Package 'tm1r'

November 14, 2020

Type Package

Title The Integration Between 'IBM COGNOS TM1' and R

Version 1.1.6

Author Muhammed Ali Onder

Maintainer Muhammed Ali Onder <muhammedalionder@gmail.com>

Description Useful functions to connect to 'TM1' <https://www.ibm.com/uk-en/products/planning-and-analytics> instance from R via REST API. With the functions in the package, data can be imported from 'TM1' via mdx view or native view, data can be sent to 'TM1', processes and chores can be executed, and cube and dimension metadata information can be taken.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports jsonlite, httr

Depends R (>= 3.0.0)

NeedsCompilation no

URL https://github.com/muhammedalionder/tm1r

BugReports https://github.com/muhammedalionder/tm1r/issues

Repository CRAN

Date/Publication 2020-11-14 18:20:02 UTC

R topics documented:

`tm1_api_request` *TM1 API Request*

Description

Makes a api request to tm1 server with url and body specified

Usage

```
tm1_api_request(tm1_connection, url, body ="", type = "GET")
```

Arguments

`tm1_connection`	tm1 connection object returned by the function tm1_connection
`url`	URL address for rest api request
`body`	body text of request
`type`	type of api request. Requests in httr package are supported like GET, POST, DELETE, PATCH

Examples

```
## Not run:
con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
url <- "https://localhost:8881/api/v1/Cubes('SalesCube')/Dimensions"
tm1_api_request(con_obj, url, type = "GET")


## End(Not run)
```

`tm1_connection` *TM1 Connection*

Description

Creates and returns a connection object to connect to TM1 via REST API.

Usage

```
tm1_connection(adminhost = "localhost", httpport = "",
username = "admin", password = "apple",
namespace="", ssl=TRUE, base_url="")
```

Arguments

`adminhost`	adminhost of tm1 model
`httpport`	httpport of tm1 model
`username`	username to connect to tm1 model
`password`	password of the username
`namespace`	ID of namespace should be specified if there is CAM security. Should be blank for native security
`ssl`	If UseSSL parameter is T in tm1s.cfg file, then TRUE. Else FALSE. Default is TRUE
`base_url`	when connecting to cloud, this option can be used instead of adminhost and httpport

Examples

```
## Not run:
tm1_connection("localhost", "8881", "admin", "apple")
tm1_connection(username="admin", password="apple",
base_url = "https://[Customer_Name].planning-analytics.ibmcloud.com/tm1/api/[Server Name]/")

## End(Not run)
```

tm1_create_element *TM1 Create New Element to a Dimension*

Description

Inserts a new element to the dimension

Usage

```
tm1_create_element(tm1_connection,
  dimension, element, parent="", weight=1)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
dimension	Name of dimension
element	Name of new element
parent	Name of parent of new element. Leave bland if there is no parent.
weight	Weight of the element as a component to the parent. Default is 1

Examples

```
## Not run:
tm1_create_element(tm1_connection("localhost", "8881", "admin", "apple"),
"month", "test", "Year")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_create_element(con_obj, "month", "test", "Year")

## End(Not run)
```

tm1_create_mdx *TM1 Generate mdx for a cube view*

Description

Returns mdx as a string to use in the function tm1_get_mdx_view

Usage

