top\_artists(self, period='overall', limit=None)

| Returns the top artists played by a user.

| \* period: The period of time. Possible values:

| o PERIOD\_OVERALL

| o PERIOD\_7DAYS

| o PERIOD\_1MONTH

| o PERIOD\_3MONTHS

| o PERIOD\_6MONTHS

| o PERIOD\_12MONTHS

|

| get\_top\_tags(self, limit=None, cacheable=True)

| Returns a sequence of the top tags used by this user with their counts

| as TopItem objects.

| \* limit: The limit of how many tags to return.

| \* cacheable: Whether to cache results.

|

| get\_top\_tracks(self, period='overall', limit=None, cacheable=True, stream=False)

| Returns the top tracks played by a user.

| \* period: The period of time. Possible values:

| o PERIOD\_OVERALL

| o PERIOD\_7DAYS

| o PERIOD\_1MONTH

| o PERIOD\_3MONTHS

| o PERIOD\_6MONTHS

| o PERIOD\_12MONTHS

|

| get\_track\_scrobbles(self, artist, track, cacheable=False, stream=False)

| Get a list of this user's scrobbles of this artist's track,

| including scrobble time.

|

| get\_unixtime\_registered(self)

| Returns the user's registration date as a UNIX timestamp.

|

| get\_url(self, domain\_name=0)

| Returns the URL of the user page on the network.

| \* domain\_name: The network's language domain. Possible values:

| o DOMAIN\_ENGLISH

| o DOMAIN\_GERMAN

| o DOMAIN\_SPANISH

| o DOMAIN\_FRENCH

| o DOMAIN\_ITALIAN

| o DOMAIN\_POLISH

| o DOMAIN\_PORTUGUESE

| o DOMAIN\_SWEDISH

| o DOMAIN\_TURKISH

| o DOMAIN\_RUSSIAN

| o DOMAIN\_JAPANESE

| o DOMAIN\_CHINESE

|

| is\_subscriber(self)

| Returns whether the user is a subscriber or not. True or False.

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from User:

|

| name = None

|

| ----------------------------------------------------------------------

| Methods inherited from \_Chartable:

|

| get\_weekly\_album\_charts(self, from\_date=None, to\_date=None)

| Returns the weekly album charts for the week starting from the

| from\_date value to the to\_date value.

| Only for User.

|

| get\_weekly\_artist\_charts(self, from\_date=None, to\_date=None)

| Returns the weekly artist charts for the week starting from the

| from\_date value to the to\_date value.

| Only for User.

|

| get\_weekly\_chart\_dates(self)

| Returns a list of From and To tuples for the available charts.

|

| get\_weekly\_charts(self, chart\_kind, from\_date=None, to\_date=None)

| Returns the weekly charts for the week starting from the

| from\_date value to the to\_date value.

| chart\_kind should be one of "album", "artist" or "track"

|

| get\_weekly\_track\_charts(self, from\_date=None, to\_date=None)

| Returns the weekly track charts for the week starting from the

| from\_date value to the to\_date value.

| Only for User.

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class Country(\_BaseObject)

| Country(name, network)

|

| A country at Last.fm.

|

| Method resolution order:

| Country

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_eq\_\_(self, other)

| Return self==value.

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| \_\_init\_\_(self, name, network)

| Initialize self. See help(type(self)) for accurate signature.

|

| \_\_ne\_\_(self, other)

| Return self!=value.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| \_\_str\_\_ = r(\*args)

|

| get\_name(self)

| Returns the country name.

|

| get\_top\_artists(self, limit=None, cacheable=True)

| Returns a sequence of the most played artists.

|

| get\_top\_tracks(self, limit=None, cacheable=True, stream=False)

| Returns a sequence of the most played tracks

|

| get\_url(self, domain\_name=0)

| Returns the URL of the country page on the network.

| \* domain\_name: The network's language domain. Possible values:

| o DOMAIN\_ENGLISH

| o DOMAIN\_GERMAN

| o DOMAIN\_SPANISH

| o DOMAIN\_FRENCH

| o DOMAIN\_ITALIAN

| o DOMAIN\_POLISH

| o DOMAIN\_PORTUGUESE

| o DOMAIN\_SWEDISH

| o DOMAIN\_TURKISH

| o DOMAIN\_RUSSIAN

| o DOMAIN\_JAPANESE

| o DOMAIN\_CHINESE

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| name = None

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class Image(builtins.tuple)

| Image(title, url, dateadded, format, owner, sizes, votes)

|

| Image(title, url, dateadded, format, owner, sizes, votes)

|

| Method resolution order:

| Image

| builtins.tuple

| builtins.object

|

| Methods defined here:

|

| \_\_getnewargs\_\_(self)

| Return self as a plain tuple. Used by copy and pickle.

|

| \_\_repr\_\_(self)

| Return a nicely formatted representation string

|

| \_asdict(self)

| Return a new dict which maps field names to their values.

|

| \_replace(self, /, \*\*kwds)

| Return a new Image object replacing specified fields with new values

|

| ----------------------------------------------------------------------

| Class methods defined here:

|

| \_make(iterable) from builtins.type

| Make a new Image object from a sequence or iterable

|

| ----------------------------------------------------------------------

| Static methods defined here:

|

| \_\_new\_\_(\_cls, title, url, dateadded, format, owner, sizes, votes)

| Create new instance of Image(title, url, dateadded, format, owner, sizes, votes)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| title

| Alias for field number 0

|

| url

| Alias for field number 1

|

| dateadded

| Alias for field number 2

|

| format

| Alias for field number 3

|

| owner

| Alias for field number 4

|

| sizes

| Alias for field number 5

|

| votes

| Alias for field number 6

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| \_field\_defaults = {}

|

| \_fields = ('title', 'url', 'dateadded', 'format', 'owner', 'sizes', 'v...

|

| \_fields\_defaults = {}

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.tuple:

|

| \_\_add\_\_(self, value, /)

| Return self+value.

|

| \_\_contains\_\_(self, key, /)

| Return key in self.

|

| \_\_eq\_\_(self, value, /)

| Return self==value.

|

| \_\_ge\_\_(self, value, /)

| Return self>=value.

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_getitem\_\_(self, key, /)

| Return self[key].

|

| \_\_gt\_\_(self, value, /)

| Return self>value.

|

| \_\_hash\_\_(self, /)

| Return hash(self).

|

| \_\_iter\_\_(self, /)

| Implement iter(self).

|

| \_\_le\_\_(self, value, /)

| Return self<=value.

|

| \_\_len\_\_(self, /)

| Return len(self).

|

| \_\_lt\_\_(self, value, /)

| Return self<value.

|

| \_\_mul\_\_(self, value, /)

| Return self\*value.

|

| \_\_ne\_\_(self, value, /)

| Return self!=value.

|

| \_\_rmul\_\_(self, value, /)

| Return value\*self.

|

| count(self, value, /)

| Return number of occurrences of value.

|

| index(self, value, start=0, stop=9223372036854775807, /)

| Return first index of value.

|

| Raises ValueError if the value is not present.

class ImageSizes(builtins.tuple)

| ImageSizes(original, large, largesquare, medium, small, extralarge)

|

| ImageSizes(original, large, largesquare, medium, small, extralarge)

|

| Method resolution order:

| ImageSizes

| builtins.tuple

| builtins.object

|

| Methods defined here:

|

| \_\_getnewargs\_\_(self)

| Return self as a plain tuple. Used by copy and pickle.

|

| \_\_repr\_\_(self)

| Return a nicely formatted representation string

|

| \_asdict(self)

| Return a new dict which maps field names to their values.

|

| \_replace(self, /, \*\*kwds)

| Return a new ImageSizes object replacing specified fields with new values

|

| ----------------------------------------------------------------------

| Class methods defined here:

|

| \_make(iterable) from builtins.type

| Make a new ImageSizes object from a sequence or iterable

|

| ----------------------------------------------------------------------

| Static methods defined here:

|

| \_\_new\_\_(\_cls, original, large, largesquare, medium, small, extralarge)

| Create new instance of ImageSizes(original, large, largesquare, medium, small, extralarge)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| original

| Alias for field number 0

|

| large

| Alias for field number 1

|

| largesquare

| Alias for field number 2

|

| medium

| Alias for field number 3

|

| small

| Alias for field number 4

|

| extralarge

| Alias for field number 5

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| \_field\_defaults = {}

|

| \_fields = ('original', 'large', 'largesquare', 'medium', 'small', 'ext...

|

| \_fields\_defaults = {}

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.tuple:

|

| \_\_add\_\_(self, value, /)

| Return self+value.

|

| \_\_contains\_\_(self, key, /)

| Return key in self.

|

| \_\_eq\_\_(self, value, /)

| Return self==value.

|

| \_\_ge\_\_(self, value, /)

| Return self>=value.

