Package ‘imguR’

August 29, 2016

Type Package
Title An Imgur.com API Client Package
Version 1.0.3
Date 2016-03-29
Maintainer Thomas J. Leeper <thosjleeper@gmail.com>
Imports httr, png, jpeg, tools
Description A complete API client for the image hosting service Imgur.com, including the an imgur graphics device, enabling the easy upload and sharing of plots.
License GPL-3
URL https://github.com/leeper/imguR
BugReports https://github.com/leeper/imguR/issues
NeedsCompilation no
Author Thomas J. Leeper [aut, cre],
Aaron Statham [ctb],
Yihui Xie [ctb]
Repository CRAN
Date/Publication 2016-03-29 11:55:23

R topics documented:

imguR-package .................................................. 3
account ........................................................... 3
account_albums ............................................... 4
account_comments .......................................... 5
account_favorites .......................................... 6
account_gallery ............................................ 7
account_images ........................................... 8
account_settings ........................................ 9
add_album_images ......................................... 10
add_gallery_images ..................................... 11
Authentication ........................................... 12
block_sender .............................................................. 13
create_album .......................................................... 14
create_comment ......................................................... 15
create_message ........................................................ 16
delete_album ............................................................ 17
delete_comment ......................................................... 18
delete_conversation .................................................... 19
delete_image ............................................................. 20
favorite_album ......................................................... 21
favorite_image ........................................................ 22
get_album ................................................................. 23
get_comment ............................................................. 24
get_conversations ...................................................... 25
get_gallery ............................................................... 26
get_gallery_album ....................................................... 27
get_gallery_image ........................................................ 28
get_image ................................................................. 29
get_notifications ........................................................ 30
get_random_images ..................................................... 31
get_reddit_gallery ..................................................... 32
get_reddit_image ........................................................ 33
get_replies ............................................................... 34
http ......................................................................... 35
imgur ....................................................................... 36
imgur_login ............................................................. 37
imgur_off ................................................................. 38
mark_notification ..................................................... 39
memes ..................................................................... 40
plot ......................................................................... 41
rate_limit ................................................................. 42
remove_gallery_image ................................................ 43
report_comment ......................................................... 43
report_gallery_image .................................................. 44
report_sender ......................................................... 45
search_gallery .......................................................... 46
update_album ........................................................... 47
update_image ........................................................... 48
upload_image ........................................................... 49
vote_comment ........................................................... 51

Index 52
Description

This package enables intuitive uploading of plots created in R to the free image hosting service [Imgur](http://www.imgur.com) simply, quickly, and intuitively via the [Imgur v3 API](http://api.imgur.com). The package is a complete client library for the Imgur API, meaning that one can also delete, favorite, and comment on images, as well as organize them in albums and galleries. By default, all images are loaded to Imgur anonymously. Optionally, using an OAuth2.0 login, users can also gain fully access to their own Imgur user account.

Author(s)

Thomas J. Leeper and Aaron Statham

Maintainer: Thomas J. Leeper <thosjleeper@gmail.com>

References

[Imgur API Documentation](https://www.imgur.com)

account

Retrieve user account details

Description

Retrieve details of a user’s account

Usage

account(account = 'me', ...)

account_verified(...)

send_verification(...)

Arguments

account A username, by default the current user (if token is supplied). account_verified and send_verification require a token argument.

... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).
account_albums

Details

account returns basic details of a user account. account_verified indicates whether one’s user account has been email-verified. send_verification sends a verification email to the user’s email account.

Value

An object of class “imgur_account”.

Author(s)

Thomas J. Leeper

Examples

```r
# Not run:
# get account details for a user account
account('thosjleeper')

# check account verification
tkn <- imgur_login()
if(!account_verified(token = tkn))
  send_verification(token = tkn)
account(token = tkn)

# End(Not run)
```

account_albums | Retrieve user albums

Description

Retrieve details of a user’s albums

Usage

```r
account_albums(account = 'me', ids = TRUE, ...)

account_album_count(account = 'me', ...)
```

Arguments

- **account**: A username, by default the current user (if token is supplied).
- **ids**: A logical indicating whether to only return album IDs.
- **...**: Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).
account_comments

Details

account_albums retrieves details of a user's albums. By default (and if a token argument is supplied), the current user's albums are returned. Specifying another username is also possible with or without a token. account_album_count returns just the number of albums, which is useful for handling pagination of the results.

Value

For account_albums, a list of objects of class "imgur_albums". For account_album_count, a number.

Author(s)

Thomas J. Leeper

Examples

```r
## Not run:
account_albums('thosjleeper')
## End(Not run)
```

Description

Retrieve user albums

Usage

```r
account_comments(account = 'me', ids = FALSE, ...)
account_comment_count(account = 'me', ...)
```

Arguments

- **account**: A username, by default the current user (if token is supplied).
- **ids**: A logical indicating whether to only return album IDs.
- **...**: Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

account_comments retrieves details of a user's comments. By default (and if a token argument is supplied), the current user's comments are returned. Specifying another username is also possible with or without a token. account_comment_count returns just the number of comments, which is useful for handling pagination of the results.
accountFavorites

Description
Retrieve details of a user’s favorite images

Usage
accountFavorites(account = 'me', gallery = FALSE, ...)

