Package ‘qbr’

October 13, 2022

Title Access the ‘Quickbase’ JSON API
Version 1.1.0
Description

URL https://github.com/BHII-KSC/qbr
BugReports https://github.com/BHII-KSC/qbr/issues
License GPL (>= 3)
Encoding UTF-8
RoxygenNote 7.1.2
Imports dplyr (>= 1.0.9), httr (>= 1.4.3), jsonlite (>= 1.8.0),
  magrittr (>= 2.0.3), purrr (>= 0.3.4), stringr (>= 1.4.0),
  tibble (>= 3.1.7)
Suggests testthat (>= 3.0.0), keyring
Config/testthat/edition 3
NeedsCompilation no
Author John Erdmann [aut, cre],
  Keene State College [cph, fnd]
Maintainer John Erdmann <john.erdmann@keene.edu>
Repository CRAN
Date/Publication 2022-06-30 18:10:02 UTC

R topics documented:

  clone_token ................................................................. 2
  copy_app ................................................................. 3
  deactivate_token ....................................................... 4
  delete_app ............................................................... 5
  delete_token ............................................................ 6
**clone_token**

Clone a user token

**Description**

`clone_token` makes a copy of the supplied token and returns its value.

**Usage**

```r
close_token(
    subdomain,
    auth,
    agent = NULL,
    clone_name = NULL,
    clone_desc = NULL
)
```

**Arguments**

- `subdomain`: Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
- `auth`: Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
- `clone_name`: Optional. Character vector with one element. Name the token clone.
- `clone_desc`: Optional. Character vector with one element. Provide a description for the token clone.

**Value**

A character vector with one element containing the token clone.

**References**

Quickbase API documentation
Examples

```r
## Not run:
x <- clone_token(subdomain = "abc",
                 auth = keyring::key_get("qb_example"),
                 clone_name = "My new token",
                 clone_desc = "This clone was created using R")

## End(Not run)
```

---

**copy_app**

**Copy an app**

**Description**

copy_app Copy an app. Provides options to copy data and users.

**Usage**

copy_app(
  subdomain,
  auth,
  app_id,
  app_name,
  app_desc = NULL,
  agent = NULL,
  users_and_roles = FALSE,
  keep_data = FALSE,
  exclude_files = TRUE,
  assign_user_token = TRUE
)

**Arguments**

- **subdomain**: Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
- **auth**: Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
- **app_id**: Character vector with one element. Unique identifier of an app. Found in the URL of the app’s homepage after /db/
- **app_name**: Character vector with one element. Name the app’s copy.
- **app_desc**: Optional. Character vector with one element. Describe the app’s copy.
- **agent**: Optional. Character vector with one element. Describes user/agent making API call.
- **users_and_roles**: Logical. If true, users will be copied along with their assigned roles. If false, users and roles will be copied but roles will not be assigned.
**deactivate_token**

Deactivate a user token

**Description**
deactivate_token makes an active user token inactive.

**Usage**
deactivate_token(subdomain, auth, agent = NULL)

**Arguments**
- **subdomain**: Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
- **auth**: Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
- **agent**: Optional. Character vector with one element. Describes user/agent making API call.

**Value**
A message confirming deactivation was successful.
delete_app

Examples

```r
## Not run:
  x <- deactivate_token(subdomain = "abc",
                        auth = keyring::key_get("qb_example"))

## End(Not run)
```

delete_app  Delete an app

Description

delete_app Delete an entire app, including all of the tables and data.

Usage

```r
delete_app(subdomain, auth, app_id, app_name, agent = NULL)
```

Arguments

- `subdomain` Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
- `auth` Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
- `app_id` Character vector with one element. Unique identifier of an app. Found in the URL of the app’s homepage after /db/
- `app_name` Character vector with one element. The name of the app to be delete. Confirms you want to delete the app.
- `agent` Optional. Character vector with one element. Describes user/agent making API call.

Value

A list.

