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)


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]

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

**Index** 46
bs_auth

Authenticate a user

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

Get current time in Bluesky format

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()

hs_create_record  Create a record in a repo

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 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 record information

Lexicon references

repo/createRecord.json (2023-11-25)
bs_delete_record

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

Delete a record in a repo

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 http status code

Lexicon references

repo/deleteRecord.json (2023-11-25)

Function introduced

v0.1.0 (2023-11-25)

Examples

```r
# 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**

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

Arguments

- **repo**: Character, length 1. The handle or DID of the repo.
- **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 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')
```

Description

Get a list of lists that belong to an actor.

Usage

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

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')

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
)
bs_get_author_feed

Arguments

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

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
)
bs_get_blocked_lists

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
)
bs_get_blocks

Arguments

- cursor: Character, length 1. A cursor property from a prior response. Default: NULL.
- 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

Retrieve user (self) blocks

Description

Retrieve user (self) blocks

Usage

```r
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**

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

**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**

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

---

**bs_get_feeds**  
*Retrieve a list of feeds created by a given actor*

**Description**

Retrieve a list of feeds created by a given actor

**Usage**

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

**Arguments**

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

**Value**

A `tibble::tibble` of feeds

**Lexicon references**

`feed/getActorFeeds.json (2023-10-01)`

**Function introduced**

`v0.0.1 (2023-10-01)`

**Examples**

```r
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**

```r
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**

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

**Description**

Get information about a list of feed generators

**Usage**

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

**Get a list of feed suggestions**

```r
bs_get_feed_suggestions()
```

**Description**

Get a list of feed suggestions

**Arguments**

- **feeds** Character. Vector of feeds 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/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**

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

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

Retrieve an actor’s follows

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

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

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

```r
bs_get_follow_suggestions(
  actor,
  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`
- **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/getSuggestedFollowsByActor.json (2023-10-02)`

### Function introduced

- v0.0.1 (2023-10-02)

### Examples

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

Retrieve posts liked by an actor

Usage

```r
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

```r
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 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_muted_lists()
bs_get_mutes

Retrieve a user's (self) muted accounts

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

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

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**

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

**Arguments**

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

**Value**

A tibble with notifications

**Lexicon references**

`notification/listNotifications.json` (2023-10-02)

**Function introduced**

`v0.0.1` (2023-10-02)

**Examples**

```r
bs_get_notifications()
```
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

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 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

uris Character. Vector of URIs for posts 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 posts

Lexicon references

feed/getPostThread.json (2023-10-01)

Function introduced

v0.0.1 (2023-10-01)

Examples

bs_get_posts('at://did:plc:ic6zqvuw5ulmfpjiwhtsr2ns/app.bsky.feed.post/3k7qmjev5lr2s')
bs_get_post_likes

Retrieve likes on a post

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 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 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:ic6zqvuw5ulmfpiwnhsr2ns/app.bsky.feed.post/3k7qmjev5lr2s')
**bs_get_post_thread**  
*Retrieve thread of posts*

**Description**
Retrieve thread of posts

**Usage**

```r
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/3k7qjev5lr2s')`
Get (Self) Preferences

Usage

```r
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 `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 preferences

Lexicon references

- `actor/getPreferences.json (2023-10-01)`

Function introduced

- v0.0.1 (2023-10-01)

Examples

```r
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

actors  character vector of actor(s), such as 'chriskenny.bsky.social'
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 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

Get an arbitrary record from a repo

Description

Get an arbitrary record from a repo

Usage

```r
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 `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 upload blob information

Lexicon references

- `repo/getRecord.json (2023-11-24)`

Function introduced

- `v0.1.0 (2023-11-24)`

Examples

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

Retrieve actors who reposted a post

Usage

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

Arguments

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

Value

A `tibble::tibble` of actors

Lexicon references

`feed/getRepostedBy.json (2023-10-01)`

Function introduced

`v0.0.1 (2023-10-02)`

Examples

```r
bs_get_reposts('at://did:plc:ic6zqvuw5ulmfpjiwnhsr2ns/app.bsky.feed.post/3kaa2gxjhzr2a')
```
bs_get_timeline

Retrieve the user’s home timeline

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**

```r
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 `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 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**

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

*List records in a repo*

**Description**

List records in a repo

**Usage**

```r
def 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 `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 record information

**Lexicon references**

- `repo/createRecord.json (2023-11-25)`

**Function introduced**

- v0.1.0 (2023-11-25)
Examples

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

---

```r
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 `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 post information
bs_repost

Lexicon references

feed/post.json (2023-10-02) repo/createRecord.json (2023-10-02)

Function introduced

v0.0.1 (2023-10-02)

Examples

```r
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/3exwuoyqj32g')
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/3exwuoyqj32g', quote = 'https://bsky.app/profile/bskyr.bsky.social/post/3kf24wd6cmb2a')
```

bs_repost

Repost an existing post

Description

Repost an existing post

Usage

```r
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 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.
**bs_resolve_handle**

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

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

---

**bs_resolve_handle**  
*Resolve a Handle to Decentralized Identifier (DID)*

Description

Resolve a Handle to Decentralized Identifier (DID)

Usage

```r
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 `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 decentralized identifier
**bs_search_actors**

Find profiles matching search criteria

**Description**

Find profiles matching search criteria

**Usage**

```r
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

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

---

bs_search_posts

Find posts matching search criteria

Usage

```r
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 `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

feed/searchPosts.json (2023-12-13)

Function introduced

v0.1.1 (2023-12-13)

Examples

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

### bs_upload_blob

**Upload a blob to a repo**

**Description**

Upload a blob to a repo

**Usage**

```r
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 `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 upload blob information
bs_uri_to_url

Convert Universal Resource Identifiers to Hypertext Transfer Protocol Secure URLs

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:ic6zqvw5ulmpjwhr2ns/app.bsky.feed.post/3k7qmev5lr2s')
Check or Get Bluesky App Password

**Description**

Check or Get Bluesky App Password

**Usage**

```r
has_bluesky_pass()
get_bluesky_pass()
bs_get_pass()
bs_has_pass()
```

**Value**

logical if has, pass if get

**Examples**

```r
has_bluesky_pass()
```

---

Add Entry to Renviron

**Description**

Adds Bluesky App Password to .Renviron.

**Usage**

```r
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.
set_bluesky_user

Value

pass, invisibly

Examples

```r
e.example_env <- tempfile(fileext = '
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 Adds Bluesky User to .Renviron.

Description

Adds Bluesky User to .Renviron.

Usage

```r
set_bluesky_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 Renvi
- `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

```r
e.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

Check or Get Bluesky User

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_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
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
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

give_bluesky_pass (pass), 43
give_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