Arguments
- account: A username, by default the current user (if token is supplied).
- gallery: A logical indicating whether to only return gallery favorites.
- ...: Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details
accountFavorites retrieves details of a user’s favorites. By default (and if a token argument is supplied), the current user’s favorites are returned. Specifying another username is also possible with or without a token.

Value
For accountFavorites, an object of class “imgur_gallery_image” or “imgur_gallery_album”.

Author(s)
Thomas J. Leeper
account_gallery

Examples

## not run:
account_favorites('thosjleeper')

## End(Not run)

---

account_gallery  User gallery profile

Description

Retrieve details of a user’s gallery profile

Usage

account_gallery(account = 'me', ...)

Arguments

account  A username, by default the current user (if token is supplied). account_verified and send_verification require a token argument.

...  Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

account_gallery returns basic details of a user’s gallery profile, including counts of comments, likes, submissions, and trophies.

Value

An object of class “imgur_gallery_profile”.

Author(s)

Thomas J. Leeper

Examples

## not run:
tkn <- imgur_login()
account_gallery(token = tkn)

## End(Not run)
account_images

Retrieve user images

Description

Retrieve details of a user’s images

Usage

account_images(account = 'me', page = NULL, ids = TRUE, ...)

account_image_count(account = 'me', ...)

Arguments

account  A username, by default the current user (if token is supplied).
page     Optionally, a page of results to return.
ids      A logical indicating whether to only return image IDs.
...      Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

account_images retrieves details of a user’s images. By default (and if a token argument is supplied), the current user’s images are returned. Specifying another username is also possible with or without a token. account_image_count returns just the number of images, which is useful for handling pagination of the results.

Value

For account_images, an object of class “imgur_image”. For account_image_count, a number.

Author(s)

Thomas J. Leeper

Examples

## Not run:
account_images('thosjleeper')

## End(Not run)
Description

Retrieve and modify account settings

Usage

```python
account_settings(bio = NULL,
    public_images = NULL,
    messaging_enabled = NULL,
    album_privacy = NULL,
    accepted_gallery_terms = NULL,
    ...)
```

Arguments

- **bio** The biography of user, which is displayed in the gallery profile page.
- **public_images** A logical indicating whether images public by default.
- **messaging_enabled** A logical indicating whether private messages are allowed.
- **album_privacy** The default privacy for albums. One of “public”, “hidden”, or “secret”.
- **accepted_gallery_terms** A logical indicating whether the user agrees to the Imgur gallery terms.
- **...** Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token), which is required.

Details

If none of the named arguments (other than `token`) are specified, this retrieves settings of the account. If any of the named arguments are specified, those parameters are modified.

Value

If not modifying settings, an object of “imgur_account_settings”, otherwise an object of “imgur_basic”.

Author(s)

Thomas J. Leeper
add_album_images

Examples

```r
## Not run:
tkn <- imgur_login()

# retrieve settings
account_settings(token = tkn)

# set albums to default to public
account_settings(album_privacy = 'public', token = tkn)

## End(Not run)
```

Description

Add images to or remove images from an album

Usage

```r
add_album_images(album, id, ...)
remove_album_images(album, id, ...)
set_album_images(album, id, ...)
```

Arguments

- **album**: An album ID or object of class “imgur_album”.
- **id**: A vector of one or more image IDs to add to the album or a list of objects of class “imgur_image”.
- **...**: Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

Details

- `add_album_images` adds the specified images to the album.
- `set_album_images` removes all current images from the album and adds the specified images to the album.
- `remove_album_images` removes the specified images from the album.

Value

An object of class “imgur_basic”.
add_gallery_images

Author(s)

Thomas J. Leeper

See Also

create_album
update_album

---

add_gallery_images Add gallery images

Description

Add images to the gallery

Usage

add_gallery_images(album = NULL, id = NULL, title = NULL, ...)

Arguments

album An album ID or object of class “imgur_album” to add to the gallery.
id An image ID or an object of class “imgur_image” to add to the gallery.
title A title for the image
...

Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

Details

This function adds the specified image or album to the gallery. title is required by the API. If NULL, the title is drawn from a call to get_album or get_image.

Value

An object of class “imgur_gallery_image” or “imgur_gallery_album”.

Author(s)

Thomas J. Leeper

See Also

remove_gallery_image
Authentication

### Description

Description of Imgur API authentication options

### Details

All calls to the Imgur.com API must be authenticated. The package provides two authentication methods: anonymous API key authentication (the default) and OAuth2 authentication. By default, all calls are made anonymously. Most operations can be performed in this way. The package provides an API key for performing operations in this way. This default key can be overridden by passing the `key` argument through any function. This may be useful if you intend to make a large number of anonymous requests using your own registered application.

Alternatively, calls can be made with a user-specific OAuth2.0 token. This token is generated using `imgur_login` in an interactive session. The token allows calls to be made to a specific Imgur.com user account, which eases making subsequent changes to uploaded images (e.g., updating their metadata, adding images to an album, or deleting images). The token can then be passed using the `token` argument to any function.

Note: if both `key` and `token` arguments are supplied to a function, the `token` argument always takes precedence.

