

# Package ‘imguR’

October 13, 2022

**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

---

imguR-package

*Imgur API Client Package for R*

---

## 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](#)

---

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).

**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**

```
## 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	<i>Retrieve user albums</i>
----------------	-----------------------------

---

**Description**

Retrieve details of a user's albums

**Usage**

```
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).

**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**

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

## End(Not run)
```

---

account_comments	<i>Retrieve user albums</i>
------------------	-----------------------------

---

**Description**

Retrieve details of a user's albums

**Usage**

```
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.

**Value**

For account\_comments, an object of class “imgur\_comment”. For account\_comment\_count, a number.

**Author(s)**

Thomas J. Leeper

**Examples**

```
## Not run:  
account_comments('thosjleeper')  
  
## End(Not run)
```

---

account\_favorites      *Retrieve user favorites*

---

**Description**

Retrieve details of a user’s favorite images

**Usage**

```
account_favorites(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**

account\_favorites 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 account\_favorites, an object of class “imgur\_gallery\_image” or “imgur\_gallery\_album”.

**Author(s)**

Thomas J. Leeper

**Examples**

```
## Not run:
account_favorites('thosjleeper')

## End(Not run)
```

---

account_gallery	<i>User gallery profile</i>
-----------------	-----------------------------

---

**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	<i>Retrieve user images</i>
----------------	-----------------------------

---

### 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)
```



---

account_settings	<i>Retrieve and modify account settings</i>
------------------	---

---

### Description

Retrieve and modify account settings

### Usage

```
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

## Examples

```
## 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)
```

---

add_album_images	<i>Manage album images</i>
------------------	----------------------------

---

## Description

Add images to or remove images from an album

## Usage

```
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”.

**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**

<code>album</code>	An album ID or object of class “imgur_album” to add to the gallery.
<code>id</code>	An image ID or an object of class “imgur_image” to add to the gallery.
<code>title</code>	A title for the image
<code>...</code>	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](#)

**Description**

Description of Imgur API authentication options

**Details**

All calls to the Imgur.com API must be authenticated. `imgur` provides two authentication methods: anonymous API key authentication (the default) and OAuth2.0 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 `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**

```
## 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:
```

```
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

**Examples**

```
## Not run:
tkn <- imgur_login()
block_sender("thosjleeper", token = tkn)

## End(Not run)
```

---

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 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

**See Also**

[update\\_album](#)  
[add\\_album\\_images](#)  
[remove\\_album\\_images](#)  
[set\\_album\\_images](#)

**Examples**

```

## 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	<i>Create a comments</i>
----------------	--------------------------

---

**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 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 `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**

```
## 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)
```

---

<code>create_message</code>	<i>Create a comments</i>
-----------------------------	--------------------------

---

**Description**

Create a comment on an image

**Usage**

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

**Arguments**

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

**Details**

Send a message to another Imgur user.



**Value**

An object of class “imgur\_basic”.

**Author(s)**

Thomas J. Leeper

**See Also**

[get\\_conversations](#)

[delete\\_conversation](#)

**Examples**

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

## End(Not run)
```

---

delete\_album

*Delete album*

---

**Description**

Delete an album

**Usage**

```
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 <a href="#">upload_image</a> 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”.

**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)  
delete_album(a[[1]], token = tkn)  
  
## End(Not run)
```

---

delete_comment	<i>Delete a comment</i>
----------------	-------------------------

---

**Description**

Delete a comment from an image

**Usage**

```
delete_comment(comment, ...)
```

**Arguments**

comment	A comment ID, perhaps returned by <a href="#">create_comment</a> .
...	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

**See Also**

[get\\_comment](#)  
[create\\_comment](#)

**Examples**

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

```
delete_comment(cmt, token = tkn)
```

```
## End(Not run)
```

---

delete\_conversation    *Delete conversation*

---

**Description**

Delete a conversation

**Usage**

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

**Arguments**

id	A conversation ID, possibly returned by <a href="#">get_conversations</a> .
...	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	<i>Delete image</i>
--------------	---------------------

---

**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 <a href="#">upload_image</a> 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**

```
## 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)
```

---

favorite_album	<i>Favorite album</i>
----------------	-----------------------

---

**Description**

Favorite an album

**Usage**

```
favorite_album(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**

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	<i>Favorite image</i>
----------------	-----------------------

---

**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)
```

---

`get_album`*Get album*

---

**Description**

Get album details and metadata, or images.

**Usage**

```
get_album(album, ...)
```

```
get_album_images(album, ...)
```

**Arguments**

<code>album</code>	An album ID or object of class “imgur_album”.
<code>...</code>	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	<i>Get a comment</i>
-------------	----------------------

---

### Description

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

### Usage

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

### Arguments

comment	A comment ID, perhaps returned by <a href="#">create_comment</a> .
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

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

---

`get_gallery`*Get gallery*

---

**Description**

Get gallery details and metadata

**Usage**

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

**Arguments**

<code>section</code>	One of “hot” (the default), “top”, or “user”.
<code>sort</code>	One of “viral” (the default) or “time”.
<code>page</code>	A page number of results between 0 and 50.
<code>window</code>	If <code>section == 'top'</code> , one of “day” (the default), “week”, “month”, “year”, or “all”.
<code>showViral</code>	If <code>section == 'user'</code> , a logical indicating whether show viral images.
<code>...</code>	Other arguments passed to HTTP request functions, for example: <code>token</code> (an OAuth2.0 token) or <code>key</code> (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](#)

**Examples**

