

Package ‘bskyr’

January 9, 2024

Title Interact with 'Bluesky' Social

Version 0.1.2

Description Collect data from and make posts on 'Bluesky' Social via the Hypertext Transfer Protocol (HTTP) Application Programming Interface (API), as documented at <https://atproto.com/specs/xrpc>. This further supports broader queries to the Authenticated Transfer (AT) Protocol <https://atproto.com/> which 'Bluesky' Social relies on. Data is returned in a tidy format and posts can be made using a simple interface.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports cli, dplyr, fs, httr2, lubridate, mime, purrr, rlang, stringi, stringr, tibble, tidyr

Suggests httptest2, jsonlite, spelling, testthat (>= 3.0.0)

URL <https://github.com/christopherkenny/bskyr>,
<http://christophertkenny.com/bskyr/>

BugReports <https://github.com/christopherkenny/bskyr/issues>

Config/testthat/edition 3

Language en-US

Depends R (>= 4.1.0)

NeedsCompilation no

Author Christopher T. Kenny [aut, cre]
(<https://orcid.org/0000-0002-9386-6860>)

Maintainer Christopher T. Kenny <christopherkenny@fas.harvard.edu>

Repository CRAN

Date/Publication 2024-01-09 21:00:09 UTC

R topics documented:

bs_auth	3
bs_created_at	3
bs_create_record	4
bs_delete_record	5
bs_describe_repo	6
bs_get_actor_lists	7
bs_get_actor_suggestions	8
bs_get_author_feed	9
bs_get_blocked_lists	10
bs_get_blocks	11
bs_get_feed	12
bs_get_feeds	13
bs_get_feed_generator	14
bs_get_feed_generators	15
bs_get_feed_suggestions	16
bs_get_followers	17
bs_get_follows	19
bs_get_follow_suggestions	20
bs_get_likes	21
bs_get_muted_lists	22
bs_get_mutes	23
bs_get_notifications	24
bs_get_notification_count	25
bs_get_posts	26
bs_get_post_likes	27
bs_get_post_thread	28
bs_get_preferences	29
bs_get_profile	30
bs_get_record	31
bs_get_reposts	32
bs_get_timeline	33
bs_like	34
bs_list_records	35
bs_post	36
bs_repost	37
bs_resolve_handle	38
bs_search_actors	39
bs_search_posts	40
bs_upload_blob	41
bs_uri_to_url	42
pass	43
set_bluesky_pass	43
set_bluesky_user	44
user	45

bs_auth	<i>Authenticate a user</i>
---------	----------------------------

Description

Authenticate a user

Usage

```
bs_auth(user, pass)
```

Arguments

user	Character. User name to log in with.
pass	Character. App password to log in with.

Value

a list of authentication information

Lexicon references

[server/createSession.json \(2023-09-30\)](#)

Function introduced

v0.0.1 (2023-09-30)

Examples

```
bs_auth(user = get_bluesky_user(), pass = get_bluesky_pass())
```

bs_created_at	<i>Get current time in Bluesky format</i>
---------------	---

Description

Get current time in Bluesky format

Usage

```
bs_created_at()
```

Value

a length 1 character vector

Function introduced

v0.1.0 (2023-11-25)

Examples

```
bs_created_at()
```

bs_create_record	<i>Create a record in a repo</i>
------------------	----------------------------------

Description

Create a record in a repo

Usage

```
bs_create_record(  
  collection,  
  record,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

collection	Character, length 1. The NSID of the record collection.
record	List, length 1. Description of a record.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of record information

Lexicon references

[repo/createRecord.json \(2023-11-25\)](#)

Function introduced

v0.1.0 (2023-11-25)

Examples

```
# get info about a record
post_rcd <- bs_get_record('https://bsky.app/profile/bskyr.bsky.social/post/3kf2577exva2x')
# create a record, to like the post
like <- list(
  subject = list(
    uri = post_rcd$uri,
    cid = post_rcd$cid
  ),
  createdAt = bs_created_at()
)

bs_create_record(collection = 'app.bsky.feed.like', record = like)
```

bs_delete_record	<i>Delete a record in a repo</i>
------------------	----------------------------------

Description

Delete a record in a repo

Usage