### References

https://api.imgur.com/oauth2

### See Also

`imgur_login`

### Examples

```r
## Not run:
# OAuth2.0-authenticated use

# interactive OAuth login
tkn <- imgur_login()

# upload a simple graph
i <- imgur(token = tkn)
hist(rnorm(100))
u <- imgur_off(i)

# pass the `imgur_image` object to `update_image`:
update_image(u, title = 'My graph', description = 'A simple graph', token = tkn)

# or, pass just the image id:
```
block_sender

update_image(u$id, title = 'My graph', description = 'A simple graph', token = tkn)

# periodically refresh token
tkn$refresh()

## End(Not run)

## Not run:
# anonymous use

# upload a simple graph
i <- imgur()
hist(rnorm(100))
u <- imgur_off(i)

# pass just the image deletehash:
update_image(u$deletehash, title = 'My graph', description = 'A simple graph')

## End(Not run)

---

**block_sender**  
*Block a user*

**Description**

Block a user from sending messages

**Usage**

```
block_sender(username, ...)
```

**Arguments**

- `username`  
  A character string containing an Imgur username.

- `...`  
  Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token), which is required.

**Details**

Block an Imgur user from sending messages.

**Value**

An object of class “imgur_basic”.

**Author(s)**

Thomas J. Leeper
create_album

Create album

Description

Create an album

Usage

create_album(id = NULL,
  title = NULL, description = NULL,
  privacy = NULL, layout = NULL, cover_id = NULL, ...)

Arguments

id
  A vector of one or more image IDs to add to the album or a list of objects of
class "imgur_image".
title
  Optionally, an album title.
description
  Optionally, an album description.
privacy
  The privacy for the album. One of "public", "hidden", or "secret".
layout
  The layout for the album. One of "blog", "grid", "horizontal", or "vertical".)
cover_id
  An image ID or an object of class "imgur_image".
...
  Other arguments passed to HTTP request functions, for example: token (an
  OAuth2.0 token) or key (an API key, used by default).

Details

Creates an Imgur album with the specified images and metadata. All options are are optional.
Images can only be added to anonymous albums with create_album, not after the fact. Albums
created with an OAuth2.0 authentication token can be modified in any way.

Value

An object of class "imgur_basic".

Author(s)

Thomas J. Leeper


create_comment

See Also

update_album
add_album_images
remove_album_images
set_album_images

Examples

```r
## Not run:
i <- imgur()
hist(rnorm(20))
img <- imgur_off(i)
create_album(id = img,
  title = 'My first Imgur album',
  description = 'A simple album',
  privacy = 'hidden',
  layout = 'grid',
  cover_id = img)

## End(Not run)
```

create_comment Create a comments

Description

Create a comment on an image

Usage

create_comment(id, comment, parent = NULL, ...)

Arguments

id An image ID or an object of class “imgur_image”.
comment A character string containing the comment.
parent An optional comment ID to reply to.
... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

Details

account_favorites retrieves details of a user’s favoriates. By default (and if a token argument is supplied), the current user’s favorites are returned. Specifying another username is also possible with or without a token.
create_message

**Value**

For `account_favorites`, an object of class “imgur_gallery_image” or “imgur_gallery_album”.

**Author(s)**

Thomas J. Leeper

**See Also**

- `get_comment`
- `delete_comment`

**Examples**

```r
## Not run:
tkn <- imgur_login()

# comment on an image
cmt <- create_comment("vsxfu46", "This is a test comment", token = tkn)

# create a reply comment
create_comment("vsxfu46", "This is a test comment", parent = cmt, token = tkn)

## End(Not run)
```

---

**create_message**

Create a comment on an image

**Description**

Create a comment on an image

**Usage**

```r
create_message(recipient, body, ...)
```

**Arguments**

- **recipient**: A character string containing the recipient’s Imgur username.
- **body**: A character string containing the message.
- **...**: Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token), which is required.

**Details**

Send a message to another Imgur user.
**delete_album**

**Value**

An object of class “imgur_basic”.

**Author(s)**

Thomas J. Leeper

**See Also**

- get_conversations
- delete_conversation

**Examples**

```r
## not run:
tkn <- imgur_login()
create_message("thosjleeper", "This is a test comment from imgur", token = tkn)

## End(not run)
```

---

**Description**

Delete an album

**Usage**

```r
delete_album(album, ...)
```

**Arguments**

- `album`  
  An album ID or object of class “imgur_image”. If no token is supplied, `id` must be a deletehash from the original upload_image response.

- `...`  
  Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token) or `key` (an API key, used by default).

**Details**

Deletes an album.

