Hacking RStudio

Part 4: Connections

Colin Fay - ThinkR

What are RStudio Connections?

Connections are a way to extend RStudio connection features, so that you can connect to external data sources easily.

Connections can be handled through:

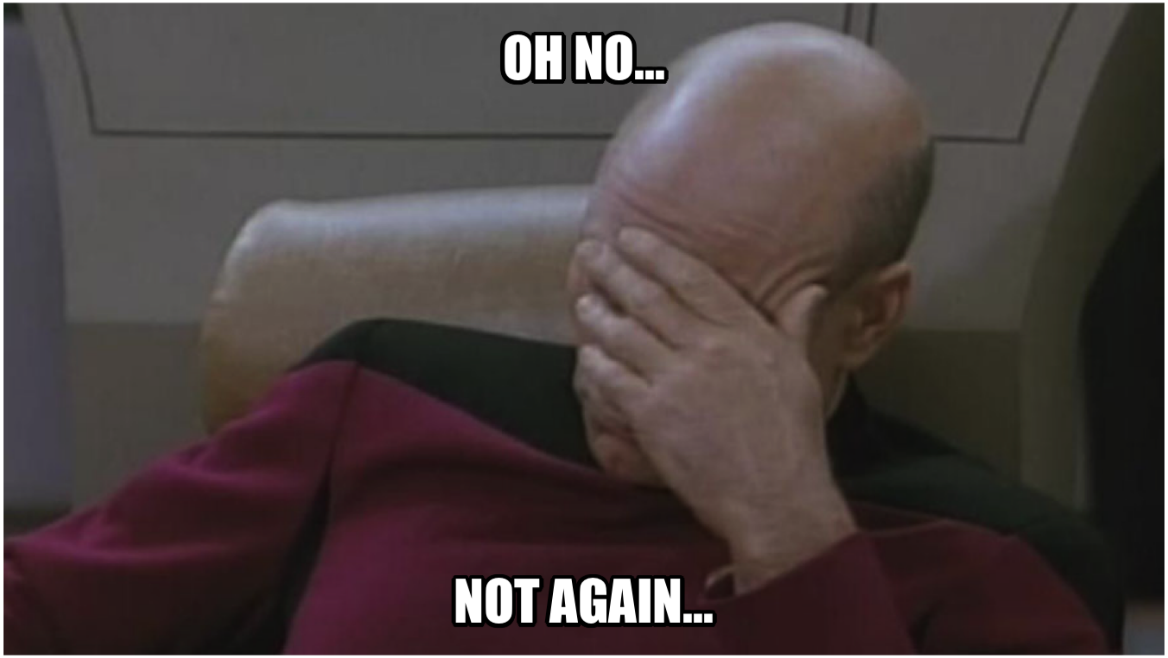
- Snippet Files, which are read with the Connection Pane
- Packages that can implement either Snippet Files or Shiny Applications
- Package can use Connection Contracts

RStudio Connections example

- `{sparklyr}` - R interface for Apache Spark
- `{odbc}` - Connect to ODBC databases (using the DBI interface)
- `{neo4r}` - A Modern and Flexible Neo4J Driver
- `{fryingpane}` - Serve datasets from a package inside the RStudio Connection Pane.

RStudio Connections

Step 1… Create a package!



Creating a snippet

```r
dir.create(path = "inst/rstudio/connections", recursive = TRUE)
file.create(path = "inst/rstudio/connections.dcf")
file.create(path = "inst/rstudio/connections/snippet.R")
rstudioapi::navigateToFile("inst/rstudio/connections.dcf")
rstudioapi::navigateToFile("inst/rstudio/connections/snippet.R")
```

