

Package ‘FlickrAPI’

July 21, 2025

Title Access to Flickr API

Version 0.1.0.1

Description Provides an interface to the Flickr API <<https://www.flickr.com/services/api/>>
and allows R users to download data on Flickr.

Depends R (>= 3.1)

License GPL (>= 3)

Encoding UTF-8

Imports httr2, janitor, magrittr, rlang, utils

Suggests sf, knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.2.0.9000

NeedsCompilation no

Author Koki Ando [aut, cre],
Eli Pousson [aut, ctb]

Maintainer Koki Ando <koki.25.ando@gmail.com>

Repository CRAN

Date/Publication 2022-07-11 12:40:10 UTC

Contents

FlickrAPIRequest	2
getExif	2
getHotTags	3
getPhotoInfo	4
getPhotos	5
getPhotoSearch	7
setFlickrAPIKey	9

Index	11
--------------	-----------

FlickrAPIRequest	<i>Request data from the Flickr API</i>
------------------	---

Description

Request data from the Flickr API using the provided method, API key, and any additional values passed to [httr2::req_url_query].

Usage

```
FlickrAPIRequest(
  method = NULL,
  api_key = NULL,
  format = "json",
  simplifyVector = TRUE,
  check_type = FALSE,
  ...
)
```

Arguments

method	Flickr API method to use for request.
api_key	Flickr API key. If api_key is 'NULL', the function uses [getFlickrAPIKey()] to use the environment variable "FLICKR_API_KEY" as the key.
format	Format parameter passed to [httr2::req_url_query]
simplifyVector	Default to 'TRUE', passed to [httr2::resp_body_json]
check_type	Default to 'FALSE', passed to [httr2::resp_body_json]
...	Additional parameters passed to [httr2::req_url_query]

getExif	<i>Get EXIF data for a Flickr photo.</i>
---------	--

Description

Return a data of EXIF data for a given photo. The calling user must have permission to view the photo.

Usage

```
getExif(api_key = NULL, photo_id = NULL)

get_exif(api_key = NULL, photo_id = NULL)
```

Arguments

api_key	Flickr API key. If api_key is 'NULL', the function uses [getFlickrAPIKey()] to use the environment variable "FLICKR_API_KEY" as the key.
photo_id	The id of the photo to fetch information for

Value

This function returns a data frame of EXIF information of given photograph

Examples

```
## Not run:
getExif(api_key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX", photo_id = "45961963324")

## End(Not run)
```

getHotTags	<i>Get a data frame of hot tags for a given time period.</i>
------------	--

Description

Get a data frame of hot tags for a given time period.

Usage

```
getHotTags(api_key = NULL, period = c("day", "week"), count = 20)

get_hot_tags(api_key = NULL, period = c("day", "week"), count = 20)
```

Arguments

api_key	Flickr API key. If api_key is 'NULL', the function uses [getFlickrAPIKey()] to use the environment variable "FLICKR_API_KEY" as the key.
period	The period for which to fetch hot tags. Valid values are day or week. Defaults to day
count	The number of tags to return. Defaults to 20. Maximum allowed value is 200.

Value

This function a data frame of hot tags for the given period

See Also

<https://www.flickr.com/services/api/flickr.tags.getHotList.html>

Examples

```
## Not run:
getHotTags(
  api_key = "*****",
  period = "day", count = 20
)

## End(Not run)
```

getPhotoInfo	<i>Get available information for a Flickr photo.</i>
--------------	--

Description

R access to photo information of photos posted on Flickr via Flickr API.

Usage

```
getPhotoInfo(
  api_key = NULL,
  photo_id,
  output = c("location", "date", "url", "tags")
)

get_photo_info(
  api_key = NULL,
  photo_id,
  output = c("location", "date", "url", "tags")
)
```

Arguments

api_key	Flickr API key. If api_key is 'NULL', the function uses [getFlickrAPIKey()] to use the environment variable "FLICKR_API_KEY" as the key.
photo_id	The id of the photo to get information for.
output	Output data type. Supported options include "all", "location", "date", "url" or "tags". If output = "all", the function returns a list with all available data. Otherwise the function returns a data frame.

Value

This function returns data of specific photo's information.

See Also

<https://www.flickr.com/services/api/flickr.photos.getInfo.html>

Examples

```
## Not run:
getPhotoInfo(
  api_key = "XXXXXXXXXX",
  photo_id = "30484882493",
  output = "location"
)

## End(Not run)
```

getPhotos

Get Flickr user photos or a user's favorite photos

Description

Returns photos from the given user's photo stream. Only photos visible to the calling user will be returned.