```
bs_delete_record(
  collection,
  rkey,
  user = get_bluesky_user(),
  pass = get_bluesky_pass(),
  auth = bs_auth(user, pass)
)
```

Arguments

collection	Character, length 1. The NSID of the record collection.
rkey	Character, length 1. The CID of the version of the record. If not specified, then return the most recent version.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).

Value

an httr2 status code

Lexicon references

[repo/deleteRecord.json \(2023-11-25\)](#)

Function introduced

v0.1.0 (2023-11-25)

Examples

```
# get info about a record
post_rcd <- bs_get_record('https://bsky.app/profile/bskyr.bsky.social/post/3kf2577exva2x')
# create a record, to like the post
like <- list(
  subject = list(
    uri = post_rcd$uri,
    cid = post_rcd$cid
  ),
  createdAt = bs_created_at()
)

rec <- bs_create_record(collection = 'app.bsky.feed.like', record = like)
bs_delete_record(collection = 'app.bsky.feed.like',
  rkey = stringr::str_split_i(rec$uri, '/', i = 5))
```

bs_describe_repo *Describe a repo*

Description

Describe a repo

Usage

```
bs_describe_repo(
  repo,
  user = get_bluesky_user(),
  pass = get_bluesky_pass(),
  auth = bs_auth(user, pass),
  clean = TRUE
)
```

Arguments

repo	Character, length 1. The handle or DID of the repo.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of record information

Lexicon references

[repo/describeRepo.json \(2023-11-25\)](#)

Function introduced

v0.1.0 (2023-11-25)

Examples

```
bs_describe_repo('chriskenny.bsky.social')
```

`bs_get_actor_lists` *Get a list of lists that belong to an actor.*

Description

Get a list of lists that belong to an actor.

Usage

```
bs_get_actor_lists(  
  actor,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

actor	Character, length 1. name of 1 actor, such as 'chriskenny.bsky.social'
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of lists

Lexicon references

[graph/getLists.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_actor_lists('profmusgrave.bsky.social')
```

bs_get_actor_suggestions

Get a list of actors suggested for following

Description

Get a list of actors suggested for following

Usage

```
bs_get_actor_suggestions(  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```


Arguments

cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of suggested accounts to follow

Lexicon references

[actor/getSuggestions.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_actor_suggestions()
```

`bs_get_author_feed` *Retrieve posts on an actor's feed*

Description

Retrieve posts on an actor's feed

Usage

```
bs_get_author_feed(  
  actor,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

actor	Character, length 1. name of 1 actor, such as 'chriskenny.bsky.social'
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of posts

Lexicon references

[feed/getAuthorFeed.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_author_feed('chriskenny.bsky.social')
```

`bs_get_blocked_lists` *Retrieve a user's (self) muted lists*

Description

Retrieve a user's (self) muted lists

Usage

```
bs_get_blocked_lists(  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Maximum number to request.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of actors

Lexicon references

[graph/getListMutes.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_blocked_lists()
```

bs_get_blocks	<i>Retrieve user (self) blocks</i>
---------------	------------------------------------

Description

Retrieve user (self) blocks

Usage

```
bs_get_blocks(  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of blocked accounts

Lexicon references

[graph/getBlocks.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_blocks()
```

bs_get_feed

Build feed from user's feed generator

Description

Build feed from user's feed generator

Usage

```
bs_get_feed(  
  feed,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

feed	Character, length 1. Feed to get.
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of posts

Lexicon references

[feed/getFeed.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_feed('at://did:plc:z72i7hdynmk6r22z27h6tvur/app.bsky.feed.generator/bsky-team')
```

bs_get_feeds	<i>Retrieve a list of feeds created by a given actor</i>
--------------	--

Description

Retrieve a list of feeds created by a given actor

Usage

```
bs_get_feeds(  
  actor,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

actor	Character, length 1. name of 1 actor, such as 'chriskenny.bsky.social'
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of feeds

Lexicon references

[feed/getActorFeeds.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_feeds('chriskenny.bsky.social')
```

bs_get_feed_generator *Get specific information about one feed generator*

Description

Get specific information about one feed generator

Usage

```
bs_get_feed_generator(  
  feed,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

feed	Character, length 1. Feed to get.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of feeds

Lexicon references

[feed/getFeedGenerator.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

See Also

[bs_get_feed_generators\(\)](#) for less detailed information about multiple feed generators.