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_getitem\_\_(self, key, /)

| Return self[key].

|

| \_\_gt\_\_(self, value, /)

| Return self>value.

|

| \_\_hash\_\_(self, /)

| Return hash(self).

|

| \_\_iter\_\_(self, /)

| Implement iter(self).

|

| \_\_le\_\_(self, value, /)

| Return self<=value.

|

| \_\_len\_\_(self, /)

| Return len(self).

|

| \_\_lt\_\_(self, value, /)

| Return self<value.

|

| \_\_mul\_\_(self, value, /)

| Return self\*value.

|

| \_\_ne\_\_(self, value, /)

| Return self!=value.

|

| \_\_rmul\_\_(self, value, /)

| Return value\*self.

|

| count(self, value, /)

| Return number of occurrences of value.

|

| index(self, value, start=0, stop=9223372036854775807, /)

| Return first index of value.

|

| Raises ValueError if the value is not present.

class LastFMNetwork(\_Network)

| LastFMNetwork(api\_key='', api\_secret='', session\_key='', username='', password\_hash='', token='')

|

| A Last.fm network object

|

| api\_key: a provided API\_KEY

| api\_secret: a provided API\_SECRET

| session\_key: a generated session\_key or None

| username: a username of a valid user

| password\_hash: the output of pylast.md5(password) where password is the

| user's password

|

| if username and password\_hash were provided and not session\_key,

| session\_key will be generated automatically when needed.

|

| Either a valid session\_key or a combination of username and password\_hash

| must be present for scrobbling.

|

| Most read-only webservices only require an api\_key and an api\_secret, see

| about obtaining them from:

| <https://www.last.fm/api/account>

|

| Method resolution order:

| LastFMNetwork

| \_Network

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, api\_key='', api\_secret='', session\_key='', username='', password\_hash='', token='')

| name: the name of the network

| homepage: the homepage URL

| ws\_server: the URL of the webservices server

| api\_key: a provided API\_KEY

| api\_secret: a provided API\_SECRET

| session\_key: a generated session\_key or None

| username: a username of a valid user

| password\_hash: the output of pylast.md5(password) where password is

| the user's password

| domain\_names: a dict mapping each DOMAIN\_\* value to a string domain

| name

| urls: a dict mapping types to URLs

| token: an authentication token to retrieve a session

|

| if username and password\_hash were provided and not session\_key,

| session\_key will be generated automatically when needed.

|

| Either a valid session\_key or a combination of username and

| password\_hash must be present for scrobbling.

|

| You should use a preconfigured network object through a

| get\_\*\_network(...) method instead of creating an object

| of this class, unless you know what you're doing.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| ----------------------------------------------------------------------

| Methods inherited from \_Network:

|

| \_\_str\_\_(self)

| Return str(self).

|

| disable\_caching(self)

| Disables all caching features.

|

| disable\_proxy(self)

| Disable using the web proxy

|

| disable\_rate\_limit(self)

| Disables rate limiting for this network

|

| enable\_caching(self, file\_path=None)

| Enables caching request-wide for all cacheable calls.

|

| \* file\_path: A file path for the backend storage file. If

| None set, a temp file would probably be created, according the backend.

|

| enable\_proxy(self, host, port)

| Enable a default web proxy

|

| enable\_rate\_limit(self)

| Enables rate limiting for this network

|

| get\_album(self, artist, title)

| Return an Album object

|

| get\_album\_by\_mbid(self, mbid)

| Looks up an album by its MusicBrainz ID

|

| get\_artist(self, artist\_name)

| Return an Artist object

|

| get\_artist\_by\_mbid(self, mbid)

| Looks up an artist by its MusicBrainz ID

|

| get\_authenticated\_user(self)

| Returns the authenticated user

|

| get\_country(self, country\_name)

| Returns a country object

|

| get\_geo\_top\_artists(self, country, limit=None, cacheable=True)

| Get the most popular artists on Last.fm by country.

| Parameters:

| country (Required) : A country name, as defined by the ISO 3166-1

| country names standard.

| limit (Optional) : The number of results to fetch per page.

| Defaults to 50.

|

| get\_geo\_top\_tracks(self, country, location=None, limit=None, cacheable=True)

| Get the most popular tracks on Last.fm last week by country.

| Parameters:

| country (Required) : A country name, as defined by the ISO 3166-1

| country names standard

| location (Optional) : A metro name, to fetch the charts for

| (must be within the country specified)

| limit (Optional) : The number of results to fetch per page.

| Defaults to 50.

|

| get\_tag(self, name)

| Returns a tag object

|

| get\_top\_artists(self, limit=None, cacheable=True)

| Returns the most played artists as a sequence of TopItem objects.

|

| get\_top\_tags(self, limit=None, cacheable=True)

| Returns the most used tags as a sequence of TopItem objects.

|

| get\_top\_tracks(self, limit=None, cacheable=True)

| Returns the most played tracks as a sequence of TopItem objects.

|

| get\_track(self, artist, title)

| Return a Track object

|

| get\_track\_by\_mbid(self, mbid)

| Looks up a track by its MusicBrainz ID

|

| get\_user(self, username)

| Returns a user object

|

| is\_caching\_enabled(self)

| Returns True if caching is enabled.

|

| is\_proxy\_enabled(self)

| Returns True if a web proxy is enabled.

|

| is\_rate\_limited(self)

| Return True if web service calls are rate limited

|

| scrobble(self, artist, title, timestamp, album=None, album\_artist=None, track\_number=None, duration=None, stream\_id=None, context=None, mbid=None)

| Used to add a track-play to a user's profile.

|

| Parameters:

| artist (Required) : The artist name.

| title (Required) : The track name.

| timestamp (Required) : The time the track started playing, in UNIX

| timestamp format (integer number of seconds since 00:00:00,

| January 1st 1970 UTC). This must be in the UTC time zone.

| album (Optional) : The album name.

| album\_artist (Optional) : The album artist - if this differs from

| the track artist.

| context (Optional) : Sub-client version (not public, only enabled

| for certain API keys)

| stream\_id (Optional) : The stream id for this track received from

| the radio.getPlaylist service.

| track\_number (Optional) : The track number of the track on the

| album.

| mbid (Optional) : The MusicBrainz Track ID.

| duration (Optional) : The length of the track in seconds.

|

| scrobble\_many(self, tracks)

| Used to scrobble a batch of tracks at once. The parameter tracks is a

| sequence of dicts per track containing the keyword arguments as if

| passed to the scrobble() method.

|

| search\_for\_album(self, album\_name)

| Searches for an album by its name. Returns a AlbumSearch object.

| Use get\_next\_page() to retrieve sequences of results.

|

| search\_for\_artist(self, artist\_name)

| Searches of an artist by its name. Returns a ArtistSearch object.

| Use get\_next\_page() to retrieve sequences of results.

|

| search\_for\_track(self, artist\_name, track\_name)

| Searches of a track by its name and its artist. Set artist to an

| empty string if not available.