Examples

```r
## Not run:
delete_app(subdomain = "abc",
            auth = keyring::key_get("qb_example"),
            app_id = "bsf5hphe5",
            app_name = "R Testing copy")

## End(Not run)
```
delete_token  
\textit{Delete a user token}

\textbf{Description}  
delete_token permanently deletes an active user token.

\textbf{Usage}  
delete_token(subdomain, auth, agent = NULL)

\textbf{Arguments}  
\begin{itemize}
  \item \texttt{subdomain}  
  Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
  \item \texttt{auth}  
  Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
  \item \texttt{agent}  
  Optional. Character vector with one element. Describes user/agent making API call.
\end{itemize}

\textbf{Value}  
A message confirming deactivation was successful.

\textbf{Examples}  

\begin{verbatim}
## Not run:
x <- delete_token(subdomain = "abc",
    auth = keyring::key_get("qb_example"))

## End(Not run)
\end{verbatim}

get_app_events  
\textit{Get app events}

\textbf{Description}  
get_app_events Get a tibble of events that can be triggered based on data or user actions in this application, includes: Email notification, Reminders, Subscriptions, QB Actions, Webhooks, record change triggered Automations (does not include scheduled).

\textbf{Usage}  
get_app_events(subdomain, auth, app_id, agent = NULL)
get_report

Arguments

subdomain Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
auth Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
app_id Character vector with one element. Unique identifier of an app. Found in the URL of the app’s homepage after /db/
agent Optional. Character vector with one element. Describes user/agent making API call.

Value

A tibble.

Examples

## Not run:
get_app_events(subdomain = "abc",
               auth = keyring::key_get("qb_example"),
               app_id = "bn9d8f78g")
## End(Not run)

get_report

Get a report

Description

get_report retrieves metadata about the specified Quickbase report.

Usage

get_report(subdomain, auth, table_id, report_id, agent = NULL)

Arguments

subdomain Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
auth Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
table_id Character vector with one element. Found in the URL of a Quickbase table between /db/ and ?
report_id Character vector with one element. Found in the 'Reports & Charts' page in Quickbase and in the report URL.
agent Optional. Character vector with one element. Describes user/agent making API call.
get_reports

Description

get_reports retrieves metadata for each report in a table.

Usage

get_reports(subdomain, auth, table_id, agent = NULL)

Arguments

subdomain Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
auth Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
table_id Character vector with one element. Found in the URL of a Quickbase table between /db/ and ?
agent Optional. Character vector with one element. Describes user/agent making API call.

Value

A tibble.

Examples

## Not run:
get_reports(subdomain = "abc",
            auth = keyring::key_get("qb_example"),
            table_id = "bn9d8iesz",
            report_id = "7")

## End(Not run)
**get_users**

_Get users_

**Description**

get_users Get all users in an account. Provides options to limit values to a set of users and/or apps.

**Usage**

```r
get_users(
  subdomain,
  auth,
  agent = NULL,
  account_id = NULL,
  user_emails = NULL,
  app_ids = NULL
)
```

**Arguments**

- `subdomain`: Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
- `auth`: Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
- `account_id`: Optional. Positive integer. The account ID being used to get users. If no value is specified, the first account associated with the requesting user token is chosen.
- `user_emails`: Optional. Character vector. Limit returned users to those specified in this list.
- `app_ids`: Optional. Character vector. Limit returned users to those assigned to these app ID’s. The provided app ID’s should belong to the same account.

**Value**

A tibble.

**Examples**

```r
## Not run:
get_users(subdomain = "abc",
           auth = keyring::key_get("qb_example"))

## End(Not run)
```
qbr

qbr: A package for interacting with the 'Quickbase' JSON API

Description

The qbr package is currently focused on extracting data from 'Quickbase' via its JSON API.

qbr functions

* qb_run Runs a 'Quickbase' report and returns its data as a tibble.

qb_run

Run a Quickbase report

Description

qb_run asks the Quickbase API to run a report and returns its data.