```
tm1_create_mdx(cube,
                              rowdim1, rowsub1, rowel1,
                              rowdim2, rowsub2, rowel2,
                              rowdim3, rowsub3, rowel3,
                              coldim1, colsub1, colel1,
                              coldim2, colsub2, colel2,
                              titledim1, titleel1,
                              titledim2, titleel2,
                              titledim3, titleel3,
                              titledim4, titleel4,
                              titledim5, titleel5,
                              titledim6, titleel6,
                              titledim7, titleel7,
                              titledim8, titleel8,
                              titledim9, titleel9,
                              titledim10, titleel10,
                              rowsuppress, colsuppress )
```

Arguments

cube	Name of the cube
rowdim1	Name of dimension in 1st row
rowsub1	Subset of dimension in 1st row
rowel1	Element of dimension in 1st row. If multiple, seperated by "|". This should be passed if subset is not provided
rowdim2	Name of dimension in 2nd row
rowsub2	Subset of dimension in 2nd row
rowel2	Element of dimension in 2nd row. If multiple, seperated by "|". This should be passed if subset is not provided
rowdim3	Name of dimension in 3rd row
rowsub3	Subset of dimension in 3rd row
rowel3	Element of dimension in 3rd row. If multiple, seperated by "|". This should be passed if subset is not provided
coldim1	Name of dimension in 1st col
colsub1	Subset of dimension in 1st col
colel1	Element of dimension in 1st col. If multiple, seperated by "|". This should be passed if subset is not provided
coldim2	Name of dimension in 2nd col
colsub2	Subset of dimension in 2nd col
colel2	Element of dimension in 2nd col. If multiple, seperated by "|". This should be passed if subset is not provided
titledim1	Name of dimension in title

titleel1	Element of dimension in corresponding titledim
titledim2	Name of dimension in title
titleel2	Element of dimension in corresponding titledim
titledim3	Name of dimension in title
titleel3	Element of dimension in corresponding titledim
titledim4	Name of dimension in title
titleel4	Element of dimension in corresponding titledim
titledim5	Name of dimension in title
titleel5	Element of dimension in corresponding titledim
titledim6	Name of dimension in title
titleel6	Element of dimension in corresponding titledim
titledim7	Name of dimension in title
titleel7	Element of dimension in corresponding titledim
titledim8	Name of dimension in title
titleel8	Element of dimension in corresponding titledim
titledim9	Name of dimension in title
titleel9	Element of dimension in corresponding titledim
titledim10	Name of dimension in title
titleel10	Element of dimension in corresponding titledim
rowsuppress	TRUE if zeroes are suppressed on rows
colsuppress	TRUE if zeroes are suppressed on columns

Examples

```
## Not run:

tm1_create_mdx( "SalesCube", rowdim1="account1", rowel1 = "Sales", coldim1="month", colel1="Jan",
titledim1 = "actvsbud", titleel1 = "Actual",
titledim2 = "region", titleel2 = "Argentina",
titledim3 = "model", titleel3 = "S Series 1.8 L Sedan",
rowsuppress=TRUE, colsuppress = FALSE)

## End(Not run)
```

tm1_create_subset *TM1 Create New Subset to a Dimension*

Description

Creates a new subset to the dimension

Usage

```
tm1_create_subset(tm1_connection,
  dimension, subset, element="", mdx="", overwrite=TRUE)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
dimension	Name of dimension
subset	Name of new subset
element	Name of elements seperated by | for static subset
mdx	mdx of subset for dynamic subset
overwrite	TRUE or FALSE. If TRUE, subset is overwritten

Examples

```
## Not run:
tm1_create_subset(tm1_connection("localhost", "8881", "admin", "apple"),
"month", "Q1Months", element = "Jan|Feb|Mar")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_create_element(con_obj, "month", "all", mdx = "[month].MEMBERS")

## End(Not run)
```

tm1_create_view *TM1 Create View from mdx*

Description

Creates cube view with mdx

Usage

```
tm1_create_view(tm1_connection, cube, view, mdx)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
cube	Name of cube
view	Name of view to be created
mdx	MDX of view as a string

Examples

```
## Not run:
mdx <- "SELECT
  NON EMPTY
   {[month].[Jan],[month].[Feb],[month].[Mar]}
  ON COLUMNS,
  NON EMPTY
     {[account1].[Price],[account1].[Units]}
  ON ROWS
 FROM [SalesCube]
 WHERE
  (
   [actvsbud].[actvsbud].[Actual],
   [region].[region].[Argentina],
   [model].[model].[S Series 1.8 L Sedan]
  )"
tm1_create_view(
  tm1_connection("localhost", "8881", "admin", "apple"),
  "SalesCube", "test", mdx)