**Value**

An object of class “imgur_basic”. 

```
delete_comment

Author(s)
Thomas J. Leeper

See Also
create_album
favorite_album
update_album

Examples
## not run:
# using an OAuth token
tkn <- imgur_login()
a <- account_albums(token = tkn)
deleate_album(a[[1]], token = tkn)
## End(Not run)

delete_comment  Delete a comment

Description
Delete a comment from an image

Usage
delete_comment(comment, ...)

Arguments

comment  A comment ID, perhaps returned by create_comment.

...  Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

Details
Delete a comment.

Value
An object of class “imgur_basic”

Author(s)
Thomas J. Leeper
**delete_conversation**

See Also

- `get_comment`
- `create_comment`

Examples

```r
## Not run:
tkn <- imgur_login()
cmt <- create_comment("vSXfU46", comment = 'This is a test comment.')
delete_comment(cmt, token = tkn)

## End(Not run)
```

Description

Delete a conversation

Usage

```r
delete_conversation(id, ...)
```

Arguments

- `id` A conversation ID, possibly returned by `get_conversations`.
- `...` Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

Details

Deletes a conversation

Value

An object of class "imgur_basic".

Author(s)

Thomas J. Leeper

See Also

- `get_conversations`
- `create_message`
- `delete_conversation`
delete_image

**Description**
Delete an image

**Usage**
delete_image(id, ...)

**Arguments**
id An image ID or object of class “imgur_image”. If no token is supplied, id must be a deletehash from the original upload_image response.
... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

**Details**
Deletes an image.

**Value**
An object of class “imgur_basic”.

**Author(s)**
Thomas J. Leeper

**See Also**
upload_image
update_image

**Examples**
```r
## not run:
# using a deletehash, anonymously
u <- upload_image('file.png')
delete_image(u$deletehash)

# using an OAuth token
tkn <- imgur_login()
u <- upload_image('file.png', token = tkn)
delete_image(u, token = tkn)
```

## End(not run)
Description

Favorite an album

Usage

favorite_album(album, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>album</td>
<td>An album ID or object of class “imgur_album”.</td>
</tr>
<tr>
<td>...</td>
<td>Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).</td>
</tr>
</tbody>
</table>

Details

Favorites an album.

Value

An object of class “imgur_basic”.

Author(s)

Thomas J. Leeper

Examples

## not run:
```
tkn <- imgur_login()
favorite_album("cVsXX", token = tkn)
```

## End(not run)
favorite_image  Favorite image

Description
Favorite an image

Usage
favorite_image(id, ...)

Arguments
id  An image ID or object of class “imgur_image”.
...  Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details
Favorites an image.

Value
An object of class “imgur_basic”.

Author(s)
Thomas J. Leeper

Examples
## not run:
  tkn <- imgur_login()
  favorite_image("DgqMueq", token = tkn)

## End(not run)
Description

Get album details and metadata, or images.

Usage

get_album(album, ...)

get_album_images(album, ...)

Arguments

album An album ID or object of class “imgur_album”.

... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

Get album details and metadata, including images contained in the album.

Value

An object of class “imgur_album”.

Author(s)

Thomas J. Leeper

Examples

## not run:
get_album("cVsXX")

get_album_images("cVsXX")

## End(Not run)
get_comment

Get a comment

**Description**

Get a comment (and, optionally, its replies).

**Usage**

```r
get_comment(comment, replies = FALSE, ...)
```

**Arguments**

- `comment`: A comment ID, perhaps returned by `create_comment`.
- `replies`: A logical indicating whether to return replies to the comment, in addition to the comment itself.
- `...`: Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token) or `key` (an API key, used by default).

**Details**

Retrieves a comment and, optionally, any replies to that comment.

**Value**

An object of class “imgur_comment”

**Author(s)**

Thomas J. Leeper

**See Also**

- `create_comment`
- `delete_comment`
**get_conversations**  

Get conversations

**Description**

Get a specific conversation or all conversations

**Usage**

```
get_conversations(id = NULL, ...)
```

**Arguments**

- `id`  
  A conversation ID. If NULL, all conversations are returned.

- `...`  
  Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token), which is required.

**Details**

Retrieves a named conversation, possibly returned by `get_conversations`, or all conversations.

**Value**

An object of class “imgur_message”.

**Author(s)**

Thomas J. Leeper

**See Also**

- `create_message`
- `delete_conversation`

**Examples**

```r
## Not run:
tkn <- imgur_login()
get_conversations(token = tkn)

## End(Not run)
```
get_gallery

**Description**

Get gallery details and metadata

**Usage**

```r
get_gallery(section = 'hot', sort = 'viral', page = 0,
            window = 'day', showViral = TRUE, ...)
```

**Arguments**

- **section**: One of “hot” (the default), “top”, or “user”.
- **sort**: One of “viral” (the default) or “time”.
- **page**: A page number of results between 0 and 50.
- **window**: If `section` == ‘top’, one of “day” (the default), “week”, “month”, “year”, or “all”.
- **showViral**: If `section` == ‘user’, a logical indicating whether show viral images.
- **...**: Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token) or `key` (an API key, used by default).

**Details**

Get gallery image details and metadata.

**Value**

An object of class “imgur_image”.

**Author(s)**

Thomas J. Leeper

**See Also**

- `search_gallery`
- `get_gallery_image`
- `get_random_images`
get_gallery_album

Examples

```r
## Not run:
get_gallery()

get_gallery('top', window = 'week')
```

## end(not run)

---

**get_gallery_album**  
Get gallery album

**Description**

Get album details and metadata for an album in the gallery.

**Usage**