Usage

```
getPhotos(user_id = NULL, img_size = "s", extras = NULL, api_key = NULL, ...)
```

```
get_photos(user_id = NULL, img_size = "s", extras = NULL, api_key = NULL, ...)
```

```
getFavePhotos(
  user_id = NULL,
  img_size = "s",
  extras = NULL,
  fave_date = NULL,
  public = TRUE,
  api_key = NULL,
  page = NULL,
  per_page = 100,
  ...
)
```

```
get_fave_photos(
  user_id = NULL,
  img_size = "s",
  extras = NULL,
  fave_date = NULL,
  public = TRUE,
  api_key = NULL,
  page = NULL,
  per_page = 100,
  ...
)
```

Arguments

<code>user_id</code>	The NSID of the user whose photos to return. A value of "me" return the calling user's photos.
<code>img_size</code>	A character string with the abbreviation for one or more image sizes ("sq", "t", "s", "q", "m", "n", "z", "c", "l", or "o"). If a single <code>img_size</code> is provided the url, width, and height columns are renamed (e.g. <code>img_url</code> instead of <code>url_sq</code>) and an <code>img_asp</code> column is added to the results; defaults to 'NULL'.
<code>extras</code>	A vector of extra information to fetch for each returned record. Currently supported fields are: <code>c("description", "license", "date_upload", "date_taken", "owner_name", "icon_server", "original_format", "last_update", "geo", "tags", "machine_tags", "o_dims", "views", "media", "path_alias", "url_sq", "url_t", "url_s", "url_q", "url_m", "url_n", "url_z", "url_c", "url_l", "url_o")</code>
<code>api_key</code>	Flickr API key. If <code>api_key</code> is 'NULL', the function uses <code>[getFlickrAPIKey()]</code> to use the environment variable "FLICKR_API_KEY" as the key.
<code>...</code>	Additional parameters passed to <code>[getPhotoSearch]</code>
<code>fave_date</code>	Length 1 or 2 vector with UNIX formatted date (may include minimum and maximum favorite date).
<code>public</code>	If 'TRUE', get public favorites (no authentication needed). If 'FALSE', get all favorite (requires authentication for access).
<code>page</code>	Number specifying which search results page to return. Default is page 1 of results returned.
<code>per_page</code>	Number specifying how many results per page to return. Default 100 results per page. Maximum of 250 if 'bbox' provided or 500 otherwise.

Details

For 'getFavePhotos' the provided `user_id` may need to be the NSID code or the user name rather than the typical user id due to some inconsistencies in the Flickr API.

Value

This function returns a `data.frame` including columns:

- `id`
- `owner`
- `secret`
- `server`
- `farm`
- `title`
- `ispublic`
- `isfriend`
- `isfamily`

Examples

```
## Not run:
getPhotos(api_key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX", user_id = "141696738@N08")

## End(Not run)
```

getPhotoSearch	<i>Search for photos on Flickr by user id, tags, license, or bounding box</i>
----------------	---

Description

Use the Flickr Search API to return pages of photos sorted by date posted, date taken, interestingness, and relevance. Optional search parameters including spatial bounding box, user id, tags, and image license.

Usage

```
getPhotoSearch(
  api_key = NULL,
  user_id = NULL,
  tags = NULL,
  license_id = NULL,
  sort = "date-posted",
  desc = FALSE,
  bbox = NULL,
  img_size = NULL,
  extras = NULL,
  per_page = 100,
  page = NULL,
  ...
)
```

```
get_photo_search(
  api_key = NULL,
  user_id = NULL,
  tags = NULL,
  license_id = NULL,
  sort = "date-posted",
  desc = FALSE,
  bbox = NULL,
  img_size = NULL,
  extras = NULL,
  per_page = 100,
  page = NULL,
  ...
)
```