Examples

```
bs_get_feed_generator('at://did:plc:z72i7hdynmk6r22z27h6tvur/app.bsky.feed.generator/bsky-team')
```

bs_get_feed_generators

Get information about a list of feed generators

Description

Get information about a list of feed generators

Usage

```
bs_get_feed_generators(  
  feeds,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

feeds	Character. Vector of feeds to get.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of feeds

Lexicon references

[feed/getFeedGenerators.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

See Also

[bs_get_feed_generators\(\)](#) for more detailed information about one feed generator.

Examples

```
bs_get_feed_generators('at://did:plc:z72i7hdynmk6r22z27h6tvur/app.bsky.feed.generator/bsky-team')
bs_get_feed_generators(c(
  'at://did:plc:z72i7hdynmk6r22z27h6tvur/app.bsky.feed.generator/bsky-team',
  'at://did:plc:z72i7hdynmk6r22z27h6tvur/app.bsky.feed.generator/whats-hot'
))
```

bs_get_feed_suggestions

Get a list of feed suggestions

Description

Get a list of feed suggestions

Usage

```
bs_get_feed_suggestions(  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of suggested feeds

Lexicon references

[feed/getSuggestedFeeds.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_feed_suggestions()
```

bs_get_followers	<i>Retrieve an actor's followers</i>
------------------	--------------------------------------

Description

Retrieve an actor's followers

Usage

```
bs_get_followers(  
  actor,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

actor	Character, length 1. name of 1 actor, such as 'chriskenny.bsky.social'
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of actors

Lexicon references

[graph/getFollowers.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_followers('chriskenny.bsky.social')
```

bs_get_follows	<i>Retrieve an actor's follows</i>
----------------	------------------------------------

Description

Retrieve an actor's follows

Usage

```
bs_get_follows(  
  actor,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

actor	Character, length 1. name of 1 actor, such as 'chriskenny.bsky.social'
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of actors

Lexicon references

[graph/getFollows.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_follows('chriskenny.bsky.social')
```

`bs_get_follow_suggestions`*Get suggested follows related to a given actor*

Description

Get suggested follows related to a given actor

Usage

```
bs_get_follow_suggestions(  
  actor,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

<code>actor</code>	Character, length 1. name of 1 actor, such as 'chriskenny.bsky.social'
<code>user</code>	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
<code>pass</code>	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
<code>auth</code>	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
<code>clean</code>	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of actors

Lexicon references

[graph/getSuggestedFollowsByActor.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_follow_suggestions('chriskenny.bsky.social')
```

bs_get_likes	<i>Retrieve posts liked by an actor</i>
--------------	---

Description

Retrieve posts liked by an actor

Usage

```
bs_get_likes(  
  actor,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

actor	Character, length 1. name of 1 actor, such as 'chriskenny.bsky.social'
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of likes

Lexicon references

[feed/getActorLikes.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_likes('chriskenny.bsky.social')
```

bs_get_muted_lists *Retrieve a user's (self) muted lists*

Description

Retrieve a user's (self) muted lists

Usage

```
bs_get_muted_lists(  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of actors

Lexicon references

[graph/getListMutes.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_muted_lists()
```

bs_get_mutes	<i>Retrieve a user's (self) muted accounts</i>
--------------	--

Description

Retrieve a user's (self) muted accounts

Usage

```
bs_get_mutes(  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of actors

Lexicon references

[graph/getMutes.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_mutes()
```

bs_get_notifications *Get the user's (self) notifications*

Description

Get the user's (self) notifications

Usage

```
bs_get_notifications(  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a tibble with notifications

Lexicon references

[notification/listNotifications.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_notifications()
```

`bs_get_notification_count`*Get the user's (self) number of unread notifications*

Description

Get the user's (self) number of unread notifications

Usage

```
bs_get_notification_count(  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

<code>user</code>	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
<code>pass</code>	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
<code>auth</code>	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
<code>clean</code>	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a tibble with a single column and row for the count

Lexicon references

[notification/getUnreadCount.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_notification_count()
```

`bs_get_posts`*Retrieve thread of posts*

Description

Retrieve thread of posts

Usage

```
bs_get_posts(  
  uris,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

<code>uris</code>	Character. Vector of URIs for posts to get.
<code>user</code>	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
<code>pass</code>	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
<code>auth</code>	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
<code>clean</code>	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of posts

