

Package ‘shinyChatR’

February 28, 2023

Type Package

Title R Shiny Chat Module

Version 1.0.0

Description Provides an easy-to-use module for adding a chat to a Shiny app. Allows users to send messages and view messages from other users. Messages can be stored in a database or a .rds file.

License GPL (>= 3)

Encoding UTF-8

Imports DBI, purrr, R6, shiny

RoxygenNote 7.2.3

VignetteBuilder knitr

Suggests covr, knitr, rmarkdown, RSQLite, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/julianschmocker/shinyChatR>

NeedsCompilation no

Author Julian Schmocker [aut, cre, cph]

Maintainer Julian Schmocker <julian.schmocker@gmail.com>

Repository CRAN

Date/Publication 2023-02-28 14:30:02 UTC

R topics documented:

chat_server	2
chat_ui	3
DBConnection	3
RDSConnection	4
render_msg_divs	6

Index	7
--------------	----------

chat_server	<i>A chat module for Shiny apps - server</i>
-------------	--

Description

Creates the server logic for the chat module, which handles adding new messages to the database or RDS file, and retrieving messages to display

Usage

```
chat_server(  
  id,  
  chat_user,  
  db_connection = NULL,  
  db_table_name = NULL,  
  rds_path = NULL,  
  invalidateDSMillis = 1000  
)
```

Arguments

<code>id</code>	The id of the module.
<code>chat_user</code>	The user name that should be displayed next to the message.
<code>db_connection</code>	A database connection object, created using the DBI package. If provided, the chat messages will be stored in a database table.
<code>db_table_name</code>	he name of the database table to use for storing the chat messages. If <code>db_connection</code> is provided, this parameter is required.
<code>rds_path</code>	The path to an RDS file to use for storing the chat messages. If provided, the chat messages will be stored in an RDS file instead of a database.
<code>invalidateDSMillis</code>	The milliseconds to wait before the data source is read again. The default is 1 second.

Value

the reactive values `chat_rv` with all the chat information

chat_ui	<i>A chat module for Shiny apps - UI</i>
---------	--

Description

Creates the user interface for the chat module, which includes a chat message display area, a text input field for entering new messages, and a send button.

Usage

```
chat_ui(id, height = "300px", width = "500px")
```

Arguments

id	The id of the module
height	The height of the chat display area. Default is 300px.
width	The width of the chat display area. Default is 500px.

DBConnection	<i>DBConnection R6 Class</i>
--------------	------------------------------

Description

DBConnection R6 Class
DBConnection R6 Class

Details

An R6 class representing a connection to a database for the chat module.

Value

The full dataset
Save a message to data source

Public fields

connection A database connection object, created using a package such as RSQLite.
table The table that contains the chat information.
Initialize the R6 Object

Methods**Public methods:**

- [DBConnection\\$new\(\)](#)
- [DBConnection\\$get_data\(\)](#)
- [DBConnection\\$insert_message\(\)](#)
- [DBConnection\\$clone\(\)](#)

Method new():*Usage:*

DBConnection\$new(connection, table)

Arguments:

connection DB connection

table Table name

Method get_data(): Reads the full dataset*Usage:*

DBConnection\$get_data()

Method insert_message():*Usage:*

DBConnection\$insert_message(message, user, time)

Arguments:

message The message to be stores

user The user who entered the message

time The time when message was submitted

Method clone(): The objects of this class are cloneable with this method.*Usage:*

DBConnection\$clone(deep = FALSE)

Arguments:

deep Whether to make a deep clone.

RDSConnection*RDSConnection R6 Class*

Description

RDSConnection R6 Class

RDSConnection R6 Class

Details

An R6 class representing a connection to a rds file for the chat module.

Value

The full dataset

Save a message to data source

Public fields

rds_path The path to the rds file.

Initialize the R6 Object

Methods**Public methods:**

- [RDSCConnection\\$new\(\)](#)
- [RDSCConnection\\$get_data\(\)](#)
- [RDSCConnection\\$insert_message\(\)](#)
- [RDSCConnection\\$clone\(\)](#)

Method new():

Usage:

`RDSCConnection$new(rds_path)`

Arguments:

rds_path The path to the rds file.

Method get_data(): Reads the full dataset

Usage:

`RDSCConnection$get_data()`

Method insert_message():

Usage:

`RDSCConnection$insert_message(message, user, time)`

Arguments:

message The message to be stores

user The user who entered the message

time The time when message was submitted

Method clone(): The objects of this class are cloneable with this method.

Usage:

`RDSCConnection$clone(deep = FALSE)`

Arguments:

deep Whether to make a deep clone.

<code>render_msg_divs</code>	<i>Render the messages for the chat</i>
------------------------------	---

Description

Render the messages for the chat

Usage

```
render_msg_divs(texts, users, act_user)
```

Arguments

<code>texts</code>	a character vector with the texts
<code>users</code>	a character vector with the users
<code>act_user</code>	a character with the current user (that is using the app)

Value

The HTML code containing the chat messages

Index

chat_server, [2](#)

chat_ui, [3](#)

DBConnection, [3](#)

RDSConnection, [4](#)

render_msg_divs, [6](#)