

# Package: pinboardr (via r-universe)

August 21, 2024

**Title** Get information from 'pinboard.in' API

**Version** 1.0.0

**Description** Full connectivity to the 'pinboard.in' service. Retrieve bookmarks, add new ones. Add, delete and rename tags, all from the convenience of your R session.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** htr, jsonlite, utils, stats

**Suggests** testthat (>= 2.1.0), spelling

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**URL** <https://github.com/RMHogervorst/pinboardr>

**BugReports** <https://github.com/RMHogervorst/pinboardr/issues>

**Language** en\_US

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

**RemoteUrl** <https://github.com/RMHogervorst/pinboardr>

**RemoteRef** HEAD

**RemoteSha** d4d145f97c054a09e8923c972acb924a084d4b70

## Contents

authentication . . . . .	2
pb_add_tag_column . . . . .	3
pb_get_note . . . . .	3
pb_get_notes_overview . . . . .	4
pb_last_update . . . . .	5
pb_posts_add . . . . .	5
pb_posts_all . . . . .	6
pb_posts_dates . . . . .	7

pb_posts_delete . . . . .	8
pb_posts_get . . . . .	9
pb_posts_recent . . . . .	10
pb_posts_suggest . . . . .	10
pb_tags_delete . . . . .	11
pb_tags_get . . . . .	12
pb_tags_rename . . . . .	12
pb_user_api_token . . . . .	13
pb_user_secret . . . . .	14
pinboard_formatting . . . . .	14

## Index 15

---

authentication	<i>How to setup your authentication</i>
----------------	---

---

### Description

All interactions with the 'pinboard.in' API (application programming interface) require authentication. This doc explains where to find your secrets and how to set up your computer so you can programmatically interact with pinboard.

### Details

You do need to have a pinboard account.

#### **Finding your secrets:**

Go to the [pinboard password page](#) and scroll down to API Token. It says something like: "this is your API token: username:NUMBERSANDLETTERS"

#### **.Renvironment file:**

I would recommend you add the token to your .renviron file, either locally or globally. A full introduction how these files work is beyond the scope of this doc but if you create a 'hidden' file named .Renviron in the project folder or in your home folder, R will read in the username and token and they will be available to pinboardr. Use `usethis::edit_r_environ()` or find the file manually.

Add to your .renviron file the following `PB_USERNAME=username PB_TOKEN="NUMBERSANDLETTERS"`  
Restart the session to make the changes active.

#### **Setting variables locally:**

This is not the recommended approach, but will work nevertheless. You can either:

- set the variables at the start of your session using `Sys.setenv(PB_USERNAME = "username", PB_TOKEN="NUMBERSANDLETTERS")`
- pass the username and token to every command: `pb_last_update(username=username, token=NUMBERSANDLETTERS)`

#### *Why is this not recommended?:*

This means your secrets, username and token are visible in your .rhistory, and if you save it in a script, it will be visible to anyone who opens your script. I think you don't want that, but my threatmodel is not your threatmodel.

---

pb_add_tag_column	<i>Extract one tag from tag columns into separate column.</i>
-------------------	---

---

**Description**

Convenience function to make working with the dataframe of bookmarks a bit easier.

**Usage**

```
pb_add_tag_column(dataframe, tag)
```

**Arguments**

dataframe	the dataframe of bookmarks
tag	tag you would like to extract into a new column

**Value**

a dataframe with one extra column, containing TRUE or FALSE for presence of the tag

**Examples**

```
## Not run:  
pb_add_tag_column(all_bookmars, "europe")  
  
## End(Not run)
```

---

pb_get_note	<i>Get one specific note</i>
-------------	------------------------------

---

**Description**

Get one specific note

**Usage**

```
pb_get_note(id = NULL, username = NULL, token = NULL)
```

**Arguments**

id	the id of a note
username	your username You probably don't need to supply this every time. See ?authentication
token	your token

**Value**

a list containing id, title, created\_at, updated\_at, length, text and hash

**Examples**

```
## Not run:  
pb_get_note(id=12345)  
  
## End(Not run)
```

---

`pb_get_notes_overview` Returns a data.frame of the user's notes

---

**Description**

Return a data.frame of notes meta information, id, hash, title, length, created\_at and updated\_at. When you know the id, you can use the other endpoint `pb_get_note`

**Usage**

```
pb_get_notes_overview(username = NULL, token = NULL)
```

**Arguments**

username	your username You probably don't need to supply this every time. See ?authentication
token	your token

**Value**

a data.frame with notes meta information

**Examples**

```
## Not run:  
pb_get_notes_overview()  
  
## End(Not run)
```

---

pb_last_update	<i>Find the last update time</i>
----------------	----------------------------------

---

**Description**

Returns the most recent time a bookmark was added, updated or deleted. Use this before calling posts\_all to see if the data has changed since the last fetch.