| Returns a TrackSearch object.

| Use get\_next\_page() to retrieve sequences of results.

|

| update\_now\_playing(self, artist, title, album=None, album\_artist=None, duration=None, track\_number=None, mbid=None, context=None)

| Used to notify Last.fm that a user has started listening to a track.

|

| Parameters:

| artist (Required) : The artist name

| title (Required) : The track title

| album (Optional) : The album name.

| album\_artist (Optional) : The album artist - if this differs

| from the track artist.

| duration (Optional) : The length of the track in seconds.

| track\_number (Optional) : The track number of the track on the

| album.

| mbid (Optional) : The MusicBrainz Track ID.

| context (Optional) : Sub-client version

| (not public, only enabled for certain API keys)

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_Network:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

class Library(\_BaseObject)

| Library(user, network)

|

| A user's Last.fm library.

|

| Method resolution order:

| Library

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| \_\_init\_\_(self, user, network)

| Initialize self. See help(type(self)) for accurate signature.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| \_\_str\_\_ = r(\*args)

|

| get\_artists(self, limit=50, cacheable=True, stream=False)

| Returns a sequence of Album objects

| if limit==None it will return all (may take a while)

|

| get\_user(self)

| Returns the user who owns this library.

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| user = None

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class LibraryItem(builtins.tuple)

| LibraryItem(item, playcount, tagcount)

|

| LibraryItem(item, playcount, tagcount)

|

| Method resolution order:

| LibraryItem

| builtins.tuple

| builtins.object

|

| Methods defined here:

|

| \_\_getnewargs\_\_(self)

| Return self as a plain tuple. Used by copy and pickle.

|

| \_\_repr\_\_(self)

| Return a nicely formatted representation string

|

| \_asdict(self)

| Return a new dict which maps field names to their values.

|

| \_replace(self, /, \*\*kwds)

| Return a new LibraryItem object replacing specified fields with new values

|

| ----------------------------------------------------------------------

| Class methods defined here:

|

| \_make(iterable) from builtins.type

| Make a new LibraryItem object from a sequence or iterable

|

| ----------------------------------------------------------------------

| Static methods defined here:

|

| \_\_new\_\_(\_cls, item, playcount, tagcount)

| Create new instance of LibraryItem(item, playcount, tagcount)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| item

| Alias for field number 0

|

| playcount

| Alias for field number 1

|

| tagcount

| Alias for field number 2

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| \_field\_defaults = {}

|

| \_fields = ('item', 'playcount', 'tagcount')

|

| \_fields\_defaults = {}

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.tuple:

|

| \_\_add\_\_(self, value, /)

| Return self+value.

|

| \_\_contains\_\_(self, key, /)

| Return key in self.

|

| \_\_eq\_\_(self, value, /)

| Return self==value.

|

| \_\_ge\_\_(self, value, /)

| Return self>=value.

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_getitem\_\_(self, key, /)

| Return self[key].

|

| \_\_gt\_\_(self, value, /)

| Return self>value.

|

| \_\_hash\_\_(self, /)

| Return hash(self).

|

| \_\_iter\_\_(self, /)

| Implement iter(self).

|

| \_\_le\_\_(self, value, /)

| Return self<=value.

|

| \_\_len\_\_(self, /)

| Return len(self).

|

| \_\_lt\_\_(self, value, /)

| Return self<value.

|

| \_\_mul\_\_(self, value, /)

| Return self\*value.

|

| \_\_ne\_\_(self, value, /)

| Return self!=value.

|

| \_\_rmul\_\_(self, value, /)

| Return value\*self.

|

| count(self, value, /)

| Return number of occurrences of value.

|

| index(self, value, start=0, stop=9223372036854775807, /)

| Return first index of value.

|

| Raises ValueError if the value is not present.

class LibreFMNetwork(\_Network)

| LibreFMNetwork(api\_key='', api\_secret='', session\_key='', username='', password\_hash='')

|

| A preconfigured \_Network object for Libre.fm

|

| api\_key: a provided API\_KEY

| api\_secret: a provided API\_SECRET

| session\_key: a generated session\_key or None

| username: a username of a valid user

| password\_hash: the output of pylast.md5(password) where password is the

| user's password

|

| if username and password\_hash were provided and not session\_key,

| session\_key will be generated automatically when needed.

|

| Method resolution order:

| LibreFMNetwork

| \_Network

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, api\_key='', api\_secret='', session\_key='', username='', password\_hash='')

| name: the name of the network

| homepage: the homepage URL

| ws\_server: the URL of the webservices server

| api\_key: a provided API\_KEY

| api\_secret: a provided API\_SECRET

| session\_key: a generated session\_key or None

| username: a username of a valid user

| password\_hash: the output of pylast.md5(password) where password is

| the user's password

| domain\_names: a dict mapping each DOMAIN\_\* value to a string domain

| name

| urls: a dict mapping types to URLs

| token: an authentication token to retrieve a session

|

| if username and password\_hash were provided and not session\_key,

| session\_key will be generated automatically when needed.

|

| Either a valid session\_key or a combination of username and

| password\_hash must be present for scrobbling.

|

| You should use a preconfigured network object through a

| get\_\*\_network(...) method instead of creating an object

| of this class, unless you know what you're doing.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| ----------------------------------------------------------------------

| Methods inherited from \_Network:

|

| \_\_str\_\_(self)

| Return str(self).

|

| disable\_caching(self)

| Disables all caching features.

|

| disable\_proxy(self)

| Disable using the web proxy

|

| disable\_rate\_limit(self)

| Disables rate limiting for this network

|

| enable\_caching(self, file\_path=None)

| Enables caching request-wide for all cacheable calls.

|

| \* file\_path: A file path for the backend storage file. If

| None set, a temp file would probably be created, according the backend.

|

| enable\_proxy(self, host, port)

| Enable a default web proxy

|

| enable\_rate\_limit(self)

| Enables rate limiting for this network

|

| get\_album(self, artist, title)

| Return an Album object

|

| get\_album\_by\_mbid(self, mbid)

| Looks up an album by its MusicBrainz ID

|

| get\_artist(self, artist\_name)

| Return an Artist object

|

| get\_artist\_by\_mbid(self, mbid)

| Looks up an artist by its MusicBrainz ID

|

| get\_authenticated\_user(self)

| Returns the authenticated user

|

| get\_country(self, country\_name)

| Returns a country object

|

| get\_geo\_top\_artists(self, country, limit=None, cacheable=True)

| Get the most popular artists on Last.fm by country.

| Parameters:

| country (Required) : A country name, as defined by the ISO 3166-1

| country names standard.

| limit (Optional) : The number of results to fetch per page.

| Defaults to 50.

|

| get\_geo\_top\_tracks(self, country, location=None, limit=None, cacheable=True)

| Get the most popular tracks on Last.fm last week by country.

| Parameters:

| country (Required) : A country name, as defined by the ISO 3166-1

| country names standard

| location (Optional) : A metro name, to fetch the charts for

| (must be within the country specified)

| limit (Optional) : The number of results to fetch per page.

| Defaults to 50.

|

| get\_tag(self, name)

| Returns a tag object

|

| get\_top\_artists(self, limit=None, cacheable=True)

| Returns the most played artists as a sequence of TopItem objects.

|

| get\_top\_tags(self, limit=None, cacheable=True)

| Returns the most used tags as a sequence of TopItem objects.

|

| get\_top\_tracks(self, limit=None, cacheable=True)

| Returns the most played tracks as a sequence of TopItem objects.

|

| get\_track(self, artist, title)

| Return a Track object

|

| get\_track\_by\_mbid(self, mbid)

| Looks up a track by its MusicBrainz ID

|

| get\_user(self, username)

| Returns a user object

|

| is\_caching\_enabled(self)

| Returns True if caching is enabled.

|

| is\_proxy\_enabled(self)

| Returns True if a web proxy is enabled.

