

Package ‘podr’

October 27, 2020

Type Package

Title Common Functions for 'PHUSE' Open Data Repository

Version 0.0.5

Author Hanming Tu [aut, cre]

Maintainer Hanming Tu <hanming.tu@gmail.com>

Description Make it easy to connect, access and review datasets in Pharmaceutical User Software Exchange ('PHUSE') open data repository ('PODR'). The R Shiny application included in this package allow users to connect to 'PODR' and access to over 260 open data sets.

Depends R (>= 3.5.0)

License MIT + file LICENSE

URL <https://github.com/TuCai/podr>

BugReports <https://github.com/TuCai/podr/issues>

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports DBI (>= 1.1.0), assertthat (>= 0.2.1), dplyr (>= 0.8.4), lubridate (>= 1.7.9), magrittr (>= 1.5), odbc (>= 1.2.3), plogr (>= 0.2.0), rstudioapi (>= 0.10), shiny (>= 1.3.2), stringr (>= 1.4.0), tibble (>= 3.0.3), RPostgres (>= 1.2.1)

Suggests DT (>= 0.15), haven (>= 2.2.0), huxtable (>= 5.1.0), knitr (>= 1.23), kableExtra (>= 1.2.1), pharmaRTF (>= 0.1.1), rmarkdown (>= 2.3), readr (>= 1.3.1), shinyBS (>= 0.61), shinydashboard (>= 0.7.1), shinyjs (>= 2.0.0), testthat (>= 2.1.0), tidyverse (>= 1.3.0)

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2020-10-27 15:10:03 UTC

R topics documented:

build_where_clause	2
conn_podr	3
echo_msg	4
get_table_defs	5
get_table_names	6
is_empty	7
read_podr	7
resolve	8
start_app	9

Index	11
--------------	-----------

build_where_clause	<i>Build Where Clause</i>
--------------------	---------------------------

Description

This function builds a where clause for a SQL statement

Usage

```
build_where_clause(
  tab_name,
  col_name,
  col_value,
  con = getOption("podr_connection")
)
```

Arguments

tab_name	table name
col_name	column name
col_value	column value
con	The connection to PODR. Use connect_podr to establish a connection, or specify a variable containing the proper PostgreSQL connection into PODR yourself

Value

column definition in a data.frame

Author(s)

Hanming Tu

Examples

```
## Not run:
cc <- conn_podr()
build_where_clause('css_2020_ae', con=cc)

## End(Not run)
```

conn_podr

Make a connection to PHUSE Open Data Repository (PODR)

Description

This function will set a PODR connection within the 'podr_connection' option. Ideally, you shouldn't need to worry about this after it's set. Using the read_podr function will utilize the connection automatically by default.

Usage

```
conn_podr(
  username = NULL,
  userpwd = NULL,
  driver = "postgresql",
  server = "podr.phuse.global",
  port = "5432",
  database = "nihpo"
)
```

Arguments

username	provides user name for PODR database connection given as a character string
userpwd	provides user password for PODR user given as a character string
driver	is database driver defaults to "postgresql"
server	is server name in FQDN or IP address defaults to "podr.phuse.global"
port	is port number defaults to "5432"
database	is database name defaults to "nihpo"

Details

You will be prompted within the RStudio interface for your user name and password if you did not provides when setting the connection,

Value

Nothing - sets the 'podr_connection' option

Author(s)

Hanming Tu

Examples

```
## Not run:
  conn_podr() # require user to provide name and password interactively
  conn_podr('my_username','my_pwd')

## End(Not run)
```

echo_msg
*Echo message***Description**

This method displays or writes the message based on debug level. The filehandler is provided through environment variable 'log_fn', and the outputs are written to the file. This method will display message or a hash array based on debug level ('d_level'). If 'd_level' is set to '0', no message or array will be displayed. If 'd_level' is set to '2', it will only display the message level (lvl) is less than or equal to '2'. If you call this method without providing a message level, the message level (lvl) is default to '0'. Of course, if no message is provided to the method, it will be quietly returned. If 'd_level' is set to '1', all the messages with default message level, i.e., 0, and '1' will be displayed. The higher level messages will not be displayed.

Usage

```
echo_msg(prg, step, msg, lvl = 0, fn = NULL)
```

Arguments

prg	program name calling from
step	step in the program
msg	the message to be displayed. No newline is needed in the end of the message. It will add the newline code at the end of the message.
lvl	the message level is assigned to the message. If it is higher than the debug level, then the message will not be displayed.
fn	log file name

Value

message

Author(s)

Hanming Tu

Examples

```
NULL;
```

get_table_defs	<i>Get Table Definition from PODR</i>
----------------	---------------------------------------

Description

This function will get column definition for a table in "PODR".