Arguments

<code>api_key</code>	Flickr API key. If <code>api_key</code> is 'NULL', the function uses <code>[getFlickrAPIKey()]</code> to use the environment variable "FLICKR_API_KEY" as the key.
<code>user_id</code>	The NSID of the user with photos to search. If this parameter is NULL passed then all public photos will be searched.
<code>tags</code>	A vector of tags to search for.
<code>license_id</code>	The license id for photos. For possible values see the Flickr API method <code>flickr.photos.licenses.getInfo</code> or see details for more information.
<code>sort</code>	Order to sort returned photos. The possible values are: "date-posted-asc", "date-posted-desc", "date-taken-asc", "date-taken-desc", "interestingness-asc", and "relevance" The trailing "-asc" or "-desc" indicator for sort direction is optional when using the desc parameter.
<code>desc</code>	If 'TRUE', sort in descending order by the selected sort variable; defaults to 'FALSE'.
<code>bbox</code>	A object of class 'bbox' or a numeric vector with values for xmin, ymin, xmax and ymax representing the bottom-left corner of the box and the top-right corner.
<code>img_size</code>	A character string with the abbreviation for one or more image sizes ("sq", "t", "s", "q", "m", "n", "z", "c", "l", or "o"). If a single <code>img_size</code> is provided the url, width, and height columns are renamed (e.g. <code>img_url</code> instead of <code>url_sq</code>) and an <code>img_asp</code> column is added to the results; defaults to 'NULL'.
<code>extras</code>	A vector of extra information to fetch for each returned record. Currently supported fields are: <code>c("description", "license", "date_upload", "date_taken", "owner_name", "icon_server", "original_format", "last_update", "geo", "tags", "machine_tags", "o_dims", "views", "media", "path_alias", "url_sq", "url_t", "url_s", "url_q", "url_m", "url_n", "url_z", "url_c", "url_l", "url_o")</code>
<code>per_page</code>	Number specifying how many results per page to return. Default 100 results per page. Maximum of 250 if 'bbox' provided or 500 otherwise.
<code>page</code>	Number specifying which search results page to return. Default is page 1 of results returned.
<code>...</code>	Additional parameters that can include <code>licence_id</code> (legacy spelling),

Details

License id options:

`license_id` can be an integer from 0 to 10 or a corresponding license code including:

- "c" (All Rights Reserved), - "by-bc-sa" (Attribution-NonCommercial-ShareAlike), - "by-nc" (Attribution-NonCommercial), - "by-nc-nd" (Attribution-NonCommercial-NoDerivs), - "by" (Attribution), - "by-sa" (Attribution-ShareAlike), - "by-nd" (Attribution-NoDerivs), - "nkc" (No known copyright restrictions), - "pd-us" (United States Government Work), - "cc0" (Public Domain Dedication), - or "pd" (Public Domain Mark).

Value

This function returns data of specific photos matching search parameters.

Examples

```
## Not run:
# Search for photos tagged "cats" and "dogs"
# Return images in descending order of date taken
getPhotoSearch(
  api_key = "XXXXXXXXXX",
  sort = "date-taken-desc",
  tags = c("cats", "dogs")
)

## End(Not run)
## Not run:
# Search for photos uploaded to the NPS Grand Canyon user account.
# Return extra fields including the date taken and square image URL.
getPhotoSearch(
  api_key = "XXXXXXXXXX",
  user_id = "grand_canyon_nps",
  extras = c("date_taken", "url_sq")
)

## End(Not run)
## Not run:
# Search for photos tagged "panda" in the area of Ueno Zoo, Tokyo, Japan
getPhotoSearch(
  api_key = "XXXXXXXXXX",
  tags = "panda",
  bbox = c(139.7682226529, 35.712627977, 139.7724605432, 35.7181464141),
  extras = c("geo", "owner_name", "tags")
)

## End(Not run)
```

setFlickrAPIKey	<i>Set and get a Flickr API key to/from environment variables.</i>
-----------------	--

Description

Set and get a Flickr API key to/from environment variables.

Get Flickr API key from environment variables

Usage

```
setFlickrAPIKey(api_key, overwrite = FALSE, install = FALSE)

set_flickr_api_key(api_key, overwrite = FALSE, install = FALSE)

getFlickrAPIKey(api_key = NULL)

get_flickr_api_key(api_key = NULL)
```

Arguments

<code>api_key</code>	Flickr API key
<code>overwrite</code>	If 'TRUE', overwrite existing token; Default: 'FALSE'
<code>install</code>	If 'TRUE', install token for use in future sessions; Default: 'FALSE'

Index

FlickrAPIRequest, [2](#)

get_exif (getExif), [2](#)

get_fave_photos (getPhotos), [5](#)

get_flickr_api_key (setFlickrAPIKey), [9](#)

get_hot_tags (getHotTags), [3](#)

get_photo_info (getPhotoInfo), [4](#)

get_photo_search (getPhotoSearch), [7](#)

get_photos (getPhotos), [5](#)

getExif, [2](#)

getFavePhotos (getPhotos), [5](#)

getFlickrAPIKey (setFlickrAPIKey), [9](#)

getHotTags, [3](#)

getPhotoInfo, [4](#)

getPhotos, [5](#)

getPhotoSearch, [7](#)

set_flickr_api_key (setFlickrAPIKey), [9](#)

setFlickrAPIKey, [9](#)