

Package: redivis (via r-universe)

October 13, 2026

Type Package

Title R interface for Redivis

Version 0.13.1

Description Supports working with Redivis API through R. Parse metadata, execute queries, and load data into R.

License MIT + file LICENSE

URL <https://github.com/redivis/redivis-r>

BugReports <https://github.com/redivis/redivis-r/issues>

Depends R (>= 3.5.0)

Imports R6 (>= 2.5.0), httr2 (>= 1.0.0), curl (>= 6.1.0), jsonlite (>= 1.8.0), digest (>= 0.1.0), data.table (>= 1.0.0), stringr (>= 1.4.0), tibble (>= 3.0.0), arrow (>= 16.1.0), openssl (>= 2.0.0), purrr (>= 0.3.0), progressr (>= 0.9.0), furr (>= 0.2.0), future (>= 1.21.0), parallelly (>= 1.21.0), rlang (>= 1.0.0), uuid (>= 1.0.0), later (>= 1.3.0), methods, stats, utils, DBI (>= 1.1.0), dbplyr (>= 2.3.0)

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

Suggests callr, dplyr (>= 1.0.0), haven, httpuv, IRdisplay, sf (>= 1.0), testthat (>= 3.1.7), withr

Config/testthat/edition 3

Collate 'util.R' 'auth.R' 'api_request.R' 'retryable_download.R' 'File.R' 'Directory.R' 'read_rows.R' 'TabularReader.R' 'Export.R' 'Parameter.R' 'Transform.R' 'Secret.R' 'Version.R' 'Organization.R' 'Dataset.R' 'User.R' 'Notebook.R' 'Datasource.R' 'Workflow.R' 'Table.R' 'Variable.R' 'Query.R' 'DBI.R' 'ReadStream.R' 'Upload.R' 'exceptions.R' 'index.R' 'retryable_upload.R' 'zzz.R'

Config/pak/sysreqs cmake libicu-dev libssl-dev

Repository <https://langcog.r-universe.dev>
Date/Publication 2026-10-13 00:41:41 UTC
RemoteUrl <https://github.com/redivis/redivis-r>
RemoteRef HEAD
RemoteSha cf428c042ac07ff194321be25bba9b29920eefef

Contents

current_notebook	2
open.File	3
organization	3
query	4
redivis	5
RedivisDBI	7
user	7
Index	9

current_notebook	<i>current_notebook</i>
------------------	-------------------------

Description

DEPRECATED: please use redivis\$current_notebook

Usage

current_notebook()

Value

class<Notebook>

Examples

```
## Not run:  
redivis::current_notebook()$create_output_table(df)  
  
## End(Not run)
```

open.File	<i>Open a connection to a Redis file</i>
-----------	--

Description

Makes `open(file)` behave like `file$open()`, returning a read-only connection backed by HTTP range requests.

Usage

```
## S3 method for class 'File'
open(con, mode = "r", ...)
```

Arguments

<code>con</code>	A Redis File object
<code>mode</code>	Character string. "r" (default) for text mode, "rb" for binary mode.
<code>...</code>	Ignored (present for compatibility with the generic)

Value

A readable connection backed by HTTP range requests

organization	<i>organization</i>
--------------	---------------------

Description

DEPRECATED: please use `redis$organization`

Usage

```
organization(name)
```

Arguments

<code>name</code>	The organization's username
-------------------	-----------------------------

Value

class<user>

Examples

```
## Not run:
redisvis::organization('demo_organization')$dataset('some_dataset')$table('a_table')$to_tibble(max_results=100)

# We can also construct a query scoped to a particular dataset, removing the need to fully qualify table names
redisvis::organization('demo_organization')$dataset('some_dataset')$query("SELECT * FROM table_1 INNER JOIN table_2 ON table_1.id = table_2.id")

## End(Not run)
```

query	<i>query</i>
-------	--------------

Description

DEPRECATED: please use redisvis\$query

Usage

```
query(query = "", default_workflow = NULL, default_dataset = NULL)
```

Arguments

- query The query string to execute
- default_workflow Optional workflow reference used to resolve unqualified table references. Defaults to the REDIVIS_DEFAULT_WORKFLOW environment variable.
- default_dataset Optional dataset reference used to resolve unqualified table references. Defaults to the REDIVIS_DEFAULT_DATASET environment variable.