Usage

```
get_table_defs(tab_name, con = getOption("podr_connection"))
```

Arguments

tab_name	table name
con	The connection to PODR. Use connect_podr to establish a connection, or specify a variable containing the proper PostgreSQL connection into PODR yourself

Value

column definition in a data.frame

Author(s)

Hanming Tu

Examples

```
## Not run:  
cc <- conn_podr()  
get_table_defs('css_2020_ae', con=cc)  
  
## End(Not run)
```

get_table_names	<i>Get Table Names from PODR</i>
-----------------	----------------------------------

Description

This function will get a list of table names from "PODR".

Usage

```
get_table_names(  
  con = getOption("podr_connection"),  
  query_string = NULL,  
  lib_sel = NULL,  
  ds_sel = NULL  
)
```

Arguments

con	The connection to PODR. Use connect_podr to establish a connection, or specify a variable containing the proper PostgreSQL connection into PODR yourself
query_string	provides full SQL statement
lib_sel	Library name selection pattern
ds_sel	Dataset name selection pattern

Value

The selected table names in a data.frame

Author(s)

Hanming Tu

Examples

```
## Not run:  
cc <- conn_podr()  
get_table_names(con=cc)  
get_table_names(query_string = "select * from information_schema.tables")  
get_table_names(query_string = "select * from information_schema.tables", lib_sel = "readme")  
  
## End(Not run)
```

is_empty	<i>Check if a variable is na or null or space</i>
----------	---

Description

check if string or list is empty (na, null or blank spaces).

Usage

```
is_empty(x)
```

Arguments

x	a list or string
---	------------------

Value

true or false

Author(s)

Hanming Tu

Examples

```
is_empty(NULL);  
is_empty('');  
is_empty(NA);
```

read_podr	<i>Read a dataset from PODR</i>
-----------	---------------------------------

Description

This function will properly query PODR for datasets. There are three "libraries" available to read from: - CDISC Pilot ADaM data (PHUSE TDF Project 2019 cut) - CDISC Pilot SDTM data (PHUSE TDF Project 2019 cut) - Janssen Synthetic SDTM data

Usage

```
read_podr(  
  dataset = NULL,  
  libname = NULL,  
  tablename = NULL,  
  con = getOption("podr_connection"),  
  query_string = NULL,  
  limit = 100  
)
```

Arguments

dataset	Dataset name, specified as a character string
libname	Library name, one of <code>cdisc_pilot_adam</code> , <code>cdisc_pilot_sdtm</code> , or <code>janssen_synthetic</code>
tablename	Full table name, i.e., the combination of <code>libname</code> and <code>dataset</code>
con	The connection to PODR. Use <code>connect_podr</code> to establish a connection, or specify a variable containing the proper PostgreSQL connection into PODR yourself
query_string	provides full SQL statement
limit	to select specified number of records; defaults to 100

Details

Read data by specifying your desired dataset as a character string, then specify the library you'd like to read from. The libnames are specified as `-cdisc_pilot_adam` - `cdisc_pilot_sdtm` - `janssen_synthetic`

Value

The desired dataset as a `data.frame`

Author(s)

Hanming Tu

Examples

```
## Not run:
conn_podr()
read_podr('adae', libname='cdisc_pilot_adam')
read_podr('ae', libname='cdisc_pilot_sdtm')
read_podr('ae', libname='janssen_synthetic')

## End(Not run)
```

resolve

Resolve absolute path

Description

Resolve absolute directory

Usage

```
resolve(dir, relpath)
```


Arguments

dir	directory
relpath	relative path

Author(s)

Hanming Tu

Examples

```
resolve("/Users/htu/myRepo", "scripts")
# get "/Users/htu/myRepo/scripts"
```

start_app	<i>Start R Shiny app</i>
-----------	--------------------------

Description

start R Shiny apps included in this package.

Usage

```
start_app(
  app_name = "01_podr",
  n = 1,
  pkg = "podr",
  pt = NULL,
  lb = getOption("shiny.launch.browser", interactive()),
  ht = getOption("shiny.host", "127.0.0.1"),
  dm = "normal",
  msg_lvl = NULL,
  loc = "local"
)
```

Arguments

app_name	app or script name
n	app number
pkg	package name
pt	Port number
lb	define the browser- shiny.launch.browser
ht	define the host or ip address
dm	display modes are auto, normal or showcase
msg_lvl	message level
loc	location of the script: local github; default to 'local'

Author(s)

Hanming Tu

Examples

```
## Not run:  
library(genTS)  
start_app() # default to "02_display"  
start_app(1) # start "01_html"  
  
## End(Not run)
```

Index

`build_where_clause`, [2](#)

`conn_podr`, [3](#)

`echo_msg`, [4](#)

`get_table_defs`, [5](#)

`get_table_names`, [6](#)

`is_empty`, [7](#)

`read_podr`, [7](#)

`resolve`, [8](#)

`start_app`, [9](#)