`connections.dcf`

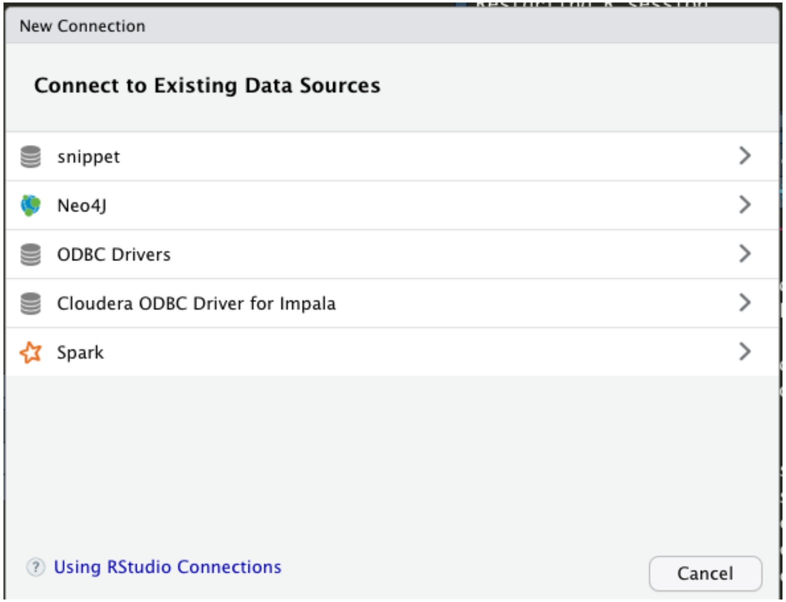
A `dcf` containing a series of `Name` refering to the `.R` files in the `connections/` folder.

```
Name: snippet
```

snippet.R

```r
library(readr)
mtcars <- read_csv(readr_example("mtcars.csv"))
```

Install and Restart RStudio

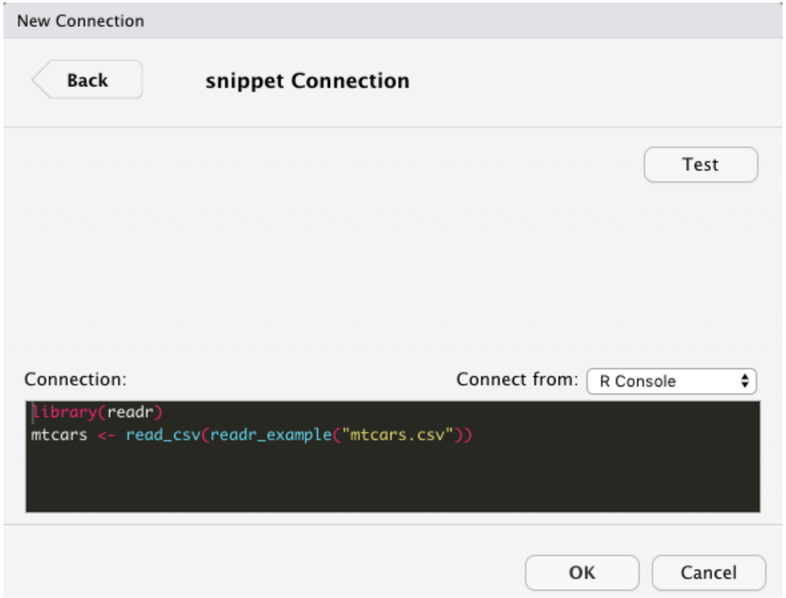


New Connection

Connect to Existing Data Sources

- snippet
- Neo4J
- ODBC Drivers
- Cloudera ODBC Driver for Impala
- Spark

Using RStudio Connections



New Connection

snippet Connection

Connection: Connect from: R Console

```
library(readr)
mtcars <- read_csv(readr_example("mtcars.csv"))
```

