

Package ‘basedosdados’

April 26, 2021

Title 'Base Dos Dados' R Client

Version 0.0.5

Description An R interface to the 'Base dos Da-
dos' API <https://basedosdados.github.io/mais/py_reference_api/>. Authenti-
cate your project, query our tables, save data to disk and memory, all from R.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Imports purrr, dplyr, tibble, httr, googleAuthR, cli, magrittr, readr,
stringr, dotenv, bigrquery, glue, rlang, writexl

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Pedro Cavalcante [aut, cre]

Maintainer Pedro Cavalcante <pedrocolrj@gmail.com>

Repository CRAN

Date/Publication 2021-04-26 06:30:10 UTC

R topics documented:

download	2
get_billing_id	3
read_sql	3
set_billing_id	4
Index	6

 download

Write the results of a query locally to a comma-separated file.

Description

Write the results of a query locally to a comma-separated file.

Usage

```
download(query, path, billing_project_id = get_billing_id(), page_size = 1000)
```

Arguments

query	a string containing a valid SQL query.
path	String with the output file's name. If running an R Project relative location can be provided. Passed to <code>readr::write_csv</code> 's file argument.
billing_project_id	a string containing your billing project id. If you've run <code>set_billing_id</code> then feel free to leave this empty.
page_size	bigrquery internal, how many rows per page should there be.

Details

Currently there's only support for UTF-8 encoding. Users requiring more control over writing should use `read_sql` to get the data in memory and custom code from there.

Value

Invisibly returns the query's output in a tibble. Intended to be used for side-effects. If you simply want to load a query's result in memory, use `read_sql`.

Examples

```
## Not run:

dir <- tempdir()

query <- "SELECT
pib.id_municipio,
pop.ano,
pib.PIB / pop.populacao * 1000 as pib_per_capita
FROM `basedosdados.br_ibge_pib.municipios` as pib
JOIN `basedosdados.br_ibge_populacao.municipios` as pop
ON pib.id_municipio = pop.id_municipio
LIMIT 5 "
```

```
data <- download(query, file.path(dir, "pib_per_capita.csv"))
```

```
## End(Not run)
```

get_billing_id *Internal functions for project billing management*

Description

Retrieves the project's billing Id.

Usage

```
get_billing_id()
```

Value

a string with the project's billing id.

read_sql *Query out datalake and get results in a tibble*

Description

Query out datalake and get results in a tibble

Usage

```
read_sql(query, billing_project_id = get_billing_id(), page_size = 1000)
```

Arguments

- `query` a string containing a valid SQL query.
- `billing_project_id` a string containing your billing project id. If you've run `set_billing_id` then feel free to leave this empty.
- `page_size` bigrquery internal, how many rows per page should there be.

Value

A tibble containing the query's output.

Examples

```
## Not run:

set_billing_id("<your id here>")

query <- "SELECT
pib.id_municipio,
pop.ano,
pib.PIB / pop.populacao * 1000 as pib_per_capita
FROM `basedosdados.br_ibge_pib.municipios` as pib
JOIN `basedosdados.br_ibge_populacao.municipios` as pop
ON pib.id_municipio = pop.id_municipio
LIMIT 5 "

data <- read_sql(query)

# in case you want to write your data on disk as a .xlsx, .csv or .Rds file.

library(writexl)
library(readr)

dir <- tempdir()

write_xlsx(data, file.path(dir, "data.xlsx"))
write_csv(data, file.path(dir, "data.csv"))
saveRDS(data, file.path(dir, "data.Rds"))

## End(Not run)
```

set_billing_id	<i>Define your Project Id</i>
----------------	-------------------------------

Description

Define your project billing ids here so all your queries are authenticated and return data, not errors. If using in production or leaving code available at public repositories, dotenv is highly recommended.

Usage

```
set_billing_id(billing_project_id)
```

Arguments

```
billing_project_id
```

a single character value containing the string. Vectors with longer lengths and non-vectors will trigger an error.

Value

No return.

Examples

```
## Not run:  
set_billing_id("my_billing_project_id")  
  
# or load from an .env file  
  
library(dotenv)  
  
load_dot_env("keys.env")  
print(Sys.getenv("billing_project_id"))  
  
set_billing_id(Sys.getenv("billing_project_id"))  
  
## End(Not run)
```

Index

download, [2](#)

get_billing_id, [3](#)

read_sql, [3](#)

set_billing_id, [4](#)