Package 'RTD'

January 2, 2019

Title Simple TD API Client

Version 0.1.1

Maintainer Aki Ariga <chezou@gmail.com>

Description Upload R data.frame to Arm Treasure Data, see <https://www.treasuredata.com/>. You can execute database or table handling for resources on Arm Treasure Data.

SystemRequirements embulk, embulk-output-td

License Apache License 2.0 | file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Collate 'TdClient.R' 'database.R' 'table.R' 'td.R'

Imports readr (>= 1.2.1), httr(>= 1.4.0), openssl, jsonlite, methods, urltools

Suggests testthat (>= 2.0.1), mockery (>= 0.4.1.1), dplyr (>= 0.7.8), webmockr

NeedsCompilation no

Author Aki Ariga [aut, cre]

Repository CRAN

Date/Publication 2019-01-02 13:50:04 UTC

R topics documented:

`create_database` *Create a database*

Description

Create a database

Usage

```
create_database(conn, dbname, params)
```

Arguments

`conn`	`Td` client
`dbname`	Target data base name
`params`	Optional parameters

Value

Returns `TRUE` or `FALSE`, whether the execution succeeded or not.

Examples

```
## Not run:
con <- Td(apikey="xxxxx")
create_database(con, "newdb")

## End(Not run)
```

`create_table` *Create a table*

Description

Create a table

Usage

```
create_table(conn, dbname, table)
```

Arguments

`conn`	`Td` connection
`dbname`	Data base name
`table`	Table name

Value

Returns TRUE or FALSE, whether the execution succeeded or not.

Examples

```
## Not run:
conn <- Td(apikey="xxxx")
create_table(conn, "mydb", "new_table")

## End(Not run)
```

`delete_database` *Delete a database*

Description

Delete a database

Usage

```
delete_database(conn, dbname)
```

Arguments

conn	Td client
dbname	Target data base name

Value

Returns TRUE or FALSE, whether the execution succeeded or not.

Examples

```
## Not run:
conn <- Td(apikey="xxxx")
delete_database(conn, "mydb")

## End(Not run)
```

delete_table *Delete a table*

Description

Delete a table

Usage

```
delete_table(conn, dbname, table)
```

Arguments

conn	Td connection
dbname	Data base name
table	Table name

Value

Returns TRUE or FALSE, whether the execution succeeded or not.

Examples

```
## Not run:
conn <- Td(apikey="xxxxx")
delete_table(conn, "mydb", "iris")

## End(Not run)
```

exist_database *Check table existence*

Description

Check table existence

Usage

```
exist_database(conn, dbname)
```

Arguments

conn	Td client
dbname	Data base name

Value

Return TRUE or FALSE, existence

Examples

```
## Not run:
conn <- Td(apikey="xxxx")
exist_database(conn, "mydb")

## End(Not run)
```

`exist_table` *Check table existence*

Description

Check table existence

Usage

```
exist_table(conn, dbname, table)
```

Arguments

conn	Td connection
dbname	Data base name
table	Table name

Value

Returns TRUE or FALSE, existence.

Examples

```
## Not run:
conn <- Td(apikey="xxxxx")
exist_table(conn, "mydb", "iris")

## End(Not run)
```

list_databases *Show database list*

Description

Show database list

Usage

```
list_databases(conn)
```

Arguments

conn	Td connection

Value

Returns a data.frame of the database list

Examples

```
## Not run:
conn <- Td(apikey="xxxx")
list_databases(conn)

## End(Not run)
```

list_tables *Show list of tables*

Description

Show list of tables

Usage

```
list_tables(conn, dbname)
```

Arguments

conn	Td connection
dbname	Data base name. Optional, but highly recommended to prevent timeout.

Value

Returns a data.frame of a list of tables or FALSE if not exists.

Examples

```
## Not run:
conn <- Td(apikey="xxxxx")
list_tables(conn, "mydb")

## End(Not run)
```

Td *Connect to TD*

Description

Connect to TD

Usage

```
Td(endpoint, apikey, user_agent, headers, http_proxy = NULL)
```

Arguments

endpoint	Endpoint to TD API
apikey	API key for TD
user_agent	User-Agent as character. optional
headers	Default headres in a named character vector. optional
http_proxy	HTTP proxy setting. optional.

Examples

```
## Not run:
client <- Td(
  endpoint="api.treasuredata.com",
  apikey="xxxxxx",
  http_proxy="http://user:pass@proxy.domain.com:8080/")

## End(Not run)
```

td_upload *Upload data.frame to TD*

Description

Upload data.frame to TD

Usage

```
td_upload(conn, dbname, table, df, embulk_dir, overwrite = FALSE)
```

Arguments

conn	Td connection
dbname	Target destination database name.
table	Target table name.
df	Input data.frame.
embulk_dir	Path to embulk. [optional]
overwrite	Flag for overwriting the table if exists. It doesn't overwrite database.

Examples

```
## Not run:
td_upload_embulk("mydb", "iris", iris)

# With overwrite option
td_upload_embulk("mydb", "iris", iris, overwrite = TRUE)

# With overwrite option
td_upload_embulk("mydb", "iris", iris, "/path/to/embulk", overwrite = TRUE)

## End(Not run)
```

Index