connections.dcf

Name: snippet

Name: more_snippet

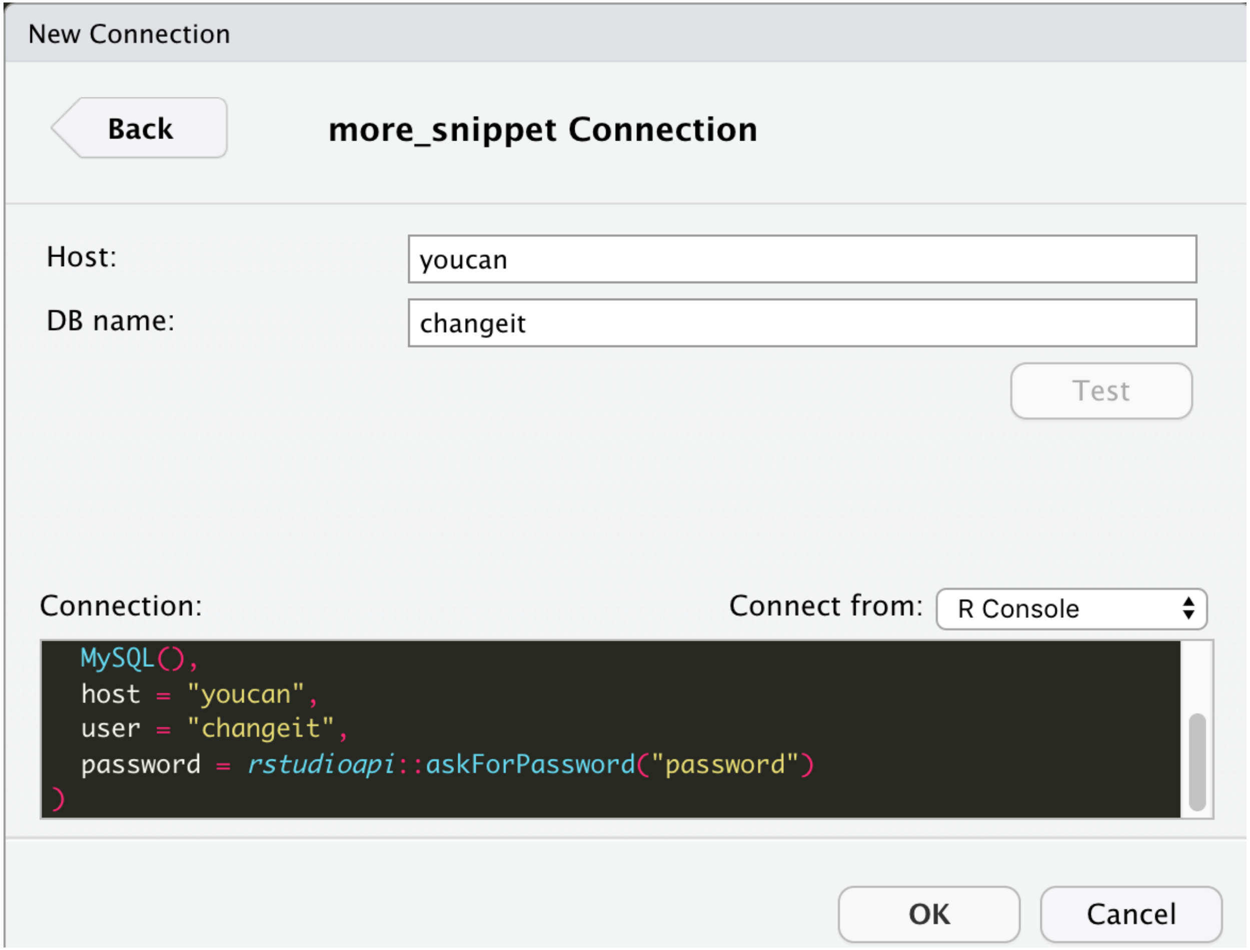
```r
file.create(path = "inst/rstudio/connections/more_snippet.R")
rstudioapi::navigateToFile("inst/rstudio/connections/more_snippet.R")
```

`more_snippet.R`

> Can pass parameters with `${Position:Label=Default}`

```r
library(DBI)
library(RMySQL)
con <- dbConnect(
MySQL(),
host = ${0:Host="mydb"},
user = ${1:DB name="colin"},
password = rstudioapi::askForPassword("password")
)
```

Install and Restart RStudio



New Connection

more_snippet Connection

Host: youcan

DB name: changeit

Connection:

Connect from: R Console

```
  MySQL(),
  host = "youcan",
  user = "changeit",
  password = rstudioapi::askForPassword("password")
)
```

Connection contract

Connection contracts are a way to interact with data sources and display an interface in the Connection Pane.

The Connection Pane is handled with the `connectionObserver` from RStudio.

You can get it with:

```
observer <- getOption("connectionObserver")
```

The observer has three methods, to be called on creation, update, and closing of the Connection with the data source.

- `connectionOpened()`
- `connectionClosed()`
- `connectionUpdated()`

`connectionOpened()`

This method takes several arguments:

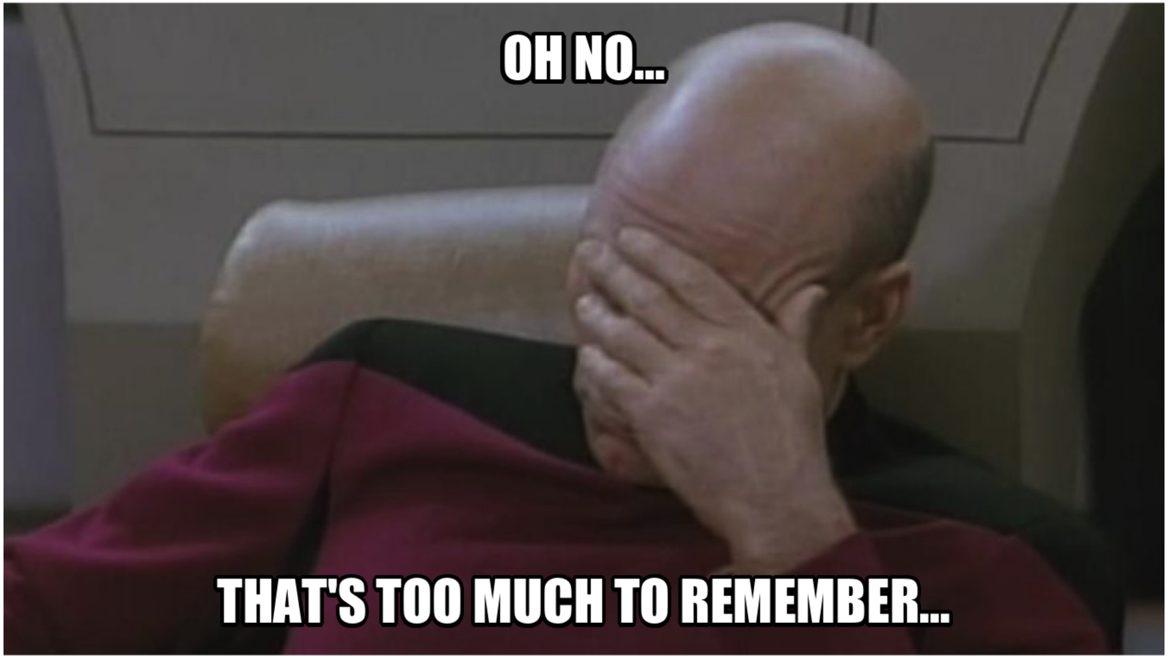
- `type` & `displayName`, texts describing the connection
- `host`, the address of the host we are connected to
- `icon`, the path to an icon for the connection pane
- `connectCode`, a snippet of R code to be reused in the connection Pane
- `disconnect`, a *function* used to close the connection from the connection Pane
- `listObjectTypes`, a *function* that returns a nested list from the connection
- `listObjects`, a *function* that list top level objects from the data source when called without arg, or the object to retreive. Returned object is a dataframe with `name` and `type` column
- `listColumns`, a *function* listing the columns of a data object. Returned object is a dataframe with `name` and `type` column

`connectionOpened()`

This method takes several arguments:

- `previewObject`, a *function* accepting row limit and an object, returns the specified number of rows from the object
- `actions`, a list of things to add to the connection pane as buttons. Lists with `icon` and `callback`
- `connectionObject`, the raw connection object

Too much?



Step 1: define a `on_connection_opened` fun

```r
on_connection_opened <- function(pkg_name = "pkg") {
  data_list  <- data(package = pkg_name)
  data_results <- as.data.frame(data_list$results)
  observer <- getOption("connectionObserver")
  if(!is.null(observer)){
    observer$connectionOpened()
  }
}
```

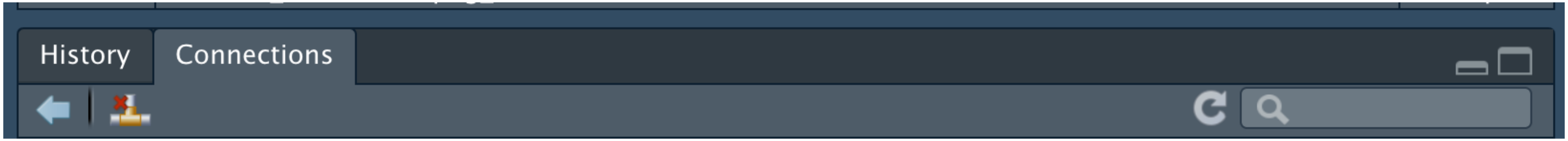
Step 2: define metadata

```
observer$connectionOpened(
  type = "Data sets",
  host = pkg_name,
  displayName = pkg_name,
  icon = system.file("img","package.png", package = "fryingpane"),
  connectCode = glue('library(fryingpane)\ncook("{pkg_name}")')
)
```

Step 3 function to call on connection closed

```
observer$connectionOpened(
  disconnect = function() {
    close_connection(pkg_name)
  }
)
```

```
close_connection <- function(pkg_name) {
  # Deconnection logic
  print("Connection closed")
}
```



Step 4 List object types

```r
observer$connectionOpened(
  listObjectTypes = function() {
    list_objects_types()
  }
)
```