|

| is\_rate\_limited(self)

| Return True if web service calls are rate limited

|

| scrobble(self, artist, title, timestamp, album=None, album\_artist=None, track\_number=None, duration=None, stream\_id=None, context=None, mbid=None)

| Used to add a track-play to a user's profile.

|

| Parameters:

| artist (Required) : The artist name.

| title (Required) : The track name.

| timestamp (Required) : The time the track started playing, in UNIX

| timestamp format (integer number of seconds since 00:00:00,

| January 1st 1970 UTC). This must be in the UTC time zone.

| album (Optional) : The album name.

| album\_artist (Optional) : The album artist - if this differs from

| the track artist.

| context (Optional) : Sub-client version (not public, only enabled

| for certain API keys)

| stream\_id (Optional) : The stream id for this track received from

| the radio.getPlaylist service.

| track\_number (Optional) : The track number of the track on the

| album.

| mbid (Optional) : The MusicBrainz Track ID.

| duration (Optional) : The length of the track in seconds.

|

| scrobble\_many(self, tracks)

| Used to scrobble a batch of tracks at once. The parameter tracks is a

| sequence of dicts per track containing the keyword arguments as if

| passed to the scrobble() method.

|

| search\_for\_album(self, album\_name)

| Searches for an album by its name. Returns a AlbumSearch object.

| Use get\_next\_page() to retrieve sequences of results.

|

| search\_for\_artist(self, artist\_name)

| Searches of an artist by its name. Returns a ArtistSearch object.

| Use get\_next\_page() to retrieve sequences of results.

|

| search\_for\_track(self, artist\_name, track\_name)

| Searches of a track by its name and its artist. Set artist to an

| empty string if not available.

| Returns a TrackSearch object.

| Use get\_next\_page() to retrieve sequences of results.

|

| update\_now\_playing(self, artist, title, album=None, album\_artist=None, duration=None, track\_number=None, mbid=None, context=None)

| Used to notify Last.fm that a user has started listening to a track.

|

| Parameters:

| artist (Required) : The artist name

| title (Required) : The track title

| album (Optional) : The album name.

| album\_artist (Optional) : The album artist - if this differs

| from the track artist.

| duration (Optional) : The length of the track in seconds.

| track\_number (Optional) : The track number of the track on the

| album.

| mbid (Optional) : The MusicBrainz Track ID.

| context (Optional) : Sub-client version

| (not public, only enabled for certain API keys)

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_Network:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

class LovedTrack(builtins.tuple)

| LovedTrack(track, date, timestamp)

|

| LovedTrack(track, date, timestamp)

|

| Method resolution order:

| LovedTrack

| builtins.tuple

| builtins.object

|

| Methods defined here:

|

| \_\_getnewargs\_\_(self)

| Return self as a plain tuple. Used by copy and pickle.

|

| \_\_repr\_\_(self)

| Return a nicely formatted representation string

|

| \_asdict(self)

| Return a new dict which maps field names to their values.

|

| \_replace(self, /, \*\*kwds)

| Return a new LovedTrack object replacing specified fields with new values

|

| ----------------------------------------------------------------------

| Class methods defined here:

|

| \_make(iterable) from builtins.type

| Make a new LovedTrack object from a sequence or iterable

|

| ----------------------------------------------------------------------

| Static methods defined here:

|

| \_\_new\_\_(\_cls, track, date, timestamp)

| Create new instance of LovedTrack(track, date, timestamp)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| track

| Alias for field number 0

|

| date

| Alias for field number 1

|

| timestamp

| Alias for field number 2

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| \_field\_defaults = {}

|

| \_fields = ('track', 'date', 'timestamp')

|

| \_fields\_defaults = {}

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.tuple:

|

| \_\_add\_\_(self, value, /)

| Return self+value.

|

| \_\_contains\_\_(self, key, /)

| Return key in self.

|

| \_\_eq\_\_(self, value, /)

| Return self==value.

|

| \_\_ge\_\_(self, value, /)

| Return self>=value.

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_getitem\_\_(self, key, /)

| Return self[key].

|

| \_\_gt\_\_(self, value, /)

| Return self>value.

|

| \_\_hash\_\_(self, /)

| Return hash(self).

|

| \_\_iter\_\_(self, /)

| Implement iter(self).

|

| \_\_le\_\_(self, value, /)

| Return self<=value.

|

| \_\_len\_\_(self, /)

| Return len(self).

|

| \_\_lt\_\_(self, value, /)

| Return self<value.

|

| \_\_mul\_\_(self, value, /)

| Return self\*value.

|

| \_\_ne\_\_(self, value, /)

| Return self!=value.

|

| \_\_rmul\_\_(self, value, /)

| Return value\*self.

|

| count(self, value, /)

| Return number of occurrences of value.

|

| index(self, value, start=0, stop=9223372036854775807, /)

| Return first index of value.

|

| Raises ValueError if the value is not present.

class MalformedResponseError(PyLastError)

| MalformedResponseError(network, underlying\_error)

|

| Exception conveying a malformed response from the music network.

|

| Method resolution order:

| MalformedResponseError

| PyLastError

| builtins.Exception

| builtins.BaseException

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, network, underlying\_error)

| Initialize self. See help(type(self)) for accurate signature.

|

| \_\_str\_\_(self)

| Return str(self).

|

| ----------------------------------------------------------------------

| Data descriptors inherited from PyLastError:

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Static methods inherited from builtins.Exception:

|

| \_\_new\_\_(\*args, \*\*kwargs) from builtins.type

| Create and return a new object. See help(type) for accurate signature.

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.BaseException:

|

| \_\_delattr\_\_(self, name, /)

| Implement delattr(self, name).

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_reduce\_\_(...)

| Helper for pickle.

|

| \_\_repr\_\_(self, /)

| Return repr(self).

|

| \_\_setattr\_\_(self, name, value, /)

| Implement setattr(self, name, value).

|

| \_\_setstate\_\_(...)

|

| with\_traceback(...)

| Exception.with\_traceback(tb) --

| set self.\_\_traceback\_\_ to tb and return self.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from builtins.BaseException:

|

| \_\_cause\_\_

| exception cause

|

| \_\_context\_\_

| exception context

|

| \_\_dict\_\_

|

| \_\_suppress\_context\_\_

|

| \_\_traceback\_\_

|

| args

class NetworkError(PyLastError)

| NetworkError(network, underlying\_error)

|

| Exception conveying a problem in sending a request to Last.fm

|

| Method resolution order:

| NetworkError

| PyLastError

| builtins.Exception

| builtins.BaseException

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, network, underlying\_error)

| Initialize self. See help(type(self)) for accurate signature.

|

| \_\_str\_\_(self)

| Return str(self).

|

| ----------------------------------------------------------------------

| Data descriptors inherited from PyLastError:

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Static methods inherited from builtins.Exception:

|

| \_\_new\_\_(\*args, \*\*kwargs) from builtins.type

| Create and return a new object. See help(type) for accurate signature.

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.BaseException:

|

| \_\_delattr\_\_(self, name, /)

| Implement delattr(self, name).