Lexicon references

[feed/getPostThread.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_posts('at://did:plc:ic6zqvuw5ulmfpjiwnhsr2ns/app.bsky.feed.post/3k7qmjev51r2s')
```

bs_get_post_likes	<i>Retrieve likes on a post</i>
-------------------	---------------------------------

Description

Retrieve likes on a post

Usage

```
bs_get_post_likes(  
  uri,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

uri	Character, length 1. URI for post to get.
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of likes

Lexicon references

[feed/getLikes.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_post_likes('at://did:plc:ic6zqvuw5ulmfpjiwnhsr2ns/app.bsky.feed.post/3k7qmjev51r2s')
```

bs_get_post_thread *Retrieve thread of posts*

Description

Retrieve thread of posts

Usage

```
bs_get_post_thread(  
  uri,  
  depth = NULL,  
  parent_height = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

uri	Character, length 1. URI for post to get.
depth	Integer. Maximum depth to request. Maximum: 1000
parent_height	Integer. Maximum parent height to request.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of posts

Lexicon references

[feed/getPostThread.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_post_thread('at://did:plc:ic6zqvuw5ulmfpjiwnhsr2ns/app.bsky.feed.post/3k7qmjev51r2s')
```

bs_get_preferences *Get (Self) Preferences*

Description

Get (Self) Preferences

Usage

```
bs_get_preferences(  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of preferences

Lexicon references

[actor/getPreferences.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_preferences()
```

`bs_get_profile`*Get Profile for a Bluesky Social User*

Description

Get Profile for a Bluesky Social User

Usage

```
bs_get_profile(  
  actors,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

<code>actors</code>	character vector of actor(s), such as 'chriskenny.bsky.social'
<code>user</code>	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
<code>pass</code>	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
<code>auth</code>	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
<code>clean</code>	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a tibble with a row for each actor

Lexicon references

[actor/getProfiles.json \(2023-10-01\)](#) [actor/getProfile.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_get_profile('chriskenny.bsky.social')  
bs_get_profile(actors = c('chriskenny.bsky.social', 'simko.bsky.social'))
```

bs_get_record	<i>Get an arbitrary record from a repo</i>
---------------	--

Description

Get an arbitrary record from a repo

Usage

```
bs_get_record(  
  repo = NULL,  
  collection = NULL,  
  rkey = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

repo	Character, length 1. The handle or DID of the repo.
collection	Character, length 1. The NSID of the record collection.
rkey	Character, length 1. The CID of the version of the record. If not specified, then return the most recent version.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of upload blob information

Lexicon references

[repo/getRecord.json \(2023-11-24\)](#)

Function introduced

v0.1.0 (2023-11-24)

Examples

```
bs_get_record('https://bsky.app/profile/bskyr.bsky.social/post/3kf2577exva2x')
```

bs_get_reposts	<i>Retrieve actors who reposted a post</i>
----------------	--

Description

Retrieve actors who reposted a post

Usage

```
bs_get_reposts(  
  uri,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

uri	Character, length 1. URI for post to get.
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of actors

Lexicon references

[feed/getRepostedBy.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_reposts('at://did:plc:ic6zqvuw5ulmfpjiwnhsr2ns/app.bsky.feed.post/3kaa2gxjh2a')
```

bs_get_timeline	<i>Retrieve the user's home timeline</i>
-----------------	--

Description

Retrieve the user's home timeline

Usage

```
bs_get_timeline(  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of posts

Lexicon references

[feed/getTimeline.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_get_timeline()
```

bs_like *Like an existing post*

Description

Like an existing post

Usage

```
bs_like(  
  post,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

post	Character vector, length 1. Link to a post.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of post information

Lexicon references

[feed/like.json \(2023-11-25\)](#) [repo/createRecord.json \(2023-11-25\)](#)

Function introduced

v0.1.0 (2023-11-25)

Examples

```
bs_like(post = 'https://bsky.app/profile/bskyr.bsky.social/post/3kf2577exva2x')
```

bs_list_records	<i>List records in a repo</i>
-----------------	-------------------------------

Description

List records in a repo

Usage

```
bs_list_records(  
  repo,  
  collection,  
  cursor = NULL,  
  limit = NULL,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

repo	Character, length 1. The handle or DID of the repo.
collection	Character, length 1. The NSID of the record collection.
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of record information