Value

```
class<Query>
```

Examples

```
## Not run:
output_table <- redisvis::query(query = 'SELECT 1 + 1 AS two')$to_tibble()

## End(Not run)
```

redivis

*redivis***Description**

Redivis wrapper, all primary constructor accessible via `redivis$<constructor>()`

Full documentation and examples available at: <https://apidocs.redivis.com/client-libraries/redivis-r>

Usage

```
redivis
```

Format

An object of class environment of length 17.

Value

`list(authenticate, current_notebook, current_user, current_workflow, dataset, datasource, file, make_api_request, notebook, organization, parameter, query, table, transform, user, workflow)`

Overview

```
library(redivis)

# Reference various resources
redivis$current_user()
redivis$dataset(name)
redivis$datasource(source)
redivis$notebook(name)
redivis$organization(name)
redivis$parameter(name)
redivis$table(name)
redivis$transform(name)
redivis$user(name)
redivis$workflow(name)

# Create a SQL query
redivis$query(query)

# Only available within Redivis notebooks
redivis$current_notebook()
redivis$current_workflow()
user$secret("MY_SECRET")$get_value()
organization$secret("OUR_SECRET")$get_value()

# Read data
```

```

query$to_tibble()
table$to_tibble()
# also:
#   to_data_table()
#   to_arrow_table()
#   to_arrow_batch_iterator()
#   to_sf_tibble()
#   ...
table$download(path=NULL, format="csv", ...)

# DBI interface
con <- redivis$connect_dbi()

# Read files (file tables only):
table$file("path/to/file")
open(file)
file$read(...)
file$download(path=NULL, ...)
dir <- table$to_directory()
dir$get("/path/to/file/or/dir")
dir$download(path=NULL, ...)

# Upload data
table$upload()$create("/path/to/file.csv", ...)
# Upload raw files
table$add_files(directory="/path/to/dir", ...)

# List resources
organization$list_datasets(max_results)
user$list_datasets(max_results)
user$list_workflows(max_results)
dataset$list_tables(max_results)
workflow$list_tables(max_results, include_datasources=FALSE)
workflow$list_transforms(max_results)
table$list_variables(max_results)
#...etc, see documentation

# Miscellaneous methods, for advanced use cases
redivis$authenticate(scope=NULL, force_reauthentication=FALSE)
redivis$make_api_request(method="GET", path, ...)

```

Examples

```

## Not run:
df <- redivis$table("demo.ghcn_daily_weather_data.stations")$to_tibble()
df <- redivis$query("SELECT 1+1 as two")$to_tibble()

redivis$dataset("demo.ghcn_daily_weather_data")$exists() # TRUE
datasets <- redivis$organization("demo")$list_datasets(max_results=10)

```

```
workflows <- redivis$current_user()$list_workflows(10)

# Only callable from within a Redivis notebook
notebook <- redivis$current_notebook();
workflow <- redivis$current_workflow();

## End(Not run)
```

RedivisDBI	Create a Redivis DBI driver
------------	-----------------------------

Description

Create a Redivis DBI driver

Usage

```
RedivisDBI()
```

Value

A RedivisDriver object

user	<i>user</i>
------	-------------

Description

DEPRECATED: please use redivis\$user

Usage

```
user(name)
```

Arguments

name	The user's username
------	---------------------

Value

```
class<user>
```

Examples

```
## Not run:
redis::user('my_username')$workflow('my_workflow')$table('my_table')$to_tibble(max_results=100)

# We can also construct a query scoped to a particular workflow, removing the need to fully qualify table names
redis::user('my_username')$workflow('my_workflow')$query("SELECT * FROM table_1 INNER JOIN table_2 ON id")$to_tibble(max_results=100)

## End(Not run)
```

Index

* **datasets**

 redivis, [5](#)

current_notebook, [2](#)

open.File, [3](#)

organization, [3](#)

query, [4](#)

redivis, [5](#)

RedivisDBI, [7](#)

user, [7](#)