

# Package: rpinterest (via r-universe)

June 6, 2024

**Type** Package

**Title** Access Pinterest API

**Version** 0.4.0

**Description** Get information (boards, pins and users) from the Pinterest <<http://www.pinterest.com>> API.

**License** MIT + file LICENSE

**Imports** attempt, curl, data.table, httr, jsonlite, base64enc, stats

**Suggests** ggplot2, knitr, rmarkdown, testthat

**Encoding** UTF-8

**LazyData** TRUE

**RoxygenNote** 6.1.0

**VignetteBuilder** knitr

**Repository** <https://colinfay.r-universe.dev>

**RemoteUrl** <https://github.com/colinfay/rpinterest>

**RemoteRef** HEAD

**RemoteSha** 066e3d9cb13eab926afdedb59b562f4c8aa90cf1

## Contents

create_board . . . . .	2
create_pin . . . . .	2
get_board_pins_by_id . . . . .	3
get_board_spec_by_id . . . . .	4
get_logged_user . . . . .	5
get_pin_spec_by_id . . . . .	5
get_user_spec_by_id . . . . .	6
ggplot_to_pinterest . . . . .	7
pinterest_token . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

create_board	<i>Create, delete, or edit a Board</i>
--------------	--

---

**Description**

Create, delete, or edit a Board

**Usage**

```
create_board(name, description, token)
```

```
delete_board(board, token)
```

```
edit_board(board, name = NULL, description = NULL, token)
```

**Arguments**

name	name of the new board for creation, new name to give for edit.
description	description of the board?
token	a character string with an access token generated with <code>'pinterest_token()'</code>
board	a board name.

**Value**

Either a data.frame, or an error.

**Examples**

```
## Not run:  
create_board("testboard", "from rpinterest", token)  
delete_board("colinfay/testboard", token)  
edit_board("colinfay/testboard",description = "A new desc", token = token)  
  
## End(Not run)
```

---

create_pin	<i>Create, delete, or edit a Pin</i>
------------	--------------------------------------

---

**Description**

Create, delete, or edit a Pin

**Usage**

```
create_pin(file, board, note, link = NULL, token)

delete_pin(id, token)

edit_pin(id, board = NULL, note = NULL, link = NULL, token)
```

**Arguments**

file	the path to the file
board	name of the board (required)
note	a description (required)
link	an url (optional)
token	a character string with an access token generated with 'pinterest_token()'
id	a character string with an ID.

**Value**

Either a data.frame, or an error.

**Examples**

```
## Not run:
create_pin("~/Pictures/graph.png", "my_board", "a graph", token = token)
delete_pin("42080577756375754", token)
edit_pin("42080577744990466",note = "run run run", token = token)

## End(Not run)
```

---

`get_board_pins_by_id` *Retrieve the Pins on a Board*

---

**Description**

Get information about all the pins on a pinterest board using the board ID or user/name.

**Usage**

```
get_board_pins_by_id(id, token)

get_board_pins_by_name(user, board, token)
```

**Arguments**

id	a character string with an ID.
token	a character string with an access token generated with 'pinterest_token()'
user	a character string with a user name.
board	a character string with a board name.

**Value**

Either a data.frame, or an error.

**Examples**

```
## Not run:  
get_board_pins_by_id(id = "42080646457333782", token = token)  
get_board_pins_by_name(user = "colinfay", board = "violet-mon-amour", token = token)  
  
## End(Not run)
```

---

get\_board\_spec\_by\_id *Retrieve information about a Board*

---

**Description**

Get information about a pinterest board using the board ID or the name/user combination.

**Usage**

```
get_board_spec_by_id(id, token)  
  
get_board_spec_by_name(user, board, token)
```

**Arguments**

id	a character string with a board ID.
token	a character string with an access token generated with 'pinterest_token()'
user	a character string with a user name.
board	a character string with a board name.

**Value**

Either a data.frame, or an error.

**Examples**

```
## Not run:  
get_board_spec_by_id(id = "42080646457333782", token = token)  
get_board_spec_by_name(user = "colinfay", board = "blanc-mon-amour", token = token)  
  
## End(Not run)
```

---

get_logged_user	<i>Get Information about the logged user</i>
-----------------	--

---

**Description**

Get user information about their account or about their boards.