Lexicon references

[repo/createRecord.json \(2023-11-25\)](#)

Function introduced

v0.1.0 (2023-11-25)

Examples

```
bs_list_records(repo = 'chriskenny.bsky.social', collection = 'app.bsky.feed.post')
```

bs_post	<i>Make a post on Bluesky Social</i>
---------	--------------------------------------

Description

Make a post on Bluesky Social

Usage

```
bs_post(
  text,
  images,
  images_alt,
  langs,
  reply,
  quote,
  user = get_bluesky_user(),
  pass = get_bluesky_pass(),
  auth = bs_auth(user, pass),
  clean = TRUE
)
```

Arguments

text	text of post
images	character vector of paths to images to attach to post
images_alt	character vector of alt text for images. Must be same length as images if used.
langs	character vector of languages in BCP-47 format
reply	character vector with link to the parent post to reply to
quote	character vector with link to a post to quote
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of post information

Lexicon references

[feed/post.json \(2023-10-02\)](#) [repo/createRecord.json \(2023-10-02\)](#)

Function introduced

v0.0.1 (2023-10-02)

Examples

```
bs_post('Test post from R CMD Check for r package `bskyr`
via @bskyr.bsky.social (https://christophertkenny.com/bskyr/)')
bs_post('Test self-reply from r package `bskyr`
via @bskyr.bsky.social (https://christophertkenny.com/bskyr/)',
  reply = 'https://bsky.app/profile/bskyr.bsky.social/post/3kexwuoyqj32g')
bs_post('Test quoting from r package `bskyr`
via @bskyr.bsky.social (https://christophertkenny.com/bskyr/)',
  quote = 'https://bsky.app/profile/bskyr.bsky.social/post/3kf24wd6cmb2a')
bs_post('Test quote and reply from r package `bskyr`
via @bskyr.bsky.social (https://christophertkenny.com/bskyr/)',
  reply = 'https://bsky.app/profile/bskyr.bsky.social/post/3kexwuoyqj32g',
  quote = 'https://bsky.app/profile/bskyr.bsky.social/post/3kf24wd6cmb2a')
```

bs_repost

Repost an existing post

Description

Repost an existing post

Usage

```
bs_repost(
  post,
  user = get_bluesky_user(),
  pass = get_bluesky_pass(),
  auth = bs_auth(user, pass),
  clean = TRUE
)
```

Arguments

post	Character vector, length 1. Link to a post.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of post information

Lexicon references

[feed/post.json \(2023-11-25\)](#) [repo/createRecord.json \(2023-11-25\)](#)

Function introduced

v0.1.0 (2023-11-25)

Examples

```
bs_repost('https://bsky.app/profile/bskyr.bsky.social/post/3kf2577exva2x')
```

bs_resolve_handle	<i>Resolve a Handle to Decentralized Identifier (DID)</i>
-------------------	---

Description

Resolve a Handle to Decentralized Identifier (DID)

Usage

```
bs_resolve_handle(
  handle,
  user = get_bluesky_user(),
  pass = get_bluesky_pass(),
  auth = bs_auth(user, pass),
  clean = TRUE
)
```

Arguments

handle	Character, length 1. Handle, such as 'chriskenny.bsky.social'
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of decentralized identifier

Lexicon references

[identity/resolveHandle.json \(2023-11-24\)](#)

Function introduced

v0.1.0 (2023-11-24)

Examples

```
bs_resolve_handle('chriskenny.bsky.social')
```

bs_search_actors	<i>Find profiles matching search criteria</i>
------------------	---

Description

Find profiles matching search criteria

Usage

```
bs_search_actors(
  query,
  typeahead = FALSE,
  cursor = NULL,
  limit = NULL,
  user = get_bluesky_user(),
  pass = get_bluesky_pass(),
  auth = bs_auth(user, pass),
  clean = TRUE
)
```

Arguments

query	character. search query, Lucene query syntax is recommended when typeahead = FALSE.
typeahead	logical. Use typeahead for search? Default is FALSE.
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to get_bluesky_user().
pass	Character. App password to log in with. Defaults to get_bluesky_pass().
auth	Authentication information. Defaults to bs_auth(user, pass).
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of suggested accounts to follow