**Usage**

```
pb_last_update(as_datetime = FALSE, username = NULL, token = NULL)
```

**Arguments**

as_datetime	defaults to FALSE, if TRUE will turn the text value into "POSIXt" "POSIXt"
username	your username You probably don't need to supply this every time. See ?authentication
token	your token

**Value**

time in text format or as "POSIXt" "POSIXt"

**Examples**

```
## Not run:  
pb_last_update(as_datetime = TRUE)  
pb_last_update()  
  
## End(Not run)
```

---

pb_posts_add	<i>Add a bookmark</i>
--------------	-----------------------

---

**Description**

You must give a url and description, all other arguments are optional.

**Usage**

```
pb_posts_add(  
  url,  
  title,  
  description = NULL,  
  tags = NULL,  
  dt = NULL,  
  replace = NULL,
```

```

    public = NULL,
    toread = NULL,
    username = NULL,
    token = NULL
)

```

### Arguments

url	<b>REQUIRED</b> the URL of the item
title	<b>REQUIRED</b> Title of the item.
description	Description of the item
tags	vector of up to 100 tags
dt	creation time for this bookmark. Defaults to current time. Datestamps more than 10 minutes ahead of server time will be reset to current server time (UTC timestamp in this format: 2010-12-11T19:48:02Z)
replace	Replace any existing bookmark with this URL. Default is TRUE If set to FALSE, will throw an error if bookmark exists
public	TRUE/FALSE Make bookmark public. Default is "TRUE" unless user has enabled the "save all bookmarks as private" user setting, in which case default is "no"
toread	TRUE/FALSE Marks the bookmark as unread. Default is "FALSE"
username	your username You probably don't need to supply this every time. See ?authentication
token	your token

### Value

status in text 'done'

### Examples

```

## Not run:
pb_posts_add(url="https://example.com", title="an example website")

## End(Not run)

```

---

pb\_posts\_all

*Return all bookmarks in the user's account*

---

### Description

This API endpoint is rate limited to once every 5 minutes.

**Usage**

```
pb_posts_all(
  tags = NULL,
  start = 0L,
  results = NULL,
  fromdt = NULL,
  todt = NULL,
  meta = NULL,
  username = NULL,
  token = NULL
)
```

**Arguments**

tags	filter by up to three tags
start	int offset value (default is 0)
results	int number of results to return. Default is all
fromdt	datetime return only bookmarks created after this time
todt	datetime return only bookmarks created before this time
meta	int include a change detection signature for each bookmark
username	your username You probably don't need to supply this every time. See ?authentication
token	your token

**Value**

a dataframe of all posts of pinboard format see ?pinboard\_formatting

**Examples**

```
## Not run:
pb_posts_all(tags = "inspiration", results = 15)

## End(Not run)
```

---

pb_posts_dates	<i>Posts per date</i>
----------------	-----------------------

---

**Description**

Returns a dataframe of dates with the number of posts at each date. Use this to see all posts per date or posts per tag per date.

**Usage**

```
pb_posts_dates(tags = NULL, username = NULL, token = NULL)
```

**Arguments**

tags	filter by up to three tags
username	your username You probably don't need to supply this every time. See ?authentication
token	your token

**Value**

a data.frame with dates and counts

**Examples**

```
## Not run:
pb_posts_dates(tags = "europe")

## End(Not run)
```

---

pb_posts_delete	<i>Delete a bookmark</i>
-----------------	--------------------------

---

**Description**

Delete a bookmark from pinboard by url.

**Usage**

```
pb_posts_delete(url = NULL, username = NULL, token = NULL)
```

**Arguments**

url	<b>REQUIRED</b> url of bookmark to delete
username	your username You probably don't need to supply this every time. See ?authentication
token	your token

**Value**

status in text: "done", "item not found"

**Examples**

```
## Not run:
pb_posts_delete("https://example.com")

## End(Not run)
```



---

`pb_posts_get`*Get all posts on a day*

---

### Description

Returns one or more posts on a single day matching the arguments. If no date or url is given, date of most recent bookmark will be used.

### Usage

```
pb_posts_get(  
  tags = NULL,  
  dt = NULL,  
  url = NULL,  
  meta = NULL,  
  username = NULL,  
  token = NULL  
)
```

### Arguments

<code>tags</code>	filter by up to three tags
<code>dt</code>	return results bookmarked on this day (UTC date in this format: 2010-12-31).
<code>url</code>	return bookmark for this URL
<code>meta</code>	TRUE/FALSE include a change detection signature in a meta attribute
<code>username</code>	your username You probably don't need to supply this every time. See ?authentication
<code>token</code>	your token

### Value

a dataframe of all posts of pinboard format see ?pinboard\_formatting

### Examples

```
## Not run:  
pb_posts_get()  
  
## End(Not run)
```

---

pb\_posts\_recent      *Get most recent posts*

---

### Description

Returns a list of the user's most recent posts, filtered by tag.