**Usage**

```
get_logged_user(token)
```

```
get_logged_user_boards(token)
```

```
get_logged_user_boards_suggestions(count = 25, token)
```

**Arguments**

token	a character string with an access token generated with 'pinterest_token()'
count	for suggestions, the max number of suggestion to retrieve.

**Value**

Either a data.frame, or an error.

**Examples**

```
## Not run:  
get_logged_user(token)  
get_logged_user_boards(token)  
get_logged_user_boards_suggestions(token = token)  
  
## End(Not run)
```

---

get_pin_spec_by_id	<i>Return information about a Pin</i>
--------------------	---------------------------------------

---

**Description**

Get information about a pinterest pin using the pin ID.

**Usage**

```
get_pin_spec_by_id(id, token)
```

**Arguments**

id                    a character string with a pin ID.  
token                a character string with an access token generated with 'pinterest\_token()'

**Value**

Either a data.frame, or an error.

**Examples**

```
## Not run:  
get_pin_spec_by_id(id = "42080577745042298", token = token)  
  
## End(Not run)
```

---

get\_user\_spec\_by\_id    *Return a user's information*

---

**Description**

Get information about a Pinterest user using the ID or name.

**Usage**

```
get_user_spec_by_id(id, token)  
  
get_user_spec_by_name(user, token)
```

**Arguments**

id                    a character string with a user ID.  
token                a character string with an access token generated with 'pinterest\_token()'  
user                 a character string with a user name.

**Value**

Either a data.frame, or an error.

**Examples**

```
## Not run:  
get_user_spec_by_id(id = "42080715176677612", token = token)  
get_user_spec_by_name(user = "colinfay", token = token)  
  
## End(Not run)
```

---

ggplot\_to\_pinterest     *Send a ggplot plot to a Pinterest board*

---

**Description**

Send a ggplot plot to a Pinterest board

**Usage**

```
ggplot_to_pinterest(plot, board, note, link = NULL, token, ...)
```

**Arguments**

plot	a ggplot object
board	the name of the board
note	description of the image
link	a link to put under the image
token	a character string with an access token generated with 'pinterest_token()'
...	arguments to be passed to 'ggsave()'

**Value**

Either a data.frame, or an error.

**Examples**

```
## Not run:  
library(ggplot2)  
x <- ggplot(iris, aes(Sepal.Length, Sepal.Width)) + geom_point()  
ggplot_to_pinterest(x, "colinfay/gris-mon-amour", "test rpinterest", token = token)  
  
## End(Not run)
```

---

pinterest\_token     *Create a Pinterest API token*

---

**Description**

Create a Pinterest API token

**Usage**

```
pinterest_token(app, app_id, app_secret, verbose = TRUE,  
  callback = "https://colinfay.me/rpinterestcallback/")
```

**Arguments**

app	the name of your app
app_id	the ID of your app
app_secret	the secret of the app
verbose	wether of not to print a message about the auth process
callback	the callback uri for your app

**Value**

an access token



# Index

create\_board, 2  
create\_pin, 2

delete\_board (create\_board), 2  
delete\_pin (create\_pin), 2

edit\_board (create\_board), 2  
edit\_pin (create\_pin), 2

get\_board\_pins\_by\_id, 3  
get\_board\_pins\_by\_name  
    (get\_board\_pins\_by\_id), 3  
get\_board\_spec\_by\_id, 4  
get\_board\_spec\_by\_name  
    (get\_board\_spec\_by\_id), 4  
get\_logged\_user, 5  
get\_logged\_user\_boards  
    (get\_logged\_user), 5  
get\_logged\_user\_boards\_suggestions  
    (get\_logged\_user), 5  
get\_pin\_spec\_by\_id, 5  
get\_user\_spec\_by\_id, 6  
get\_user\_spec\_by\_name  
    (get\_user\_spec\_by\_id), 6  
ggplot\_to\_pinterest, 7

pinterest\_token, 7