Lexicon references

[actor/searchActors.json \(2023-10-01\)](#) [actor/searchActorsTypeahead.json \(2023-10-01\)](#)

Function introduced

v0.0.1 (2023-10-01)

Examples

```
bs_search_actors('political science')
```

bs_search_posts

Find posts matching search criteria

Description

Find posts matching search criteria

Usage

```
bs_search_posts(
  query,
  cursor = NULL,
  limit = NULL,
  user = get_bluesky_user(),
  pass = get_bluesky_pass(),
  auth = bs_auth(user, pass),
  clean = TRUE
)
```

Arguments

query	character. search query, Lucene query syntax is recommended.
cursor	Character, length 1. A cursor property from a prior response. Default: NULL.
limit	Integer. Number of records to request. If over 100, multiple requests are made.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of suggested accounts to follow

Lexicon references

[feed/searchPosts.json \(2023-12-13\)](#)

Function introduced

v0.1.1 (2023-12-13)

Examples

```
bs_search_posts('redistricting')
```

bs_upload_blob	<i>Upload a blob to a repo</i>
----------------	--------------------------------

Description

Upload a blob to a repo

Usage

```
bs_upload_blob(  
  blob,  
  user = get_bluesky_user(),  
  pass = get_bluesky_pass(),  
  auth = bs_auth(user, pass),  
  clean = TRUE  
)
```

Arguments

blob	Character, files to upload to a repo.
user	Character. User name to log in with. Defaults to <code>get_bluesky_user()</code> .
pass	Character. App password to log in with. Defaults to <code>get_bluesky_pass()</code> .
auth	Authentication information. Defaults to <code>bs_auth(user, pass)</code> .
clean	Logical. Should output be cleaned into a tibble? Default: TRUE.

Value

a `tibble::tibble` of upload blob information

Lexicon references

[repo/uploadBlob.json \(2023-11-24\)](#)

Function introduced

v0.1.0 (2023-11-24)

Examples

```
fig <- fs::path_package('bskyr', 'man/figures/logo.png')
bs_upload_blob(fig)
```

bs_uri_to_url	<i>Convert Universal Resource Identifiers to Hypertext Transfer Protocol Secure URLs</i>
---------------	--

Description

Convert Universal Resource Identifiers to Hypertext Transfer Protocol Secure URLs

Usage

```
bs_uri_to_url(uri)
```

Arguments

uri Character, length 1. URI for post to get.

Value

character vector of HTTPS URLs

Examples

```
bs_uri_to_url('at://did:plc:ic6zqvuw5ulmfpjiwnhsr2ns/app.bsky.feed.post/3k7qmjev51r2s')
```

pass	<i>Check or Get Bluesky App Password</i>
------	--

Description

Check or Get Bluesky App Password

Usage

```
has_bluesky_pass()
```

```
get_bluesky_pass()
```

```
bs_get_pass()
```

```
bs_has_pass()
```

Value

logical if has, pass if get

Examples

```
has_bluesky_pass()
```

set_bluesky_pass	<i>Add Entry to Renviron</i>
------------------	------------------------------

Description

Adds Bluesky App Password to .Renviron.

Usage

```
set_bluesky_pass(pass, overwrite = FALSE, install = FALSE, r_env = NULL)
```

```
bs_set_pass(pass, overwrite = FALSE, install = FALSE, r_env = NULL)
```

Arguments

pass	Character. App Password to add to add.
overwrite	Defaults to FALSE. Boolean. Should existing BLUESKY_APP_PASS in Renviron be overwritten?
install	Defaults to FALSE. Boolean. Should this be added '~/.Renviron' file?
r_env	Path to install to if install is TRUE.

Value

pass, invisibly

Examples

```
example_env <- tempfile(fileext = '.Renviron')
set_bluesky_pass('1234-1234-1234-1234', r_env = example_env)
# r_env should likely be: file.path(Sys.getenv('HOME'), '.Renviron')
```

set_bluesky_user	<i>Adds Bluesky User to .Renviron.</i>
------------------	--

Description

Adds Bluesky User to .Renviron.

Usage

```
set_bluesky_user(user, overwrite = FALSE, install = FALSE, r_env = NULL)
```

```
bs_set_user(user, overwrite = FALSE, install = FALSE, r_env = NULL)
```

Arguments

user	Character. User to add to add.
overwrite	Defaults to FALSE. Boolean. Should existing BLUESKY_APP_USER in Renviron be overwritten?
install	Defaults to FALSE. Boolean. Should this be added '~/.Renviron' file?
r_env	Path to install to if install is TRUE.

