Package 'redquack'

July 23, 2025

Title Transfer 'REDCap' Data to Database

Version 0.2.0

Description Transfer 'REDCap' (Research Electronic Data Capture) data to a
database with additional support for 'DuckDB'. Processes data in chunks to
handle large datasets without exceeding available memory. Features include
resuming incomplete transfers, automatically converting data types for 'DuckDB',
tracking progress, and logging operations in the database.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports audio, cli, DBI, httr2, readr

Suggests arrow, dplyr, duckdb, keyring, pak, RSQLite, testthat (>=
3.0.0)

Config/testthat/edition 3

Depends R (>= 4.1.0)

NeedsCompilation no

Author Dylan Pieper [aut, cre]

Maintainer Dylan Pieper <dylanpieper@gmail.com>

Repository CRAN

Date/Publication 2025-05-14 13:40:01 UTC

Contents

---

redcap_to_db   *Transfer 'REDCap' Data to a Database*

Description

Transfer REDCap data to a database in chunks to minimize memory usage.

Usage

```
redcap_to_db(
  conn,
  redcap_uri,
  token,
  data_table_name = "data",
  log_table_name = "log",
  raw_or_label = "raw",
  raw_or_label_headers = "raw",
  export_checkbox_label = FALSE,
  export_survey_fields = FALSE,
  export_data_access_groups = FALSE,
  blank_for_gray_form_status = FALSE,
  filter_logic = "",
  datetime_range_begin = as.POSIXct(NA),
  datetime_range_end = as.POSIXct(NA),
  fields = NULL,
  forms = NULL,
  events = NULL,
  record_id_name = "record_id",
  chunk_size = 1000,
  chunk_delay = 0.5,
  max_retries = 10,
  verbose = TRUE,
  beep = TRUE,
  ...
)
```

Arguments

conn	A DBI connection object to a database.
redcap_uri	Character string specifying the URI (uniform resource identifier) of the REDCap server's API.
token	Character string containing the REDCap API token specific to your project. This token is used for authentication and must have export permissions.
data_table_name	
	Character string specifying the name of the table to create or append data to. Default is "data". Can include schema name (e.g. "schema.table").

`log_table_name`	Character string specifying the name of the table to store transfer logs. Default is "log". Can include schema name (e.g. "schema.log"). Set to NULL to disable logging.
`raw_or_label`	A string (either "raw" or "label") that specifies whether to export the raw coded values or the labels for the options of multiple choice fields. Default is "raw".
`raw_or_label_headers`	
	A string (either "raw" or "label") that specifies for the CSV headers whether to export the variable/field names (raw) or the field labels (label). Default is "raw".
`export_checkbox_label`	
	Logical that specifies the format of checkbox field values specifically when exporting the data as labels. If `raw_or_label` is "label" and `export_checkbox_label` is TRUE, the values will be the text displayed to the users. Otherwise, the values will be 0/1. Default is FALSE.
`export_survey_fields`	
	Logical that specifies whether to export the survey identifier field (e.g., 'redcap_survey_identifier') or survey timestamp fields. Default is FALSE.
`export_data_access_groups`	
	Logical that specifies whether or not to export the `redcap_data_access_group` field when data access groups are utilized in the project. Default is FALSE.
`blank_for_gray_form_status`	
	Logical that specifies whether or not to export blank values for instrument complete status fields that have a gray status icon. Default is FALSE.
`filter_logic`	String of logic text (e.g., `[gender] = 'male')` for filtering the data to be returned, where the API will only return records where the logic evaluates as TRUE. Default is an empty string.
`datetime_range_begin`	
	To return only records that have been created or modified *after* a given datetime, provide a POSIXct value. Default is NA (no begin time).
`datetime_range_end`	
	To return only records that have been created or modified *before* a given datetime, provide a POSIXct value. Default is NA (no end time).
`fields`	Character vector specifying which fields to export. Default is NULL (all fields).
`forms`	Character vector specifying which forms to export. Default is NULL (all forms).
`events`	Character vector specifying which events to export. Default is NULL (all events).
`record_id_name`	Character string specifying the field name that contains record identifiers used for chunking requests. Default is "record_id".
`chunk_size`	Integer specifying the number of record IDs to process per chunk. Default is 1000. Consider decreasing this for projects with many fields.
`chunk_delay`	Numeric value specifying the delay in seconds between chunked requests. Default is 0.5 seconds. Adjust to respect REDCap server limits.
`max_retries`	Integer specifying the maximum number of retry attempts for failed API connection or HTTP 504 error. Default is 10.
`verbose`	Logical indicating whether to show progress and completion messages. Default is TRUE.

beep   Logical indicating whether to play sound notifications when the process com-
              pletes or encounters errors. Default is TRUE.

...   Additional arguments passed to the REDCap API call.

Details

This function transfers data from REDCap to any database in chunks, which helps manage memory
usage when dealing with large projects. It creates two tables in the database:

- `data_table_name`: Contains all transferred REDCap records

- `log_table_name`: Contains timestamped logs of the transfer process

The function automatically detects existing databases and handles them in three ways:

- If no table exists, starts a new transfer process

- If a table exists but is incomplete, resumes from the last processed record ID

- If a table exists and is complete, returns success without reprocessing

The function fetches record IDs, then processes records in chunks. If any error occurs during
processing, the function will continue with remaining chunks but mark the transfer as incomplete.

Data is first set to **VARCHAR/TEXT** type for consistent handling across chunks. For DuckDB,
data types are automatically optimized after all data is inserted:

- **INTEGER**: Columns with only whole numbers

- **DOUBLE**: Columns with decimal numbers

- **DATE**: Columns with valid dates

- **TIMESTAMP**: Columns with valid timestamps

- **VARCHAR/TEXT**: All other columns remain as strings

Value

Returns a list with the following components:

- `success`: Logical if the transfer was completed with no failed processing

- `error_chunks`: Vector of chunk numbers that failed processing

- `time_s`: Numeric value for total seconds to transfer and optimize data

See Also

[dbConnect](#) for database connection details

Examples

```
## Not run:
# install.packages("pak")
# pak::pak(c("duckdb", "keyring", "redquack"))

library(redquack)

duckdb <- DBI::dbConnect(duckdb::duckdb(), "redcap.duckdb")

result <- redcap_to_db(
  redcap_uri = "https://redcap.example.org/api/",
  token = keyring::key_get("redcap_token"),
  conn = duckdb,
)

data <- DBI::dbGetQuery(duckdb, "SELECT * FROM data LIMIT 1000")
log <- DBI::dbGetQuery(duckdb, "SELECT * FROM log")

DBI::dbDisconnect(duckdb)

## End(Not run)
```

Index