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_create_view(con_obj, "SalesCube", "test", mdx)

## End(Not run)
```

---

tm1_delete_element *TM1 Delete Element or Component*

Description

Deletes element or component from dimensions

Usage

```
tm1_delete_element(tm1_connection,
  dimension, element, parent="")
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
dimension	Name of dimension
element	Name of element
parent	Name of parent of element. If parent is specified, component delete will be done. If parent is omitted, element will be deleted from dimension

Examples

```
## Not run:
tm1_delete_element(tm1_connection("localhost", "8881", "admin", "apple"),
"month", "test", "Year")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_delete_element(con_obj, "month", "test")

## End(Not run)
```

---

tm1_delete_subset *TM1 Delete Subset*

Description

Deletes subset from dimensions

Usage

```
tm1_delete_subset(tm1_connection,
  dimension, subset)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
dimension	Name of dimension
subset	Name of subset

Examples

```
## Not run:
tm1_delete_subset(tm1_connection("localhost", "8881", "admin", "apple"),
"month", "test")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_delete_subset(con_obj, "month", "test")

## End(Not run)
```

tm1_delete_view *TM1 Delete View*

Description

Deletes cube view

Usage

```
tm1_delete_view(tm1_connection, cube, view)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
cube	Name of cube
view	Name of view to be deleted

Examples

```
## Not run:

tm1_delete_view(
  tm1_connection("localhost", "8881", "admin", "apple"),
  "SalesCube", "test")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_delete_view(con_obj, "SalesCube", "test")

## End(Not run)
```

tm1_get_config *TM1 Get Configuration*

Description

Gets configuration of tm1 instance

Usage

```
tm1_get_config(tm1_connection)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection

Examples

```
## Not run:
tm1_get_config(tm1_connection("localhost", "8881", "admin", "apple"))

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_config(con_obj)

## End(Not run)
```

`tm1_get_cubes` *TM1 Get Cubes*

Description

Gets list of cubes

Usage

```
tm1_get_cubes(tm1_connection, ShowControlObjects = FALSE)
```

Arguments

`tm1_connection`	tm1 connection object returned by the function tm1_connection
`ShowControlObjects`	If TRUE, control cubes are also listed. Default is FALSE

Examples

```
## Not run:
tm1_get_cubes(tm1_connection("localhost", "8881", "admin", "apple"))

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_cubes(con_obj)

## End(Not run)
```

`tm1_get_cube_dimensions`
TM1 Get Dimensions of a Cube

Description

Gets dimensions of a cube

Usage

```
tm1_get_cube_dimensions(tm1_connection, cube)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
cube	Name of a cube as a string

Examples

```
## Not run:
tm1_get_cube_dimensions(
tm1_connection("localhost", "8881", "admin", "apple"),
"SalesCube")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_cube_dimensions(con_obj, "SalesCube")

## End(Not run)
```

---

tm1_get_data *TM1 Get Data from a Cube*

Description

Gets data from a cube, Supports up-to 10 dimension for now

Usage

```
tm1_get_data(tm1_connection, cube,
             element1="", element2="",
             element3="", element4="",
             element5="",element6="",
             element7="", element8="",
             element9="", element10="")
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
cube	Name of a cube as a string
element1	Element from 1st dimension of cube. Leave empty if there is no corresponding dimension
element2	Element from 2nd dimension of cube. Leave empty if there is no corresponding dimension
element3	Element from 3rd dimension of cube. Leave empty if there is no corresponding dimension
element4	Element from 4th dimension of cube. Leave empty if there is no corresponding dimension
element5	Element from 5th dimension of cube. Leave empty if there is no corresponding dimension

element6	Element from 6th dimension of cube. Leave empty if there is no corresponding dimension
element7	Element from 7th dimension of cube. Leave empty if there is no corresponding dimension
element8	Element from 8th dimension of cube. Leave empty if there is no corresponding dimension
element9	Element from 9th dimension of cube. Leave empty if there is no corresponding dimension
element10	Element from 10th dimension of cube. Leave empty if there is no corresponding dimension

Examples