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_reduce\_\_(...)

| Helper for pickle.

|

| \_\_repr\_\_(self, /)

| Return repr(self).

|

| \_\_setattr\_\_(self, name, value, /)

| Implement setattr(self, name, value).

|

| \_\_setstate\_\_(...)

|

| with\_traceback(...)

| Exception.with\_traceback(tb) --

| set self.\_\_traceback\_\_ to tb and return self.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from builtins.BaseException:

|

| \_\_cause\_\_

| exception cause

|

| \_\_context\_\_

| exception context

|

| \_\_dict\_\_

|

| \_\_suppress\_context\_\_

|

| \_\_traceback\_\_

|

| args

class PlayedTrack(builtins.tuple)

| PlayedTrack(track, album, playback\_date, timestamp)

|

| PlayedTrack(track, album, playback\_date, timestamp)

|

| Method resolution order:

| PlayedTrack

| builtins.tuple

| builtins.object

|

| Methods defined here:

|

| \_\_getnewargs\_\_(self)

| Return self as a plain tuple. Used by copy and pickle.

|

| \_\_repr\_\_(self)

| Return a nicely formatted representation string

|

| \_asdict(self)

| Return a new dict which maps field names to their values.

|

| \_replace(self, /, \*\*kwds)

| Return a new PlayedTrack object replacing specified fields with new values

|

| ----------------------------------------------------------------------

| Class methods defined here:

|

| \_make(iterable) from builtins.type

| Make a new PlayedTrack object from a sequence or iterable

|

| ----------------------------------------------------------------------

| Static methods defined here:

|

| \_\_new\_\_(\_cls, track, album, playback\_date, timestamp)

| Create new instance of PlayedTrack(track, album, playback\_date, timestamp)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| track

| Alias for field number 0

|

| album

| Alias for field number 1

|

| playback\_date

| Alias for field number 2

|

| timestamp

| Alias for field number 3

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| \_field\_defaults = {}

|

| \_fields = ('track', 'album', 'playback\_date', 'timestamp')

|

| \_fields\_defaults = {}

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.tuple:

|

| \_\_add\_\_(self, value, /)

| Return self+value.

|

| \_\_contains\_\_(self, key, /)

| Return key in self.

|

| \_\_eq\_\_(self, value, /)

| Return self==value.

|

| \_\_ge\_\_(self, value, /)

| Return self>=value.

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_getitem\_\_(self, key, /)

| Return self[key].

|

| \_\_gt\_\_(self, value, /)

| Return self>value.

|

| \_\_hash\_\_(self, /)

| Return hash(self).

|

| \_\_iter\_\_(self, /)

| Implement iter(self).

|

| \_\_le\_\_(self, value, /)

| Return self<=value.

|

| \_\_len\_\_(self, /)

| Return len(self).

|

| \_\_lt\_\_(self, value, /)

| Return self<value.

|

| \_\_mul\_\_(self, value, /)

| Return self\*value.

|

| \_\_ne\_\_(self, value, /)

| Return self!=value.

|

| \_\_rmul\_\_(self, value, /)

| Return value\*self.

|

| count(self, value, /)

| Return number of occurrences of value.

|

| index(self, value, start=0, stop=9223372036854775807, /)

| Return first index of value.

|

| Raises ValueError if the value is not present.

class PyLastError(builtins.Exception)

| Generic exception raised by PyLast

|

| Method resolution order:

| PyLastError

| builtins.Exception

| builtins.BaseException

| builtins.object

|

| Data descriptors defined here:

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.Exception:

|

| \_\_init\_\_(self, /, \*args, \*\*kwargs)

| Initialize self. See help(type(self)) for accurate signature.

|

| ----------------------------------------------------------------------

| Static methods inherited from builtins.Exception:

|

| \_\_new\_\_(\*args, \*\*kwargs) from builtins.type

| Create and return a new object. See help(type) for accurate signature.

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.BaseException:

|

| \_\_delattr\_\_(self, name, /)

| Implement delattr(self, name).

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_reduce\_\_(...)

| Helper for pickle.

|

| \_\_repr\_\_(self, /)

| Return repr(self).

|

| \_\_setattr\_\_(self, name, value, /)

| Implement setattr(self, name, value).

|

| \_\_setstate\_\_(...)

|

| \_\_str\_\_(self, /)

| Return str(self).

|

| with\_traceback(...)

| Exception.with\_traceback(tb) --

| set self.\_\_traceback\_\_ to tb and return self.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from builtins.BaseException:

|

| \_\_cause\_\_

| exception cause

|

| \_\_context\_\_

| exception context

|

| \_\_dict\_\_

|

| \_\_suppress\_context\_\_

|

| \_\_traceback\_\_

|

| args

class SessionKeyGenerator(builtins.object)

| SessionKeyGenerator(network)

|

| Methods of generating a session key:

| 1) Web Authentication:

| a. network = get\_\*\_network(API\_KEY, API\_SECRET)

| b. sg = SessionKeyGenerator(network)

| c. url = sg.get\_web\_auth\_url()

| d. Ask the user to open the URL and authorize you, and wait for it.

| e. session\_key = sg.get\_web\_auth\_session\_key(url)

| 2) Username and Password Authentication:

| a. network = get\_\*\_network(API\_KEY, API\_SECRET)

| b. username = raw\_input("Please enter your username: ")

| c. password\_hash = pylast.md5(raw\_input("Please enter your password: ")

| d. session\_key = SessionKeyGenerator(network).get\_session\_key(username,

| password\_hash)

|

| A session key's lifetime is infinite, unless the user revokes the rights

| of the given API Key.

|

| If you create a Network object with just a API\_KEY and API\_SECRET and a

| username and a password\_hash, a SESSION\_KEY will be automatically generated

| for that network and stored in it so you don't have to do this manually,

| unless you want to.

|

| Methods defined here:

|

| \_\_init\_\_(self, network)

| Initialize self. See help(type(self)) for accurate signature.

|

| get\_session\_key(self, username, password\_hash)

| Retrieve a session key with a username and a md5 hash of the user's

| password.

|

| get\_web\_auth\_session\_key(self, url, token='')

| Retrieves the session key of a web authorization process by its URL.

|

| get\_web\_auth\_session\_key\_username(self, url, token='')

| Retrieves the session key/username of a web authorization process by its URL.

|

| get\_web\_auth\_url(self)

| The user must open this page, and you first, then

| call get\_web\_auth\_session\_key(url) after that.

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

class SimilarItem(builtins.tuple)

| SimilarItem(item, match)

|

| SimilarItem(item, match)

|

| Method resolution order:

| SimilarItem

| builtins.tuple

| builtins.object

|

| Methods defined here:

|

| \_\_getnewargs\_\_(self)

| Return self as a plain tuple. Used by copy and pickle.

|

| \_\_repr\_\_(self)

| Return a nicely formatted representation string

|

| \_asdict(self)

| Return a new dict which maps field names to their values.

|

| \_replace(self, /, \*\*kwds)

| Return a new SimilarItem object replacing specified fields with new values

|

| ----------------------------------------------------------------------

| Class methods defined here:

|

| \_make(iterable) from builtins.type

| Make a new SimilarItem object from a sequence or iterable

|

| ----------------------------------------------------------------------

| Static methods defined here:

|

| \_\_new\_\_(\_cls, item, match)

| Create new instance of SimilarItem(item, match)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| item

| Alias for field number 0

|

| match

| Alias for field number 1

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| \_field\_defaults = {}

|

| \_fields = ('item', 'match')

|

| \_fields\_defaults = {}

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.tuple:

|

| \_\_add\_\_(self, value, /)

| Return self+value.

|

| \_\_contains\_\_(self, key, /)

| Return key in self.

|

| \_\_eq\_\_(self, value, /)

| Return self==value.

|

| \_\_ge\_\_(self, value, /)

| Return self>=value.

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_getitem\_\_(self, key, /)

| Return self[key].

|

| \_\_gt\_\_(self, value, /)

| Return self>value.

|

| \_\_hash\_\_(self, /)

| Return hash(self).

|

| \_\_iter\_\_(self, /)

| Implement iter(self).

|

| \_\_le\_\_(self, value, /)

| Return self<=value.

|

| \_\_len\_\_(self, /)

| Return len(self).

|

| \_\_lt\_\_(self, value, /)

| Return self<value.

|

| \_\_mul\_\_(self, value, /)

| Return self\*value.

|

| \_\_ne\_\_(self, value, /)

| Return self!=value.

|

| \_\_rmul\_\_(self, value, /)

| Return value\*self.

|

| count(self, value, /)

| Return number of occurrences of value.

|

| index(self, value, start=0, stop=9223372036854775807, /)

| Return first index of value.

|

| Raises ValueError if the value is not present.

class Tag(\_Chartable)

| Tag(name, network)

|

| A Last.fm object tag.

|

| Method resolution order:

| Tag

| \_Chartable

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_eq\_\_(self, other)

| Return self==value.

