Package ‘openappr’

December 3, 2024

Title Retrieve App Data from 'OpenAppBuilder'

Version 0.2.0

Description Provides an interface to connect R with the <https://github.com/IDEMSInternational/open-app-builder> 'OpenAppBuilder' platform, enabling users to retrieve and work with user and notification data for analysis and processing. It is designed for developers and analysts to seamlessly integrate data from 'OpenAppBuilder' into R workflows via a 'Postgres' database connection, allowing direct querying and import of app data into R.

License LGPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests knitr,
testthat,
rmarkdown

Imports DBI,
dplyr,
jsonlite,
magrittr,
purrr,
RPostgres,
stringr

VignetteBuilder knitr

Contents

get_app_connection — *Get the app connection from the environment*

Description

Call the app connection. The connection is set in the function `set_app_connection`.

Usage

```
get_app_connection()
```

Value

returns a the app connection to the app data.

Examples

```
# Establish a connection to the PostgreSQL database
set_app_connection(
  dbname = "vmc",
  host = "apps-server.idems.international",
  port = 5432,
  user = "vmc",
  password = "LSQkyYg5KzL747"
)

get_app_connection()
```

get_nf_data — *Get notification data from OpenAppBuilder*

Description

This function retrieves data from the `app_notification_interaction` table in OpenAppBuilder and efficiently parses the `notification_meta` column from JSON format.

Usage

```
get_nf_data(
  site = get_app_connection(),
  filter = FALSE,
  filter_variable = NULL,
  filter_variable_value = NULL
)
```

Arguments

`site`	The name of the PostgreSQL database connection (using `DBI::dbConnect` or `set_app_connection()`).
`filter`	A logical value indicating whether to filter the data (defaults to `FALSE`).
`filter_variable`	A character string representing the name of the column to filter if `filter` == `TRUE` and `filter_variable_value` is provided.
`filter_variable_value`	A character string representing the value of the `filter_variable` column to filter if `filter` == `TRUE`.

Value

A data frame containing notification interaction data from OpenAppBuilder, with the `notification_meta` column parsed into separate columns.

Examples

```
# First we need to set an app connection
set_app_connection(
  dbname = "vmc",
  host = "apps-server.idems.international",
  port = 5432,
  user = "vmc",
  password = "LSQkyYg5KzL747"
)

# Retrieve all notification data
data_all_nf <- get_nf_data()

# Retrieve data where 'app_user_id' is '3e68fcda-d4cd-400e-8b12-6ddfabced348'
# or '223925c7-443a-411c-aa2a-a394f991dd52'
valid_ids <- c("3e68fcda-d4cd-400e-8b12-6ddfabced348",
               "223925c7-443a-411c-aa2a-a394f991dd52")
data_filtered_users <- get_nf_data(
  filter = TRUE,
  filter_variable = "app_user_id",
  filter_variable_value = valid_ids
)
```

`get_openapp_data` *Get data from OpenAppBuilder*

Description

Retrieves data from OpenAppBuilder by querying the specified PostgreSQL database. The function can either retrieve all data from a specific table (e.g., `app_users` or `app_notification_interaction`) or execute a custom SQL query provided by the user.

Usage

```
get_openapp_data(
  site = get_app_connection(),
  name = c("app_users", "app_notification_interaction"),
  filter = FALSE,
  filter_variable = NULL,
  filter_variable_value = NULL,
  qry = NULL
)
```

Arguments

`site`	The name of the PostgreSQL database connection (using `DBI::dbConnect` or `set_app_connection()`).
`name`	A character string specifying the table to retrieve data from. Default is `"app_users"`, but `"app_notification_interaction"` can also be specified. This parameter is ignored if `qry` is provided.
`filter`	A logical value indicating whether to filter the data based on a specific column (defaults to `FALSE`).
`filter_variable`	A character string representing the name of the column to filter if `filter` == `TRUE`.
`filter_variable_value`	A character string or vector representing the value(s) of the `filter_variable` column to filter if `filter` == `TRUE`.
`qry`	An optional character string containing an SQL query. If provided, this query overrides the `name` parameter and allows for custom SQL to be executed.

Value

A data frame containing the retrieved data from the specified PostgreSQL table or the result of the executed SQL query.

