RIBBiTR Database Connection Setup

Cob Staines

2024-10-28

Table of contents

Motivation

- Connect to the RIBBiTR (or another remote) database with ease and repeatability
- Manage login credentials for ease and security, so they won't be lost or shared with your code.

Database Connection setup for RStudio

Load packages

Using "librarian"

"librarian" is a package and library management package in R which makes it easier to install, load, update and unload packages to meet dynamic environment needs. There are other ways to download, load, and maintain packages in R (e.g. "install.packages()" and "library()", but we recommend librarian for its simplicity and portability.

```
# install and load "librarian" package if not already required
if (!require(librarian)) {
  install.packages("librarian")
  library(librarian)
}
```

librarian downloads and loads packages using the "librarian::shelf" function. Below are the minimal recommended packages to establish a connection to the RIBBiTR database.

```
# minimal packages for establishing RIBBiTR DB connection
shelf(tidyverse, RPostgres, DBI, usethis, RIBBiTR-BII/ribbitrrr)
```

Defining your database connection parameters

Access your local .Renviron file

Your .Renviron file a local file where you can save and reference your login credentials for easy use within R and RStudio, without risking losing them or potentially sharing them on accident when you share your code.

```
# open your local .Reniron file
usethis::edit_r_environ()
```

Save connections parameters locally

Copy the following database connection parameters to your .Renviron file, substituting your login credentials (user & password).

```
# RIBBiTR DB credentials
ribbitr.dbname = "ribbitr"
ribbitr.host = "ribbitr.c6p56tuocn5n.us-west-1.rds.amazonaws.com"
ribbitr.port = "5432"
ribbitr.user = "[YOUR-LOGIN-HERE]"
ribbitr.password = "[YOUR-PASSWORD-HERE]"
```

Save and close .Renviron, and restart RStudio.

Establish database connection

Now, using the ribbitrrr:HopToDB() function, let's establish a connection!

```
# establish database connection
dbcon = HopToDB("ribbitr")
```

```
Connecting to database... Success!
```

HopToDB() returns a database connection object ("dbcon"). Keep track of this, you will need it to explore and pull data later.

Begin using your connection!

Try out your connection by loading table metadata from the database

```
mdt = tbl(dbcon, Id("public", "all_tables")) %>%
  collect()
head(mdt)
```

```
# A tibble: 6 x 4
  table_schema table_name         column_count table_description
  <chr>        <chr>                     <int64> <chr>
1 bay_area     amphib_dissect               41 <NA>
2 bay_area     amphib_parasite              11 <NA>
3 bay_area     water_quality_info           27 <NA>
4 bay_area     site                         25 <NA>
5 bay_area     wetland_info                 25 <NA>
6 bay_area     bd_results                   25 <NA>
```

Also try

- For those managing multiple database connections, the HopToDB() function allows you to store and fetch various sets of login credentials with a single keyword. Just substitute "ribbitr" in the example above with your own keyword!
- Your login credentials can also be accessed explicitly anytime using Sys.getenv("ribbitr.dbname"), etc. In most cases the HopToDB() function is all you need, however.