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| \_\_init\_\_(self, name, network)

| Initialize self. See help(type(self)) for accurate signature.

|

| \_\_ne\_\_(self, other)

| Return self!=value.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| \_\_str\_\_ = r(\*args)

|

| get\_name(self, properly\_capitalized=False)

| Returns the name of the tag.

|

| get\_top\_albums(self, limit=None, cacheable=True)

| Returns a list of the top albums.

|

| get\_top\_artists(self, limit=None, cacheable=True)

| Returns a sequence of the most played artists.

|

| get\_top\_tracks(self, limit=None, cacheable=True, stream=False)

| Returns a list of the most played Tracks for this tag.

|

| get\_url(self, domain\_name=0)

| Returns the URL of the tag page on the network.

| \* domain\_name: The network's language domain. Possible values:

| o DOMAIN\_ENGLISH

| o DOMAIN\_GERMAN

| o DOMAIN\_SPANISH

| o DOMAIN\_FRENCH

| o DOMAIN\_ITALIAN

| o DOMAIN\_POLISH

| o DOMAIN\_PORTUGUESE

| o DOMAIN\_SWEDISH

| o DOMAIN\_TURKISH

| o DOMAIN\_RUSSIAN

| o DOMAIN\_JAPANESE

| o DOMAIN\_CHINESE

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| name = None

|

| ----------------------------------------------------------------------

| Methods inherited from \_Chartable:

|

| get\_weekly\_album\_charts(self, from\_date=None, to\_date=None)

| Returns the weekly album charts for the week starting from the

| from\_date value to the to\_date value.

| Only for User.

|

| get\_weekly\_artist\_charts(self, from\_date=None, to\_date=None)

| Returns the weekly artist charts for the week starting from the

| from\_date value to the to\_date value.

| Only for User.

|

| get\_weekly\_chart\_dates(self)

| Returns a list of From and To tuples for the available charts.

|

| get\_weekly\_charts(self, chart\_kind, from\_date=None, to\_date=None)

| Returns the weekly charts for the week starting from the

| from\_date value to the to\_date value.

| chart\_kind should be one of "album", "artist" or "track"

|

| get\_weekly\_track\_charts(self, from\_date=None, to\_date=None)

| Returns the weekly track charts for the week starting from the

| from\_date value to the to\_date value.

| Only for User.

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class TopItem(builtins.tuple)

| TopItem(item, weight)

|

| TopItem(item, weight)

|

| Method resolution order:

| TopItem

| builtins.tuple

| builtins.object

|

| Methods defined here:

|

| \_\_getnewargs\_\_(self)

| Return self as a plain tuple. Used by copy and pickle.

|

| \_\_repr\_\_(self)

| Return a nicely formatted representation string

|

| \_asdict(self)

| Return a new dict which maps field names to their values.

|

| \_replace(self, /, \*\*kwds)

| Return a new TopItem object replacing specified fields with new values

|

| ----------------------------------------------------------------------

| Class methods defined here:

|

| \_make(iterable) from builtins.type

| Make a new TopItem object from a sequence or iterable

|

| ----------------------------------------------------------------------

| Static methods defined here:

|

| \_\_new\_\_(\_cls, item, weight)

| Create new instance of TopItem(item, weight)

|

| ----------------------------------------------------------------------

| Data descriptors defined here:

|

| item

| Alias for field number 0

|

| weight

| Alias for field number 1

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| \_field\_defaults = {}

|

| \_fields = ('item', 'weight')

|

| \_fields\_defaults = {}

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.tuple:

|

| \_\_add\_\_(self, value, /)

| Return self+value.

|

| \_\_contains\_\_(self, key, /)

| Return key in self.

|

| \_\_eq\_\_(self, value, /)

| Return self==value.

|

| \_\_ge\_\_(self, value, /)

| Return self>=value.

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_getitem\_\_(self, key, /)

| Return self[key].

|

| \_\_gt\_\_(self, value, /)

| Return self>value.

|

| \_\_hash\_\_(self, /)

| Return hash(self).

|

| \_\_iter\_\_(self, /)

| Implement iter(self).

|

| \_\_le\_\_(self, value, /)

| Return self<=value.

|

| \_\_len\_\_(self, /)

| Return len(self).

|

| \_\_lt\_\_(self, value, /)

| Return self<value.

|

| \_\_mul\_\_(self, value, /)

| Return self\*value.

|

| \_\_ne\_\_(self, value, /)

| Return self!=value.

|

| \_\_rmul\_\_(self, value, /)

| Return value\*self.

|

| count(self, value, /)

| Return number of occurrences of value.

|

| index(self, value, start=0, stop=9223372036854775807, /)

| Return first index of value.

|

| Raises ValueError if the value is not present.

class Track(\_Opus)

| Track(artist, title, network, username=None, info=None)

|

| A Last.fm track.

|

| Method resolution order:

| Track

| \_Opus

| \_Taggable

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| \_\_init\_\_(self, artist, title, network, username=None, info=None)

| Create an opus instance.

| # Parameters:

| \* artist: An artist name or an Artist object.

| \* title: The album or track title.

| \* ws\_prefix: 'album' or 'track'

|

| get\_album(self)

| Returns the album object of this track.

|

| get\_correction(self)

| Returns the corrected track name.

|

| get\_duration(self)

| Returns the track duration.

|

| get\_similar(self, limit=None)

| Returns similar tracks for this track on the network,

| based on listening data.

|

| get\_url(self, domain\_name=0)

| Returns the URL of the album or track page on the network.

| # Parameters:

| \* domain\_name str: The network's language domain. Possible values:

| o DOMAIN\_ENGLISH

| o DOMAIN\_GERMAN

| o DOMAIN\_SPANISH

| o DOMAIN\_FRENCH

| o DOMAIN\_ITALIAN

| o DOMAIN\_POLISH

| o DOMAIN\_PORTUGUESE

| o DOMAIN\_SWEDISH

| o DOMAIN\_TURKISH

| o DOMAIN\_RUSSIAN

| o DOMAIN\_JAPANESE

| o DOMAIN\_CHINESE

|

| get\_userloved(self)

| Whether the user loved this track

|

| is\_fulltrack\_available(self)

| Returns True if the full track is available for streaming.

|

| is\_streamable(self)

| Returns True if the track is available at Last.fm.

|

| love(self)

| Adds the track to the user's loved tracks.

|

| unlove(self)

| Remove the track to the user's loved tracks.

|

| ----------------------------------------------------------------------

| Methods inherited from \_Opus:

|

| \_\_eq\_\_(self, other)

| Return self==value.

|

| \_\_ne\_\_(self, other)

| Return self!=value.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| \_\_str\_\_ = r(\*args)

|

| get\_artist(self)

| Returns the associated Artist object.

|

| get\_cover\_image(self, size=3)

| Returns a URI to the cover image

| size can be one of:

| SIZE\_EXTRA\_LARGE

| SIZE\_LARGE

| SIZE\_MEDIUM

| SIZE\_SMALL

|

| get\_listener\_count(self)

| Returns the number of listeners on the network

|

| get\_mbid(self)

| Returns the MusicBrainz ID of the album or track.

|

| get\_name(self, properly\_capitalized=False)

| Returns the album or track title (alias to get\_title()).

|

| get\_playcount(self)

| Returns the number of plays on the network

|

| get\_title(self, properly\_capitalized=False)

| Returns the artist or track title.

|

| get\_userplaycount(self)

| Returns the number of plays by a given username

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_Opus:

|

| artist = None

|

| title = None

|

| username = None

|

| ----------------------------------------------------------------------

| Methods inherited from \_Taggable:

|

| add\_tag(self, tag)

| Adds one tag.

| \* tag: a tag name or a Tag object.

|

| add\_tags(self, tags)

| Adds one or several tags.

| \* tags: A sequence of tag names or Tag objects.

|

| clear\_tags(self)

| Clears all the user-set tags.

|

| get\_tags(self)

| Returns a list of the tags set by the user to this object.

|

| get\_top\_tags(self, limit=None)

| Returns a list of the most frequently used Tags on this object.

|

| remove\_tag(self, tag)

| Remove a user's tag from this object.

|

| remove\_tags(self, tags)

| Removes one or several tags from this object.

| \* tags: a sequence of tag names or Tag objects.

|

| set\_tags(self, tags)

| Sets this object's tags to only those tags.

| \* tags: a sequence of tag names or Tag objects.