### Usage

```
pb_posts_recent(tags = NULL, count = 15, username = NULL, token = NULL)
```

### Arguments

tags	tags filter by up to three tags
count	number of results to return default is 15 max is 100
username	your username You probably don't need to supply this every time. See ?authentication
token	your token

### Value

a dataframe of all recent of pinboard format see ?pinboard\_formatting

### Examples

```
## Not run:
pb_posts_recent(tags = "inspiration")

## End(Not run)
```

---

pb\_posts\_suggest      *Returns a list of popular tags and recommended tags for a given URL.*

---

### Description

Popular tags are tags used site-wide for the url; recommended tags are drawn from the user's own tags.

### Usage

```
pb_posts_suggest(url = NULL, username = NULL, token = NULL)
```

### Arguments

url	<b>REQUIRED</b> url to get suggestions for
username	your username You probably don't need to supply this every time. See ?authentication
token	your token

**Value**

a data.frame with popular and recommended tags

**Examples**

```
## Not run:  
pb_posts_suggest(url = "https://example.com")  
  
## End(Not run)
```

---

pb_tags_delete	<i>Delete an existing tag</i>
----------------	-------------------------------

---

**Description**

Notice that 'pinboard.in' does not tell you if a tag exists or not. It will return a 'done' unless something else goes wrong.

**Usage**

```
pb_tags_delete(tag, username = NULL, token = NULL)
```

**Arguments**

- tag           **REQUIRED** tag the delete
- username      your username You probably don't need to supply this every time. See ?authentication
- token         your token

**Value**

text confirmation

**Examples**

```
## Not run:  
pb_tags_delete(tag = "svg")  
  
## End(Not run)
```

---

pb\_tags\_get                      *Find all tags in use.*

---

### Description

Find all tags in use.

### Usage

```
pb_tags_get(username = NULL, token = NULL)
```

### Arguments

username	your username You probably don't need to supply this every time. See ?authentication
token	your token

### Value

a dataframe with all tags and their count

### Examples

```
## Not run:
pb_get_all_tags()

## End(Not run)
```

---

pb\_tags\_rename                      *Rename a tag*

---

### Description

Change the name of a tag. Notice that 'pinboard.in' does not tell you if a tag exists or not. It will return a 'done' unless something else goes wrong.

### Usage

```
pb_tags_rename(old, new = NULL, username = NULL, token = NULL)
```

### Arguments

old	<b>REQUIRED</b> old tag name
new	<b>REQUIRED</b> new tag name, if empty nothing will happen
username	your username You probably don't need to supply this every time. See ?authentication
token	your token

**Value**

text confirmation

**Examples**

```
## Not run:  
pb_tags_rename(old = "svg", new = "illustrations")  
  
## End(Not run)
```

---

pb_user_api_token	<i>Get your api token</i>
-------------------	---------------------------

---

**Description**

This is a silly endpoint and function, but I implemented it anyways. To reach this endpoint, you already are in possession of your token. And I will not implement a username and password authentication because that is inherently unsafe in this API. Be safe people!

**Usage**

```
pb_user_api_token(username = NULL, token = NULL)
```

**Arguments**

username	your username You probably don't need to supply this every time. See ?authentication
token	your token

**Value**

token in string format

**Examples**

```
## Not run:  
pb_user_api_token()  
  
## End(Not run)
```

---

pb\_user\_secret      *Returns the user's secret RSS key (for viewing private feeds)*

---

### Description

Get your secret RSS key for private feeds.

### Usage

```
pb_user_secret(username = NULL, token = NULL)
```

### Arguments

username      your username You probably don't need to supply this every time. See ?authentication  
 token          your token

### Value

the secret RSS key

### Examples

```
## Not run:
pb_user_secret()

## End(Not run)
```

---

pinboard\_formatting      *Pinboard format data.frame*

---

### Description

The pinboardr package returns the data in a format as close as possible to the data supplied by the API. Because 'pinboard.in' modeled their API on the original 'Delicious API' with the same unfortunate choices. See also the description on pinboard [posts\\_add docs](#).

### Details

I try to hide the complexity from you by translating but sometimes that gives us different names. I first give you the pinboardr name and than the name in the delicious and pinboard API:

- The 'title' of a bookmark is actually called 'description'.
- The 'description' of a bookmark is actually called 'extended'.
- The 'public' setting of bookmarks is called 'shared'.

# Index

authentication, [2](#)

pb\_add\_tag\_column, [3](#)

pb\_get\_note, [3](#)

pb\_get\_notes\_overview, [4](#)

pb\_last\_update, [5](#)

pb\_posts\_add, [5](#)

pb\_posts\_all, [6](#)

pb\_posts\_dates, [7](#)

pb\_posts\_delete, [8](#)

pb\_posts\_get, [9](#)

pb\_posts\_recent, [10](#)

pb\_posts\_suggest, [10](#)

pb\_tags\_delete, [11](#)

pb\_tags\_get, [12](#)

pb\_tags\_rename, [12](#)

pb\_user\_api\_token, [13](#)

pb\_user\_secret, [14](#)

pinboard\_formatting, [14](#)