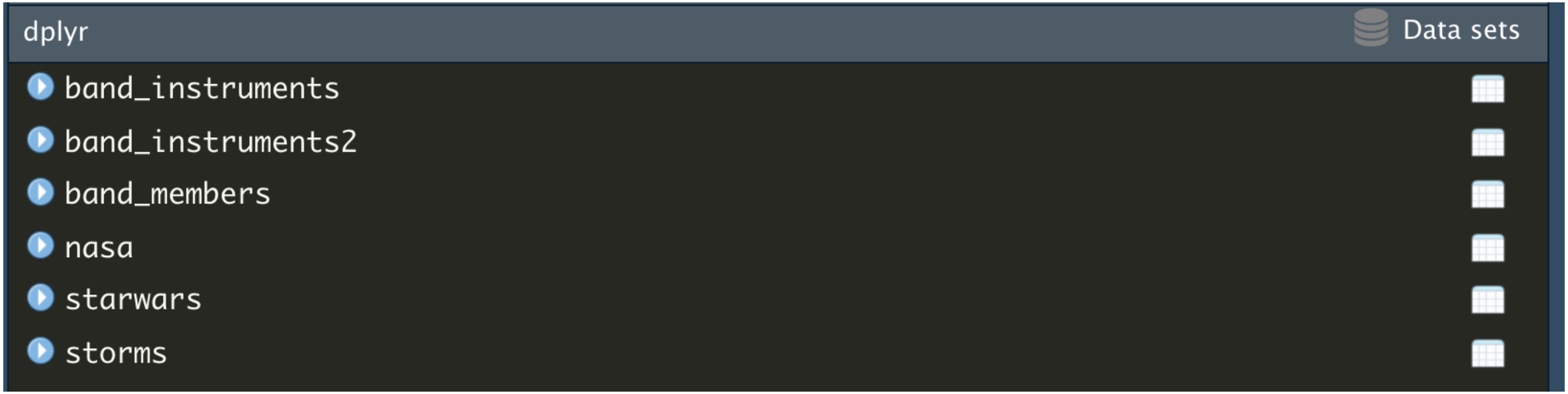
```r
list_objects_types <- function() {
  return(
    list(
      table = list(contains = "data")
    )
  )
}
```

Step 5 How are object displayed?

```r
observer$connectionOpened(
  listObjects = function( type = "table") {
    list_objects(
      includeType = TRUE,
      data_names = as.character(data_results$Item),
      data_type = "dataset"
    )
  }
)
```

Step 5 How are object displayed?

```
list_objects <- function(includeType, data_names, data_type) {
  if (includeType) {
    data.frame(
      name = data_names,
      type = rep_len("table", length(data_names)),
      stringsAsFactors = FALSE
    )
  }
}
```



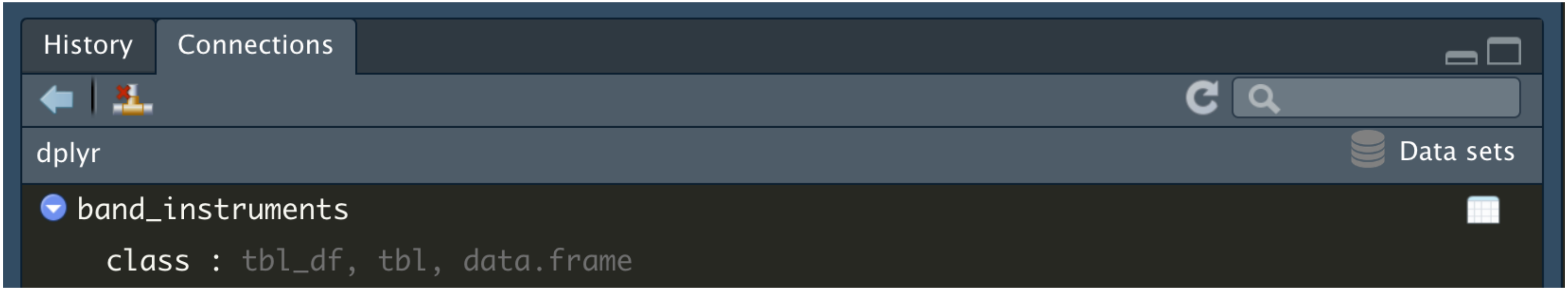
dplyr
Data sets

band_instruments
band_instruments2
band_members
nasa
starwars
storms

Step 6 List Colums

```
observer$connectionOpened(
  listColumns = function(table) {
    list_columns(table, pkg_name = pkg_name)
  }
)
```