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class TrackSearch(\_Search)

| TrackSearch(artist\_name, track\_title, network)

|

| Search for a track by track title. If you don't want to narrow the results

| down by specifying the artist name, set it to empty string.

|

| Method resolution order:

| TrackSearch

| \_Search

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, artist\_name, track\_title, network)

| Initialize self. See help(type(self)) for accurate signature.

|

| get\_next\_page(self)

| Returns the next page of results as a sequence of Track objects.

|

| ----------------------------------------------------------------------

| Methods inherited from \_Search:

|

| get\_total\_result\_count(self)

| Returns the total count of all the results.

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class User(\_Chartable)

| User(user\_name, network)

|

| A Last.fm user.

|

| Method resolution order:

| User

| \_Chartable

| \_BaseObject

| builtins.object

|

| Methods defined here:

|

| \_\_eq\_\_(self, other)

| Return self==value.

|

| \_\_hash\_\_(self)

| Return hash(self).

|

| \_\_init\_\_(self, user\_name, network)

| Initialize self. See help(type(self)) for accurate signature.

|

| \_\_ne\_\_(self, other)

| Return self!=value.

|

| \_\_repr\_\_(self)

| Return repr(self).

|

| \_\_str\_\_ = r(\*args)

|

| get\_country(self)

| Returns the name of the country of the user.

|

| get\_friends(self, limit=50, cacheable=False, stream=False)

| Returns a list of the user's friends.

|

| get\_image(self, size=3)

| Returns the user's avatar

| size can be one of:

| SIZE\_EXTRA\_LARGE

| SIZE\_LARGE

| SIZE\_MEDIUM

| SIZE\_SMALL

|

| get\_library(self)

| Returns the associated Library object.

|

| get\_loved\_tracks(self, limit=50, cacheable=True, stream=False)

| Returns this user's loved track as a sequence of LovedTrack objects in

| reverse order of their timestamp, all the way back to the first track.

|

| If limit==None, it will try to pull all the available data.

| If stream=True, it will yield tracks as soon as a page has been retrieved.

|

| This method uses caching. Enable caching only if you're pulling a

| large amount of data.

|

| get\_name(self, properly\_capitalized=False)

| Returns the user name.

|

| get\_now\_playing(self)

| Returns the currently playing track, or None if nothing is playing.

|

| get\_playcount(self)

| Returns the user's playcount so far.

|

| get\_recent\_tracks(self, limit=10, cacheable=True, time\_from=None, time\_to=None, stream=False, now\_playing=False)

| Returns this user's played track as a sequence of PlayedTrack objects

| in reverse order of playtime, all the way back to the first track.

|

| Parameters:

| limit : If None, it will try to pull all the available data.

| from (Optional) : Beginning timestamp of a range - only display

| scrobbles after this time, in UNIX timestamp format (integer

| number of seconds since 00:00:00, January 1st 1970 UTC). This

| must be in the UTC time zone.

| to (Optional) : End timestamp of a range - only display scrobbles

| before this time, in UNIX timestamp format (integer number of

| seconds since 00:00:00, January 1st 1970 UTC). This must be in

| the UTC time zone.

| stream: If True, it will yield tracks as soon as a page has been retrieved.

|

| This method uses caching. Enable caching only if you're pulling a

| large amount of data.

|

| get\_registered(self)

| Returns the user's registration date.

|

| get\_tagged\_albums(self, tag, limit=None, cacheable=True)

| Returns the albums tagged by a user.

|

| get\_tagged\_artists(self, tag, limit=None)

| Returns the artists tagged by a user.

|

| get\_tagged\_tracks(self, tag, limit=None, cacheable=True)

| Returns the tracks tagged by a user.

|

| get\_top\_albums(self, period='overall', limit=None, cacheable=True)

| Returns the top albums played by a user.

| \* period: The period of time. Possible values:

| o PERIOD\_OVERALL

| o PERIOD\_7DAYS

| o PERIOD\_1MONTH

| o PERIOD\_3MONTHS

| o PERIOD\_6MONTHS

| o PERIOD\_12MONTHS

|

| get\_top\_artists(self, period='overall', limit=None)

| Returns the top artists played by a user.

| \* period: The period of time. Possible values:

| o PERIOD\_OVERALL

| o PERIOD\_7DAYS

| o PERIOD\_1MONTH

| o PERIOD\_3MONTHS

| o PERIOD\_6MONTHS

| o PERIOD\_12MONTHS

|

| get\_top\_tags(self, limit=None, cacheable=True)

| Returns a sequence of the top tags used by this user with their counts

| as TopItem objects.

| \* limit: The limit of how many tags to return.

| \* cacheable: Whether to cache results.

|

| get\_top\_tracks(self, period='overall', limit=None, cacheable=True, stream=False)

| Returns the top tracks played by a user.

| \* period: The period of time. Possible values:

| o PERIOD\_OVERALL

| o PERIOD\_7DAYS

| o PERIOD\_1MONTH

| o PERIOD\_3MONTHS

| o PERIOD\_6MONTHS

| o PERIOD\_12MONTHS

|

| get\_track\_scrobbles(self, artist, track, cacheable=False, stream=False)

| Get a list of this user's scrobbles of this artist's track,

| including scrobble time.

|

| get\_unixtime\_registered(self)

| Returns the user's registration date as a UNIX timestamp.

|

| get\_url(self, domain\_name=0)

| Returns the URL of the user page on the network.

| \* domain\_name: The network's language domain. Possible values:

| o DOMAIN\_ENGLISH

| o DOMAIN\_GERMAN

| o DOMAIN\_SPANISH

| o DOMAIN\_FRENCH

| o DOMAIN\_ITALIAN

| o DOMAIN\_POLISH

| o DOMAIN\_PORTUGUESE

| o DOMAIN\_SWEDISH

| o DOMAIN\_TURKISH

| o DOMAIN\_RUSSIAN

| o DOMAIN\_JAPANESE

| o DOMAIN\_CHINESE

|

| is\_subscriber(self)

| Returns whether the user is a subscriber or not. True or False.

|

| ----------------------------------------------------------------------

| Data and other attributes defined here:

|

| name = None

|

| ----------------------------------------------------------------------

| Methods inherited from \_Chartable:

|

| get\_weekly\_album\_charts(self, from\_date=None, to\_date=None)

| Returns the weekly album charts for the week starting from the

| from\_date value to the to\_date value.

| Only for User.

|

| get\_weekly\_artist\_charts(self, from\_date=None, to\_date=None)

| Returns the weekly artist charts for the week starting from the

| from\_date value to the to\_date value.

| Only for User.

|

| get\_weekly\_chart\_dates(self)

| Returns a list of From and To tuples for the available charts.

|

| get\_weekly\_charts(self, chart\_kind, from\_date=None, to\_date=None)

| Returns the weekly charts for the week starting from the

| from\_date value to the to\_date value.

| chart\_kind should be one of "album", "artist" or "track"

|

| get\_weekly\_track\_charts(self, from\_date=None, to\_date=None)

| Returns the weekly track charts for the week starting from the

| from\_date value to the to\_date value.

| Only for User.

|

| ----------------------------------------------------------------------

| Methods inherited from \_BaseObject:

|

| get\_wiki(self, section)

| Returns a section of the wiki.