```r
get_gallery_album(album, ...)
```

**Arguments**

- `album`  
  An album ID or object of class “imgur_album” or “imgur_gallery_album”.

- `...`  
  Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

**Details**

Get album details and metadata, for an album in the gallery. This is basically the same as “get_album”.

**Value**

An object of class “imgur_gallery_album”.

**Author(s)**

Thomas J. Leeper

**See Also**

- `get_album`
- `get_gallery`
- `search_gallery`
get_gallery_image  
*Get gallery image*

**Description**

Get gallery image details and metadata

**Usage**

```r
get_gallery_image(id, ...)
```

**Arguments**

- `id` An image ID or object of class “imgur_image” or “imgur_gallery_image”.
- `...` Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

**Details**

Get gallery image details and metadata. This is basically the same as `get_image`.

**Value**

An object of class “imgur_image”.

**Author(s)**

Thomas J. Leeper

**See Also**

`get_gallery`

`search_gallery`

`get_image`

**Examples**

```r
## not run:
s <- search_gallery(query = 'meme')
gallery_image(s[[1]])

## End(not run)
```
Description

Get image details and metadata

Usage

get_image(id, ...)

Arguments

id An image ID or object of class “imgur_image”.
...

Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

Get image details and metadata.

Value

An object of class “imgur_image”.

Author(s)

Thomas J. Leeper

See Also

plot.imgur_image

Examples

## not run:
get_image("vsxfuTV")

## end(not run)
**get_notifications**

---

**Description**

Get a specific notification or all notifications

**Usage**

```r
get_notifications(id = NULL, only_new = FALSE, ...)
```

**Arguments**

- `id` A conversation ID. If NULL, all notifications (or all new notifications, depending on `only_new`) are returned.
- `only_new` A logical indicating whether only new notifications should be returned. Default is FALSE.
- `...` Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

**Details**

Retrieves a named notification, possibly returned by `get_notifications`, or all (new) notifications.

**Value**

An object of class “imgur_notification”.

**Author(s)**

Thomas J. Leeper

**See Also**

- `mark_notification`

**Examples**

```r
## Not run:
tkn <- imgur_login()
get_notifications(only_new = TRUE, token = tkn)
## End(Not run)
```
Description

Get random images from the public gallery

Usage

get_random_images(page = 0, ...)

Arguments

- **page**: A page number of results between 0 and 50.
- **...**: Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

Get random images from the public gallery.

Value

An object of class “imgur_gallery_album”.

Author(s)

Thomas J. Leeper

See Also

get_gallery
search_gallery
get_gallery_image

Examples

```r
## Not run:
get_random_images()

## End(Not run)
```
get_reddit_gallery  Get gallery

Description

Get gallery details and metadata

Usage

get_reddit_gallery(subreddit, sort = 'time', page = 0, window = 'day', ...)

Arguments

subreddit  A character string containing the name of a sub-reddit.
sort  One of “time” (the default) or “top”.
page  A page number of results between 0 and 50.
window  If sort == ‘top’, one of “day” (the default), “week”, “month”, “year”, or “all”.
...  Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

Get gallery image details and metadata for a sub-reddit gallery.

Value

An object of class “imgur_image”.

Author(s)

Thomas J. Leeper

See Also

search_gallery
get_gallery
get_random_images

Examples

## not run:
get_reddit_gallery('dataisbeautiful')

## End(Not run)
get_reddit_image

Description
Get sub-reddit gallery image details and metadata

Usage
get_reddit_image(subreddit, id, ...)

Arguments
subreddit  A character string containing the name of a sub-reddit.
id        An image ID or object of class “imgur_image” or “imgur_gallery_image”.
...       Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details
Get sub-reddit gallery image details and metadata. This is basically the same as get_gallery_image or get_image.

Value
An object of class “imgur_gallery_image”.

Author(s)
Thomas J. Leeper

See Also
get_reddit_gallery
get_gallery_image
get_image

Examples
## not run:
g <- get_reddit_gallery('dataisbeautiful')
get_reddit_image('dataisbeautiful', g[[1]])
get_image(g[[1]])

## End(Not run)
get_replies

Description

Get replies

Usage

get_replies(only_new = FALSE, ...)

Arguments

only_new  A logical indicating whether only new notifications should be returned. Default is FALSE.
...

Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

Details

Retrieves replies.

Value

An object of class “imgur_notification”.

Author(s)

Thomas J. Leeper

See Also

get_notifications

Examples

## Not run:
tkn <- imgur_login()
get_replies(only_new = TRUE, token = tkn)

## End(Not run)
Description

Basic workhorse HTTP method functions for executing Imgur.com API calls

Usage

`imgurGET(endpoint, base_url = "https://api.imgur.com/3/", key = "1babd0decbb90f2", token = NULL, ...)`

Arguments

- `endpoint`: A named API endpoint.
- `base_url`: The base URL for the API (by default the v3 API URL).
- `key`: An API key (by default, the imguR package API key). The API key for version of imguR < 1.0.0 was “4feb29d00face5b1b9daee536e15c373”. Either key or token must be specified.
- `token`: Optionally, an OAuth2.0 token (returned by `imgur_login`). Either key or token must be specified.
- `...`: Other arguments passed to analogous HTTP method functions from httr.

Details

These are the basic workhorse functions for the package, which users should not need to access directly (except perhaps for debugging).

Value

Generally, a list. For `imgurDELETE`, an empty character string.

Author(s)

Thomas J. Leeper

See Also

`Authentication`
imgur

Initiate an Imgur-enabled Graphics Device

Description

Open a specified graphics device function and configure it to upload to Imgur.

Usage