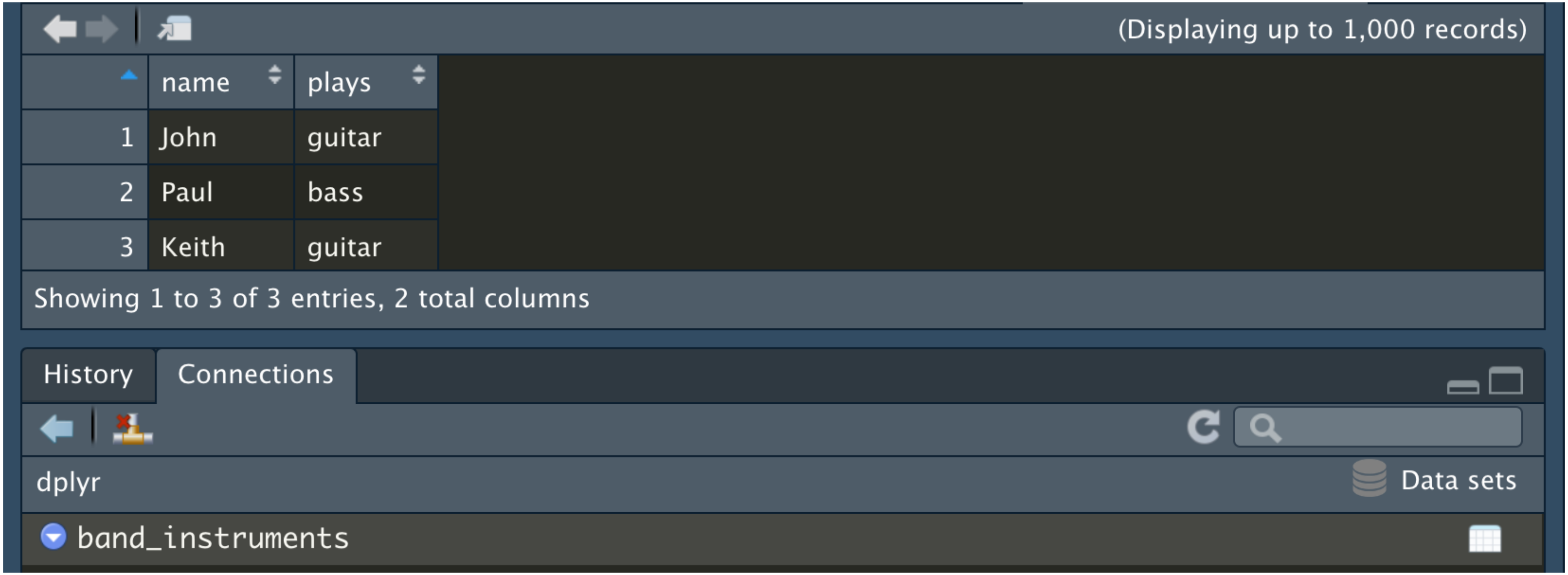
```
list_columns <- function(table, pkg_name) {
  res <- get(data(list = table, package = pkg_name))
  tibble(
    name = "class",
    type = paste(class(res), collapse = ", "))
}
```



```
band_instruments
   class : tbl_df, tbl, data.frame
```

Step 7 Preview objects

```
observer$connectionOpened(
  previewObject = function(rowLimit, table) {
    preview_object(pkg_name = pkg_name, table, rowLimit)
  }
)
```



(Displaying up to 1,000 records)

	name	plays
1	John	guitar
2	Paul	bass
3	Keith	guitar

Showing 1 to 3 of 3 entries, 2 total columns

Step 8 Actions

```r
observer$connectionOpened(
  actions = pkg_actions(pkg_name)
)
```

```r
pkg_actions <- function(pkg_name){
  list(
    help = list(
      icon = system.file("icons","github.png", package = "neo4r"),
      callback = function() {
        help(pkg_name, try.all.packages = TRUE, help_type = "text")
      }
    )
  )
}
```

Finally, wrap this in a function !

```
cook <- function(pkg_name){
  test_if_exists(pkg_name)
  on_connection_opened(pkg_name)
}

test_if_exists <- function(pkg_name){
  stop_if(find.package(pkg_name, quiet = TRUE),
          ~ length(.x) == 0,
          glue("{pkg_name} wasn't found on the machine."))
}
```