| Only for Album/Track.

| section can be "content", "summary" or

| "published" (for published date)

|

| get\_wiki\_content(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_published\_date(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| get\_wiki\_summary(self)

| Returns the summary of the wiki.

| Only for Album/Track.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from \_BaseObject:

|

| \_\_dict\_\_

| dictionary for instance variables (if defined)

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Data and other attributes inherited from \_BaseObject:

|

| network = None

class WSError(PyLastError)

| WSError(network, status, details)

|

| Exception related to the Network web service

|

| Method resolution order:

| WSError

| PyLastError

| builtins.Exception

| builtins.BaseException

| builtins.object

|

| Methods defined here:

|

| \_\_init\_\_(self, network, status, details)

| Initialize self. See help(type(self)) for accurate signature.

|

| \_\_str\_\_ = r(\*args)

|

| get\_id(self)

| Returns the exception ID, from one of the following:

| STATUS\_INVALID\_SERVICE = 2

| STATUS\_INVALID\_METHOD = 3

| STATUS\_AUTH\_FAILED = 4

| STATUS\_INVALID\_FORMAT = 5

| STATUS\_INVALID\_PARAMS = 6

| STATUS\_INVALID\_RESOURCE = 7

| STATUS\_OPERATION\_FAILED = 8

| STATUS\_INVALID\_SK = 9

| STATUS\_INVALID\_API\_KEY = 10

| STATUS\_OFFLINE = 11

| STATUS\_SUBSCRIBERS\_ONLY = 12

| STATUS\_TOKEN\_UNAUTHORIZED = 14

| STATUS\_TOKEN\_EXPIRED = 15

| STATUS\_TEMPORARILY\_UNAVAILABLE = 16

| STATUS\_LOGIN\_REQUIRED = 17

| STATUS\_TRIAL\_EXPIRED = 18

| STATUS\_NOT\_ENOUGH\_CONTENT = 20

| STATUS\_NOT\_ENOUGH\_MEMBERS = 21

| STATUS\_NOT\_ENOUGH\_FANS = 22

| STATUS\_NOT\_ENOUGH\_NEIGHBOURS = 23

| STATUS\_NO\_PEAK\_RADIO = 24

| STATUS\_RADIO\_NOT\_FOUND = 25

| STATUS\_API\_KEY\_SUSPENDED = 26

| STATUS\_DEPRECATED = 27

| STATUS\_RATE\_LIMIT\_EXCEEDED = 29

|

| ----------------------------------------------------------------------

| Data descriptors inherited from PyLastError:

|

| \_\_weakref\_\_

| list of weak references to the object (if defined)

|

| ----------------------------------------------------------------------

| Static methods inherited from builtins.Exception:

|

| \_\_new\_\_(\*args, \*\*kwargs) from builtins.type

| Create and return a new object. See help(type) for accurate signature.

|

| ----------------------------------------------------------------------

| Methods inherited from builtins.BaseException:

|

| \_\_delattr\_\_(self, name, /)

| Implement delattr(self, name).

|

| \_\_getattribute\_\_(self, name, /)

| Return getattr(self, name).

|

| \_\_reduce\_\_(...)

| Helper for pickle.

|

| \_\_repr\_\_(self, /)

| Return repr(self).

|

| \_\_setattr\_\_(self, name, value, /)

| Implement setattr(self, name, value).

|

| \_\_setstate\_\_(...)

|

| with\_traceback(...)

| Exception.with\_traceback(tb) --

| set self.\_\_traceback\_\_ to tb and return self.

|

| ----------------------------------------------------------------------

| Data descriptors inherited from builtins.BaseException:

|

| \_\_cause\_\_

| exception cause

|

| \_\_context\_\_

| exception context

|

| \_\_dict\_\_

|

| \_\_suppress\_context\_\_

|

| \_\_traceback\_\_

|

| args

FUNCTIONS

cleanup\_nodes(doc)

Remove text nodes containing only whitespace

md5(text)

Returns the md5 hash of a string.

DATA

DELAY\_TIME = 0.2

DOMAIN\_CHINESE = 11

DOMAIN\_ENGLISH = 0

DOMAIN\_FRENCH = 3

DOMAIN\_GERMAN = 1

DOMAIN\_ITALIAN = 4

DOMAIN\_JAPANESE = 10

DOMAIN\_POLISH = 5

DOMAIN\_PORTUGUESE = 6

DOMAIN\_RUSSIAN = 9

DOMAIN\_SPANISH = 2

DOMAIN\_SWEDISH = 7

DOMAIN\_TURKISH = 8

IMAGES\_ORDER\_DATE = 'dateadded'

IMAGES\_ORDER\_POPULARITY = 'popularity'

PERIOD\_12MONTHS = '12month'

PERIOD\_1MONTH = '1month'

PERIOD\_3MONTHS = '3month'

PERIOD\_6MONTHS = '6month'

PERIOD\_7DAYS = '7day'

PERIOD\_OVERALL = 'overall'

SCROBBLE\_MODE\_BANNED = 'B'

SCROBBLE\_MODE\_LOVED = 'L'

SCROBBLE\_MODE\_PLAYED = ''

SCROBBLE\_MODE\_SKIPPED = 'S'

SCROBBLE\_SOURCE\_LASTFM = 'L'

SCROBBLE\_SOURCE\_NON\_PERSONALIZED\_BROADCAST = 'R'

SCROBBLE\_SOURCE\_PERSONALIZED\_BROADCAST = 'E'

SCROBBLE\_SOURCE\_UNKNOWN = 'U'

SCROBBLE\_SOURCE\_USER = 'P'

SIZE\_EXTRA\_LARGE = 3

SIZE\_LARGE = 2

SIZE\_MEDIUM = 1

SIZE\_MEGA = 4

SIZE\_SMALL = 0

SSL\_CONTEXT = <ssl.SSLContext object>

STATUS\_API\_KEY\_SUSPENDED = 26

STATUS\_AUTH\_FAILED = 4

STATUS\_DEPRECATED = 27

STATUS\_INVALID\_API\_KEY = 10

STATUS\_INVALID\_FORMAT = 5

STATUS\_INVALID\_METHOD = 3

STATUS\_INVALID\_PARAMS = 6

STATUS\_INVALID\_RESOURCE = 7

STATUS\_INVALID\_SERVICE = 2

STATUS\_INVALID\_SIGNATURE = 13

STATUS\_INVALID\_SK = 9

STATUS\_LOGIN\_REQUIRED = 17

STATUS\_NOT\_ENOUGH\_CONTENT = 20

STATUS\_NOT\_ENOUGH\_FANS = 22

STATUS\_NOT\_ENOUGH\_MEMBERS = 21

STATUS\_NOT\_ENOUGH\_NEIGHBOURS = 23

STATUS\_NO\_PEAK\_RADIO = 24

STATUS\_OFFLINE = 11

STATUS\_OPERATION\_FAILED = 8

STATUS\_RADIO\_NOT\_FOUND = 25

STATUS\_RATE\_LIMIT\_EXCEEDED = 29

STATUS\_SUBSCRIBERS\_ONLY = 12

STATUS\_TEMPORARILY\_UNAVAILABLE = 16

STATUS\_TOKEN\_EXPIRED = 15

STATUS\_TOKEN\_UNAUTHORIZED = 14

STATUS\_TRIAL\_EXPIRED = 18

\_\_copyright\_\_ = 'Copyright (C) 2008-2010 Amr Hassan, 2013-2021 hugovk,...

\_\_email\_\_ = 'amr.hassan@gmail.com'

\_\_license\_\_ = 'apache2'

logger = <Logger pylast (WARNING)>

VERSION

4.4.0

AUTHOR

Amr Hassan, hugovk, Mice Pápai

FILE

c:\users\matth\anaconda3\lib\site-packages\pylast\\_\_init\_\_.py