Package ‘qbr’

November 11, 2022

Title  Access the 'Quickbase' JSON API
Version 1.2.0
Description Programmatically access the 'Quickbase' JSON API <https://developer.quickbase.com>. You supply parameters for an API call, 'qbr' delivers an http request to the API endpoint and returns its response. Outputs follow 'tidyverse' philosophy.

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), tidyr (>= 1.2.0), 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, fnl]
Maintainer John Erdmann <john.erdmann@keene.edu>
Repository CRAN
Date/Publication 2022-11-11 16:50:02 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone_token</td>
<td>2</td>
</tr>
<tr>
<td>copy_app</td>
<td>3</td>
</tr>
<tr>
<td>deactivate_token</td>
<td>4</td>
</tr>
<tr>
<td>delete_app</td>
<td>5</td>
</tr>
<tr>
<td>delete_token</td>
<td>6</td>
</tr>
</tbody>
</table>
**Description**

Make a copy of the supplied token and returns its value.

**Usage**

```r
clonetoken(
  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).
- `agent` Optional. Character vector with one element. Describes user/agent making API call.
- `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 an app. Provides options to copy data and users.

**Usage**

```r
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.
### deactivate_token

**Deactivate a user token**

**Description**

Make an active user token inactive.

**Usage**

```r
disable_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.

#### Examples

```r
## Not run:
copy_app(subdomain = "abc",
          auth = keyring::key_get("qb_example"),
          app_id = "bn9d8f78g",
          app_name = "Copy of my app",
          keep_data = TRUE)
## End(Not run)
```
delete_app

Value

A message confirming deactivation was successful.

Examples

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

## End(Not run)
```

---

Delete an app

Description

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  Delete a user token

Description
Permanently delete an active user token.

Usage
delete_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.

Examples
## Not run:
x <- delete_token(subdomain = "abc",
auth = keyring::key_get("qb_example"))
## End(Not run)

get_app  Get an app

Description
Get metadata for an app.
get_app_events

Usage

get_app(
    subdomain,  
    auth,  
    app_id,  
    agent = NULL,  
    include_sec = T,  
    include_vars = T
)

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.
include_sec Logical. Includes security properties if true.
include_vars Logical. Includes app variables if true.

Value

A tibble.

Examples

## Not run:
get_app(subdomain = "abc",  
    auth = keyring::key_get("qb_example"),  
    app_id = "bsf5hphe5")

## End(Not run)

desc

get_app_events  Get app events

Description

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).
Usage

get_app_events(subdomain, auth, app_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).
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_fields

Get all fields in a table

Description

Get metadata for all fields in a table.

Usage

get_fields(
    subdomain,
    auth,
    table_id,
    agent = NULL,
    include_props = T,
    include_perms = F
)
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).
- **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.
- **include_props**: Logical. Includes field properties if true.
- **include_perms**: Logical. Includes field permissions if true.

Value

A tibble.

Examples

```r
## Not run:
get_fields(subdomain = "abc",
auth = keyring::key_get("qb_example"),
table_id = "bsf5hphe5")
## End(Not run)
```

get_report

Get a report

Description

Get metadata about a report.

Usage

```r
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.
Value
A named list.

Examples

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

## End(Not run)
```

---

get_reports  
Get all reports for a table

Description
get_reports retrieves metadata for each report in a table.

Usage
```r
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

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

## End(Not run)
```
get_tables

Get all tables

Description

Get metadata for all tables in an app.

Usage

get_tables(subdomain, auth, app_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).
- app_id: Character vector with one element. Unique identifier of an app. Found in the URL of the app’s homepage after /db/

Value

A tibble.

Examples

```r
## Not run:
get_tables(subdomain = "abc", 
    auth = keyring::key_get("qb_example"), 
    app_id = "bsf5hphe5")
## End(Not run)
```

get_users

Get users

Description

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

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).
agent      Optional. Character vector with one element. Describes user/agent making API call.
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.
Run a Quickbase report

Description
Run a report and get 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.
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.
run_report

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 a report and get its data.

Usage

```r
run_report(
  subdomain, 
  auth, 
  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.
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

```
## Not run:

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

Summarize an app

Description
Get metadata for an app, its tables, and its fields.

Usage
summarize_app(subdomain, auth, app_id, agent = NULL)

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

Value
A list of tibbles.

Examples
```r
## Not run:
summarize_app(subdomain = "abc",
               auth = keyring::key_get("qb_example"),
               app_id = "bsf5hphe5")
## End(Not run)
```
Index

clone_token, 2
copy_app, 3
deactivate_token, 4
delete_app, 5
delete_token, 6
get_app, 6
get_app_events, 7
get_fields, 8
get_report, 9
get_reports, 10
get_tables, 11
get_users, 11

qb_run, 13
qbr, 12
run_report, 14
summarize_app, 16