Usage

qb_run(
  subdomain,
  token,
  table_id,
  report_id,
  agent = NULL,
  skip = 0,
  top = 0,
  type_suffix = FALSE,
  paginate = TRUE
)

Arguments

subdomain Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
token Character vector with one element. Created in 'My Preferences' under 'Manage user tokens' link.
table_id Character vector with one element. Found in the URL of a Quickbase table between /db/ and ?
report_id Character vector with one element. Found in the 'Reports & Charts' page in Quickbase and in the report URL.
agent Optional. Character vector with one element. Describes user/agent making API call.
run_report

### skip
Optional. Integer. The number of rows to skip from the top of a record set.

### top
Optional. Integer. The limit on the number of records to pull starting at the top of a record set.

### type_suffix
Optional. Logical. Set TRUE to append each field label with its Quickbase data type.

### paginate
Optional. Logical. Set TRUE to recursively call the API until all report pages are collected.

**Value**

A tibble.

**References**

Quickbase API documentation

**Examples**

```r
## Not run:

# Get all data in a report
my_tibble <- qb_run(subdomain = "abc",
                    token = keyring::key_get("qb_example"),
                    table_id = "bn9d8iesz",
                    report_id = "1")

# Get rows 3 to 6 from a report
my_tibble <- qb_run(subdomain = "abc.quickbase.com",
                    token = keyring::key_get("qb_example"),
                    table_id = "bn9d8iesz",
                    report_id = "1",
                    skip = 2,
                    top = 3)

## End(Not run)
```

---

**run_report**

*Run a report*

**Description**

`run_report` asks the Quickbase API to run a report and returns its data.
Usage

```
run_report(
  subdomain,  # Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
  auth,       # Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
  table_id,   # Character vector with one element. Found in the URL of a Quickbase table between /db/ and ?
  report_id,  # Character vector with one element. Found in the 'Reports & Charts' page in Quickbase and in the report URL.
  agent = NULL,  # Optional. Character vector with one element. Describes user/agent making API call.
  skip = 0,     # Optional. Integer. The number of rows to skip from the top of a record set.
  top = 0,      # Optional. Integer. The limit on the number of records to pull starting at the top of a record set.
  type_suffix = FALSE,  # Optional. Logical. Set TRUE to append each field label with its Quickbase data type.
  paginate = TRUE)  # Optional. Logical. Set TRUE to recursively call the API until all report pages are collected.
```

Arguments

- **subdomain**: Character vector with one element. Found at the beginning of the Quickbase URL. Realm specific.
- **auth**: Character vector with one element. The Quickbase authentication scheme you are using to authenticate the request (e.g., user token).
- **table_id**: Character vector with one element. Found in the URL of a Quickbase table between /db/ and ?
- **report_id**: Character vector with one element. Found in the 'Reports & Charts' page in Quickbase and in the report URL.
- **agent**: Optional. Character vector with one element. Describes user/agent making API call.
- **skip**: Optional. Integer. The number of rows to skip from the top of a record set.
- **top**: Optional. Integer. The limit on the number of records to pull starting at the top of a record set.
- **type_suffix**: Optional. Logical. Set TRUE to append each field label with its Quickbase data type.
- **paginate**: Optional. Logical. Set TRUE to recursively call the API until all report pages are collected.

Value

A tibble.

References

- Quickbase API documentation

Examples

```r
## Not run:

# Get all data in a report
my_tibble <- run_report(subdomain = "abc",
```
run_report

auth = keyring::key_get("qb_example"),
table_id = "bn9d8iesz",
report_id = "1")

# Get rows 3 to 6 from a report
my_tibble <- run_report(subdomain = "abc.quickbase.com",
auth = keyring::key_get("qb_example"),
table_id = "bn9d8iesz",
report_id = "1",
skip = 2,
top = 3)

## End(Not run)
Index

clone_token, 2
copy_app, 3

deactivate_token, 4
delete_app, 5
delete_token, 6

get_app_events, 6
get_report, 7
get_reports, 8
get_users, 9

gb_run, 10
gbr, 10

run_report, 11