```r
imgur(device = png, file = NULL, title = NULL, description = NULL,
       album = NULL, name = NULL, key = NULL, token = NULL, ...)
```

Arguments

- `device`: A graphics device function. Default is `png`.
- `file`: Name of the file to be uploaded to imgur.com.
- `title`: Optionally, an image title.
- `description`: Optionally, an image description.
- `album`: Optionally, an Imgur album ID.
- `name`: Optionally, a file name, otherwise the `file` filename is used.
- `key`: An API key. If `NULL`, the default imguR package API key is used.
- `token`: Optionally, an OAuth2.0 token (returned by `imgur_login`). If `NULL`, the API key used instead.
- `...`: Other arguments passed to the specified device function.

Details

This package is a wrapper for the `pdf` function. A temporary file is created by `pdf`, acting as normal graphical device. After plotting, when `dev.off` is called, the file is closed, uploaded to imguR and a URL of the files location is returned. In the case of multipaged pdfs being uploaded, imguR automatically concatenates them into a single image.

Value

An object of class “imgur_device” to be passed to `imgur_off`.

Author(s)

Thomas J. Leeper, based on code by Aaron Statham.

See Also

`imgur_off`, `upload_image`
**imgur_login**

### Examples

```r
## Not run:
i <- imgur('png')
hist(rnorm(20))
imgur_off(i)

## End(Not run)
```

---

**imgur_login**  
*Login to Imgur*

### Description

Complete an interactive login to Imgur to generate an OAuth2.0 token.

### Usage

```r
imgur_login(client_id = "1babd0decbb90f2",
    secret = "0eed15f8e3662c20d7ff95a6285326640aae5a",
    cache = TRUE)
```

### Arguments

- `client_id`: A graphics device function. Default is `png`.
- `secret`: Name of the file to be uploaded to imgur.com.
- `cache`: A logical indicating whether to cache the OAuth2.0 token for use later on. If `TRUE`, OAuth2.0 token is automatically cached in a file called ".httr-oauth" in the working directory.

### Details

Most operations in the package can be performed anonymously using the default package API key. In order to perform most account-specific operations it is necessary to use OAuth authentication, which can only be performed interactively via a web browser. `imgur_login` returns an OAuth2.0 reference class object that can be passed as the `token` argument to all other functions in the package.

The OAuth token is moderately long lived; the response object contains a timestamp indicating its expiration time. The token can be subsequently refreshed using the `refresh()` method.

If `cache = TRUE`, the OAuth2.0 token is automatically cached in a file called ".httr-oauth" in the working directory. This token will then be loaded automatically the next time `imgur_login` is called (even in a new R session). To override an existing cache, set `cache = FALSE`.

### Value

An OAuth2.0 reference class object, returned by `oauth2.0_token`. 
Author(s)

Thomas J. Leeper

Examples

## not run:
# login interactively
tkn <- imgur_login()
# use token in the `imgur` device
i <- imgur('png', token = tkn)
hist(rnorm(20))
imgur_off(i)

# reload previous token
rm(tkn)
# OAuth token is automatically stored by
# `imgur_login` if `cache = TRUE` (the default)
# It is reloaded by calling `imgur_login`
tkn <- imgur_login()
token$refresh() # refresh token

## end(not run)

---

imgur_off

Close an Imgur-linked plotting device and upload image

Description

This function closes the specified graphics device uploads the plot(s) to Imgur.

Usage

`imgur_off(obj, ...)`

Arguments

- `obj` An object of class “imgur_device”.
- `...` Additional arguments passed to `upload_image`.

Details

This function takes the output object from `imgur` as its first argument, wraps `dev.off` to close the graphics device opened by `imgur`, and uploads the device’s image to Imgur via `upload_image`. If `dev.off` has already closed the device, that step is skipped to prevent errors.

Value

The response from `upload_image`.
Author(s)
Thomas J. Leeper, based on code by Aaron Statham.

See Also
imgur
upload_image

Examples
```r
## not run:
i <- imgur('png')
hist(rnorm(20))
imgur_off(i)

## end(not run)
```

mark_notification Mark notification read

Description
Mark a notification as read

Usage
mark_notification(id, ...)

Arguments
- `id` A notification ID.
- `...` Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

Details
Marks a named notification as read so that it does not show in future `get_notifications` results.

Value
An object of class “imgur_basic”.

Author(s)
Thomas J. Leeper

See Also
get_notifications
Description

View memes and meme templates

Usage

```r
default_memes(...)
get_memes_gallery(sort = 'viral', page = 0, window = NULL, ...)
get_meme(id, ...)
```

Arguments

- `sort` One of “viral” (the default), “time”, or “top”.
- `page` A page number of results between 0 and 50.
- `window` One of “day”, “week”, “month”, “year”, or “all”.
- `id` An image ID, perhaps returned by `get_memes_gallery`.
- `...` Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

`default_memes` returns a list of default memes available on Imgur. `get_memes_gallery` returns a list of trending memes. `get_meme` returns a specific meme image.

Value

A list of objects of class “imgur_image” or “imgur_gallery_image”.

Author(s)

Thomas J. Leeper

Examples