Value

user, invisibly

Examples

```
example_env <- tempfile(fileext = '.Renviron')
set_bluesky_user('CRAN_EXAMPLE.bsky.social', r_env = example_env)
# r_env should likely be: file.path(Sys.getenv('HOME'), '.Renviron')
```

user	<i>Check or Get Bluesky User</i>
------	----------------------------------

Description

Check or Get Bluesky User

Usage

has_bluesky_user()

get_bluesky_user()

bs_get_user()

bs_has_user()

Value

logical if has, user if get

Examples

has_bluesky_user()

Index

- * **actor**
 - bs_get_actor_suggestions, 8
 - bs_get_preferences, 29
 - bs_get_profile, 30
 - bs_search_actors, 39
- * **auth**
 - bs_auth, 3
 - pass, 43
 - set_bluesky_pass, 43
 - set_bluesky_user, 44
 - user, 45
- * **feed**
 - bs_get_author_feed, 9
 - bs_get_feed, 12
 - bs_get_feed_generator, 14
 - bs_get_feed_generators, 15
 - bs_get_feed_suggestions, 16
 - bs_get_feeds, 13
 - bs_get_likes, 21
 - bs_get_post_likes, 27
 - bs_get_post_thread, 28
 - bs_get_posts, 26
 - bs_get_reposts, 32
 - bs_get_timeline, 33
 - bs_search_posts, 40
- * **graph**
 - bs_get_actor_lists, 7
 - bs_get_blocked_lists, 10
 - bs_get_blocks, 11
 - bs_get_follow_suggestions, 20
 - bs_get_followers, 17
 - bs_get_follows, 19
 - bs_get_muted_lists, 22
 - bs_get_mutes, 23
- * **helper**
 - bs_created_at, 3
- * **identity**
 - bs_resolve_handle, 38
- * **notification**
 - bs_get_notification_count, 25
 - bs_get_notifications, 24
- * **record**
 - bs_like, 34
 - bs_post, 36
 - bs_repost, 37
- * **repo**
 - bs_create_record, 4
 - bs_delete_record, 5
 - bs_describe_repo, 6
 - bs_get_record, 31
 - bs_list_records, 35
 - bs_upload_blob, 41
- bs_auth, 3
- bs_create_record, 4
- bs_created_at, 3
- bs_delete_record, 5
- bs_describe_repo, 6
- bs_get_actor_lists, 7
- bs_get_actor_suggestions, 8
- bs_get_author_feed, 9
- bs_get_blocked_lists, 10
- bs_get_blocks, 11
- bs_get_feed, 12
- bs_get_feed_generator, 14
- bs_get_feed_generators, 15
- bs_get_feed_generators(), 15, 16
- bs_get_feed_suggestions, 16
- bs_get_feeds, 13
- bs_get_follow_suggestions, 20
- bs_get_followers, 17
- bs_get_follows, 19
- bs_get_likes, 21
- bs_get_muted_lists, 22
- bs_get_mutes, 23
- bs_get_notification_count, 25
- bs_get_notifications, 24
- bs_get_pass (pass), 43
- bs_get_post_likes, 27

bs_get_post_thread, 28
bs_get_posts, 26
bs_get_preferences, 29
bs_get_profile, 30
bs_get_record, 31
bs_get_reposts, 32
bs_get_timeline, 33
bs_get_user (user), 45
bs_has_pass (pass), 43
bs_has_user (user), 45
bs_like, 34
bs_list_records, 35
bs_post, 36
bs_repost, 37
bs_resolve_handle, 38
bs_search_actors, 39
bs_search_posts, 40
bs_set_pass (set_bluesky_pass), 43
bs_set_user (set_bluesky_user), 44
bs_upload_blob, 41
bs_uri_to_url, 42

get_bluesky_pass (pass), 43
get_bluesky_user (user), 45

has_bluesky_pass (pass), 43
has_bluesky_user (user), 45

pass, 43

set_bluesky_pass, 43
set_bluesky_user, 44

tibble::tibble, 4, 7–23, 26–29, 31–36, 38,
40, 41

user, 45