Examples

```
# First we need to set an app connection
set_app_connection(
  dbname = "vmc",
  host = "apps-server.idems.international",
  port = 5432,
  user = "vmc",
  password = "LSQkyYg5KzL747"
)

# Retrieve all data from the 'app_users' table
data_all_users <- get_openapp_data(name = "app_users")

# Retrieve filtered data from the 'app_users' table where 'app_user_id' is
# a specified ID.
valid_ids <- c("3e68fcda-d4cd-400e-8b12-6ddfabced348",
               "223925c7-443a-411c-aa2a-a394f991dd52")
data_filtered_users <- get_openapp_data(
  name = "app_users",
  filter = TRUE,
  filter_variable = "app_user_id",
  filter_variable_value = valid_ids
)

# Retrieve data using a custom SQL query
custom_query_data <- get_openapp_data(
  qry = "SELECT * FROM app_users WHERE app_version = '0.16.33'"
)
```

`get_user_data` *Get user data from OpenAppBuilder*

Description

Retrieves data from the `app_users` table in OpenAppBuilder, and efficiently converts the `contact_fields` column from JSON format to a data frame. If `filter` is `TRUE`, the function further filters the data to include only rows where the specified `filter_variable` column matches `filter_variable_value`.

Usage

```
get_user_data(
  site = get_app_connection(),
  filter = FALSE,
  filter_variable = NULL,
  filter_variable_value = NULL,
  date_from = NULL,
  date_to = NULL,
  format_date = "%Y-%m-%d",
  tzone_date = "UTC"
)
```

Arguments

`site`	The name of the PostgreSQL database connection (using `DBI::dbConnect` or `set_app_connection()`).
`filter`	A logical value indicating whether to filter data (defaults to `FALSE`).
`filter_variable`	A character string representing the name of the column to filter if `filter == TRUE` and `filter_variable_value` is provided.
`filter_variable_value`	A character string representing the value of the `filter_variable` column to filter if `filter == TRUE`.
`date_from`	An optional character string representing the date from which to retrieve data.
`date_to`	An optional character string representing the date to which to retrieve data.
`format_date`	A character string specifying the format of the date strings (defaults to "%Y-%m-%d").
`tzone_date`	A character string specifying the time zone for the date strings (defaults to "UTC"). System-specific (see `as.POSIXlt`), but "" uses the current time zone, and "GMT" is UTC (Universal Time, Coordinated). Invalid values are most commonly treated as UTC, on some platforms with a warning.

Value

A data frame containing user data from the PostgreSQL database, with the `contact_fields` column parsed into separate columns.

Examples

```
# First we need to set an app connection
set_app_connection(
  dbname = "vmc",
  host = "apps-server.idems.international",
  port = 5432,
  user = "vmc",
  password = "LSQkyYg5KzL747"
)

# Retrieve all data from the 'app_users' table
data_all_users <- get_user_data()

# Retrieve data from the 'app_users' table where 'app_user_id' is
# a specified ID.
valid_ids <- c("3e68fcda-d4cd-400e-8b12-6ddfabced348",
               "223925c7-443a-411c-aa2a-a394f991dd52")
data_filtered_users <- get_user_data(
  filter = TRUE,
  filter_variable = "app_user_id",
  filter_variable_value = valid_ids
)

# Retrieve user data within a specific date range
date_filtered_data <- get_user_data(
  date_from = "2023-01-01",
  date_to = "2024-08-18"
)
```

`set_app_connection` *Set Application Database Connection*

Description

Establishes a connection to a PostgreSQL database using provided credentials. This function utilises the `DBI` and `RPostgres` packages to set up the connection.

Usage

```
set_app_connection(dbname, host, port, user, password, ...)
```

Arguments

`dbname`	The name of the database to connect to.
`host`	The host name of the server where the database is located.
`port`	The port number to connect through.
`user`	The username for database authentication.
`password`	The password for database authentication.
`...`	Additional arguments passed to `DBI::dbConnect`.

Value

A database connection object of class DBIConnection.

See Also

dbConnect for more details on the underlying connection function. For additional information on database interfaces, see https://dbi.r-dbi.org/.

Examples

```
# Establish a connection to the PostgreSQL database
set_app_connection(
  dbname = "vmc",
  host = "apps-server.idems.international",
  port = 5432,
  user = "vmc",
  password = "LSQkyYg5KzL747"
)
```

Index