```r
## Not run:
default_memes()
def get-memes_gallery('viral', page = 1)

## End(Not run)
```
Description

Plot an “imgur_image” object

Usage

## S3 method for class 'imgur_image'
plot(x, ...)
## S3 method for class 'imgur_gallery_image'
plot(x, ...)

Arguments

x

An object of class “imgur_image” or “imgur_gallery_image”.

... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

Plot an object of class “imgur_image” or “imgur_gallery_image” in an R graphics window.

Value

Invisibly, an object of class “imgur_image” or “imgur_gallery_image”.

Author(s)

Thomas J. Leeper

See Also

get_image

Examples

## Not run:
plot(get_image("vSXFU46"))

## End(Not run)
Description

Check user- and package-level rate limits

Usage

rate_limit(...)

Arguments

... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

This function returns a list detailing user- and package-level API rate limits. All calls to the API deduct one unit from rate limits, except for upload_image, which deducts 10. Rate limits are lower for anonymous requests than for authenticated requests. Limits exist at the user level and at the package level. The user-level rate limit are automatically reset periodically. The response object documents when that limit will be reset.

Rate limits are automatically checked during each request. Options can be set for low remaining usage warnings. By default (when the package is loaded) those options are: options(imgur_user_rate_warning = 20) and options(imgur_client_rate_warning = 100).

Value

An object of class “imgur_basic”.

Author(s)

Thomas J. Leeper

Examples

```r
## Not run:
# rate limit for anonymous requests
rate_limit()

# rate limit for authenticated requests
tkn <- imgur_login()
rate_limit(token = tkn)
## End(Not run)
```
remove_gallery_image  Remove gallery image

Description
Remove images from the gallery

Usage
remove_gallery_image(id, ...)

Arguments
id  An image ID or an object of class “imgur_image” to add to the gallery.
... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

Details
This function removes the specified image from the gallery.

Value
An object of class “imgur_basic”.

Author(s)
Thomas J. Leeper

See Also
add_gallery_images

report_comment  Report a comment

Description
Report a comment as inappropriate.

Usage
report_comment(comment, ...)

Arguments

comment  A comment ID, perhaps returned by `create_comment` or `get_comment`.

... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

Reports a comment as inappropriate.

Value

An object of class “imgur_basic”.

Author(s)

Thomas J. Leeper

See Also

`get_comment`

`vote_comment`

---

`report_gallery_image`  Report a gallery image

Description

Report a gallery image as inappropriate.

Usage

`report_gallery_image(id, ...)`

Arguments

id  An image ID.

... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

Details

Reports a gallery image as inappropriate.

Value

An object of class “imgur_basic”.

Description

Report a user for inappropriate behavior

Usage

report_sender(username, ...)

Arguments

- `username`: A character string containing an Imgur username.
- `...`: Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token), which is required.

Details

Report an Imgur user for inappropriate behavior.

Value

An object of class “imgur_basic”.

Examples

```r
## Not run:
tkn <- imgur_login()
block_sender("someuser", token = tkn)
## End(Not run)
```
search_gallery

**Description**

Search Imgur public gallery

**Usage**

search_gallery(query, sort = 'time', page = 1, ...)

**Arguments**

- **query**: A character string containing a search term.
- **sort**: One of “time” or “top”.
- **page**: A page number.
- **...**: Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).

**Details**

Search the gallery and return gallery images and associated metadata.

**Value**

An object of class “imgur_gallery_album”.

**Author(s)**

Thomas J. Leeper

**See Also**

get_gallery
get_gallery_image
get_random_images

**Examples**

```r
## Not run:
search_gallery(query = 'meme')

## End(Not run)
```
**update_album**  
*Update album*

**Description**
Update album metadata

**Usage**

```
update_album(album, id = NULL,
             title = NULL, description = NULL,
             privacy = NULL, layout = NULL, cover_id = NULL, ...)
```

**Arguments**

- **album**
  An album ID or object of class “imgur_album”. If no token is supplied, `album` must be a deletehash from the original `create_album` response.
- **id**
  A vector of one or more image IDs to add to the album or a list of objects of class “imgur_image”.
- **title**
  Optionally, an album title.
- **description**
  Optionally, an album description.
- **privacy**
  The privacy for the album. One of “public”, “hidden”, or “secret”.
- **layout**
  The layout for the album. One of “blog”, “grid”, “horizontal”, or “vertical”.
- **cover_id**
  An image ID or an object of class “imgur_image”.
- …
  Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token) or `key` (an API key, used by default).

**Details**
Updates the images or metadata of an album. All options are optional. To anonymously update an album, `album` must be the deletehash value from `create_album`.

**Value**

An object of class “imgur_basic”.

**Author(s)**

Thomas J. Leeper

**See Also**

- `create_album`
- `add_album_images`
- `remove_album_images`
- `set_album_images`
Examples

```r
## Not run:
# update using a deletehash, anonymously
i <- imgur()
hist(rnorm(20))
img <- imgur_off(i)
a1 <-
create_album(id = img,
    title = 'My first Imgur album',
    description = 'A simple album',
    privacy = 'hidden',
    layout = 'grid',
    cover_id = img)
update_album(a1$deletehash, title = 'My second Imgur album')