```
## Not run:
get_gallery()

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

## End(Not run)
```

---

get_gallery_album	<i>Get gallery album</i>
-------------------	--------------------------

---

**Description**

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

**Usage**

```
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

```
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

```
## Not run:  
s <- search_gallery(query = 'meme')  
get_gallery_image(s[[1]])  
  
## End(Not run)
```

---

`get_image`*Get image*

---

**Description**

Get image details and metadata

**Usage**

```
get_image(id, ...)
```

**Arguments**

<code>id</code>	An image ID or object of class “imgur_image”.
<code>...</code>	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("vSXfU46")  
  
## End(Not run)
```

---

get\_notifications      *Get notifications*

---

**Description**

Get a specific notification or all notifications

**Usage**

```
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**

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

---

get_random_images	<i>Get random images</i>
-------------------	--------------------------

---

**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**

```
## 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	<i>Get reddit image</i>
------------------	-------------------------

---

### 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`*Get replies*

---

**Description**

Get replies

**Usage**

```
get_replies(only_new = FALSE, ...)
```

**Arguments**

<code>only_new</code>	A logical indicating whether only new notifications should be returned. Default is FALSE.
<code>...</code>	Other arguments passed to HTTP request functions, for example: <code>token</code> (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)
```

---

http *Workhorse HTTP functions*

---

**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 <code>imguR</code> package API key). The API key for version of <code>imguR</code> < 1.0.0 was "4feb29d00face5bc1b9dae536e15c373". Either key or token must be specified.
token	Optionally, an OAuth2.0 token (returned by <code>imgur_login</code> ). Either key or token must be specified.
...	Other arguments passed to analogous HTTP method functions from <code>httr</code> .

**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**

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

**Arguments**

<code>device</code>	A graphics device function. Default is <code>png</code> .
<code>file</code>	Name of the file to be uploaded to <code>imgur.com</code>
<code>title</code>	Optionally, an image title.
<code>description</code>	Optionally, an image description.
<code>album</code>	Optionally, an Imgur album ID.
<code>name</code>	Optionally, a file name, otherwise the file filename is used.
<code>key</code>	An API key. If <code>NULL</code> , the default <code>imguR</code> package API key is used.
<code>token</code>	Optionally, an OAuth2.0 token (returned by <a href="#">imgur_login</a> ). If <code>NULL</code> , the API key used instead.
<code>...</code>	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](#)

## Examples

```
## 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

```
imgur_login(client_id = "1babd0decbb90f2",  
            secret = "06eed15f8e3662c20d7ff95a62853266400aae5a",  
            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**

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

---

mark_notification	<i>Mark notification read</i>
-------------------	-------------------------------

---

**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](#)

---

memes

*Mememes*

---

## Description

View memes and meme templates

## Usage

```
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 <code>get_memes_gallery</code> .
...	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

```
## Not run:  
default_memes()  
get_memes_gallery('viral', page = 1)  
  
## End(Not run)
```



---

plot

*Plot Imgur image*

---

## 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)
```

---

`rate_limit`*API rate limits*

---

### 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

```
## 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”.

**Author(s)**

Thomas J. Leeper

---

report_sender	<i>Report a user</i>
---------------	----------------------

---

**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".

**Author(s)**

Thomas J. Leeper

**Examples**

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

---

search_gallery	<i>Search gallery</i>
----------------	-----------------------

---

**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**

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

---

update_album	<i>Update album</i>
--------------	---------------------

---

### 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 <a href="#">create_album</a> 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**

```
## 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**

```
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 <a href="#">upload_image</a> 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**

```
## Not run:  
# using a deletehash, anonymously  
u <- upload_image('file.png')  
update_image(u$deletehash, title = 'New Title')  
  
# using an OAuth token  
tkn <- imgur_login()  
u <- upload_image('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**

```
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>

**See Also**

[imguR dev.off](#)

**Examples**

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

## End(Not run)
```

---

vote_comment	<i>Vote on a comment</i>
--------------	--------------------------

---

**Description**

Vote a comment up or down.

**Usage**

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

**Arguments**

comment	A comment ID, perhaps returned by <a href="#">create_comment</a> .
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

- \* **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
  
- get\_album, 11, 23, 27
- get\_album\_images (get\_album), 23
- get\_comment, 16, 19, 24, 44, 51
- get\_conversations, 17, 19, 25, 25
- get\_gallery, 26, 27, 28, 31, 32, 46
- get\_gallery\_album, 27
  
- get\_gallery\_image, 26, 28, 31, 33, 46
- 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
- get\_reddit\_gallery, 32, 33
- get\_reddit\_image, 33
- get\_replies, 34
  
- http, 35
  
- imguR, 50
- imguR (imgur), 36
- imgur, 36, 38, 39
- imguR-package, 3
- imgur\_login, 12, 35, 36, 37
- imgur\_off, 36, 38
- imgur\_upload (upload\_image), 49
- imgurDELETE (http), 35
- imgurGET (http), 35
- imgurPOST (http), 35
- imgurPUT (http), 35
- imguRupload (upload\_image), 49
  
- mark\_notification, 30, 39
- memes, 40
  
- oauth2.0\_token, 37
  
- plot, 41
- plot.imgur\_image, 29
  
- rate\_limit, 42
- remove\_album\_images, 15, 47
- remove\_album\_images (add\_album\_images), 10
- remove\_gallery\_image, 11, 43
- report\_comment, 43, 51
- report\_gallery\_image, 44
- report\_sender, 45

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](#)