# update using an OAuth token
tkn <- imgur_login()
a2 <-
create_album(id = img,
    title = 'My first Imgur album',
    description = 'A simple album',
    privacy = 'hidden',
    layout = 'grid',
    cover_id = img,
    token = tkn)
update_album(a2, title = 'My second Imgur album')

## End(Not run)
```

---

update_image | Update image

Description

Update image metadata

Usage

```r
update_image(id, title = NULL, description = NULL, ...)
```

Arguments

- `id` An image ID or object of class "imgur_image". If no token is supplied, id must be a deletehash from the original `upload_image` response.
- `title` Optionally, an image title.
- `description` Optionally, an image description.
- `...` Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token) or key (an API key, used by default).
Details

Update the title or description of an already-uploaded image.

Value

An object of class “imgur_basic”.

Author(s)

Thomas J. Leeper

See Also

upload_image
delete_image

Examples

```r
## not run:
# using a deletehash, anonymously
u <- upload_image(file = 'file.png')
update_image(u$deletehash, title = 'New Title')

# using an OAuth token
tkn <- imgur_login()
u <- upload_image(file = 'file.png', token = tkn)
update_image(u, title = 'New Title', token = tkn)

## end(not run)
```

upload_image  
Upload an image file to Imgur

Description

Uploads a named image file to Imgur, optionally to a specified album.

Usage

```r
upload_image(file, title = NULL, description = NULL, album = NULL,
        name = NULL, type = 'file', ...)
imgur_upload(file, title = NULL, description = NULL, album = NULL,
        name = NULL, type = 'file', ...)
```
Arguments

- **file**: Name of the file to be uploaded to imgur.com
- **title**: Optionally, an image title.
- **description**: Optionally, an image description.
- **album**: Optionally, an Imgur album ID.
- **name**: Optionally, a file name, otherwise the `file` filename is used.
- **type**: Should be “file”.
- **...**: Other arguments passed to HTTP request functions, for example: `token` (an OAuth2.0 token) or `key` (an API key, used by default).

Details

`upload_image` allows users to upload a named file to Imgur. `imgur_upload` is an alias that can serve as a drop-in replacement for the `imgur_upload` function in the knitr package.

Value

A list parsed from the imguR JSON response after uploading the plot (see the imgur API for details).

Author(s)

Thomas J. Leeper, adapted from code by Aaron Statham (`imgurupload` from imguR v0.1) and Yihue Xie (knitr::imgur_upload).

References

[https://api.imgur.com/endpoints/image#image-upload](https://api.imgur.com/endpoints/image#image-upload)

See Also

`imguR dev.off`

Examples

```r
## not run:
pdf(tmpfile <- tempfile())
hist(rnorm(20))
dev.off()
upload_image(tmpfile)

## End(Not run)
```
Description

Vote a comment up or down.

Usage

vote_comment(comment, vote = 'up', ...)

Arguments

comment A comment ID, perhaps returned by create_comment.
vote A character string indicating the direction of the vote, one of “up” or “down”.
... Other arguments passed to HTTP request functions, for example: token (an OAuth2.0 token), which is required.

Details

Votes a comment up or down.

Value

An object of class “imgur_basic”

Author(s)

Thomas J. Leeper

See Also

get_comment
report_comment
Index

*Topic package
  imguR-package, 3

account, 3
account_album_count (account_albums), 4
account_albums, 4
account_comment_count (account_comments), 5
account_comments, 5
account_favorites, 6
account_gallery, 7
account_image_count (account_images), 8
account_images, 8
account_settings, 9
account_verified (account), 3
add_album_images, 10, 15, 47
add_gallery_images, 11, 43
Authentication, 12, 35

block_sender, 13

create_album, 11, 14, 18, 47
create_comment, 15, 18, 19, 24, 44, 51
create_message, 16, 19, 25

default_memes (memes), 40
delete_album, 17
delete_comment, 16, 18, 24
delete_conversation, 17, 19, 19, 25
delete_image, 20, 49
dev.off, 50

favorite_album, 18, 21
favorite_image, 22

gallery_album, 11, 23, 27
gallery_images (get_album), 23
gallery_images (get_album), 23
gallery_images (get_album), 23
gallery_images (get_album), 23
gallery_images (get_album), 23
gallery_images (get_album), 23
gallery_images (get_album), 23
gallery_images (get_album), 23
gallery_images (get_album), 23
gallery_images (get_album), 23

get_image, 11, 28, 29, 33, 41
get_meme (memes), 40
get_memes_gallery (memes), 40
get_notifications, 30, 30, 34, 39
get_random_images, 26, 31, 32, 46
gallery_album, 11, 33
gallery_image, 26, 28, 31, 33, 46
search_gallery, 26–28, 31, 32, 46
send_verification(account), 3
set_album_images, 15, 47
set_album_images(add_album_images), 10
update_album, 11, 15, 18, 47
update_image, 20, 48
upload_image, 17, 20, 36, 38, 39, 42, 48, 49, 49
vote_comment, 44, 51