```
## Not run:
tm1_get_data(
  tm1_connection("localhost", "8881", "admin", "apple"),
  "SalesCube", "Actual", "Argentina", "Total", "Sales", "Jan")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_data(con_obj, "SalesCube", "Actual", "Argentina", "Total", "Sales", "Jan")

## End(Not run)
```

---

tm1_get_dimensions *TM1 Get Dimensions*

Description

Gets list of dimensions

Usage

```
tm1_get_dimensions(tm1_connection, ShowControlObjects = FALSE)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
ShowControlObjects	If TRUE, control dimensions are also listed. Default is FALSE

Examples

```
## Not run:
tm1_get_dimensions(tm1_connection("localhost", "8881", "admin", "apple"))

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_dimensions(con_obj)

## End(Not run)
```

tm1_get_dimension_attributes

TM1 Get Attributes of a Dimension

Description

Gets attributes of a dimension

Usage

```
tm1_get_dimension_attributes(tm1_connection, dimension)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
dimension	Name of a dimension as a string

Examples

```
## Not run:
tm1_get_dimension_attributes(
tm1_connection("localhost", "8881", "admin", "apple"),
"region")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_dimension_attributes(con_obj, "region")

## End(Not run)
```

tm1_get_dimension_elements

TM1 Get Elements of a Dimension

Description

Gets elements of a dimension

Usage

```
tm1_get_dimension_elements(tm1_connection, dimension)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
dimension	Name of a dimension as a string

Examples

```
## Not run:
tm1_get_dimension_elements(
tm1_connection("localhost", "8881", "admin", "apple"),
"region")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_dimension_elements(con_obj, "region")

## End(Not run)
```

---

tm1_get_dimension_subsets

TM1 Get Subsets of a Dimension

Description

Gets subsets of a dimension

Usage

```
tm1_get_dimension_subsets(tm1_connection, dimension)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
dimension	Name of a dimension as a string

Examples

```
## Not run:
tm1_get_dimension_subsets(
tm1_connection("localhost", "8881", "admin", "apple"),
"region")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_dimension_subsets(con_obj, "region")

## End(Not run)
```

tm1_get_element *TM1 Get Element of a Dimension*

Description

Gets element detail of a dimension. Name, UniqueName, Type, Level, Index, and Components. element or index should be specified

Usage

```
tm1_get_element(tm1_connection, dimension, element="", index = 0)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
dimension	Name of a dimension as a string
element	Name of element as a string
index	Index of element as a numeric

Examples

```
## Not run:
tm1_get_element(
tm1_connection("localhost", "8881", "admin", "apple"),
"month", "Year")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_element(con_obj, "month", "", 7)

## End(Not run)
```

tm1_get_instances *TM1 Get Instances*

Description

Returns the list of tm1 instances in the specified adminhost

Usage

```
tm1_get_instances(adminhost = "localhost", port = "5898", ssl=TRUE)
```

Arguments

adminhost	adminhost of tm1 models
port	port of admin server
ssl	If TRUE it will be accesses through https

Examples

```
## Not run:
tm1_get_instances()

tm1_get_instances(adminhist = "localhost",
                  port = "5898", ssl = TRUE)


## End(Not run)
```

---

tm1_get_log *TM1 Get Logs of an instance*

Description

Gets server logs from a tm1 instance

Usage

```
tm1_get_log(tm1_connection, lognumber)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
lognumber	Number of how many lines of logs you want. Default is 5

Examples

```
## Not run:
tm1_get_log(tm1_connection("localhost", "8881", "admin", "apple"), 10)

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_log(con_obj)

## End(Not run)
```

---

tm1_get_mdx_view *TM1 Get Data from an MDX View*

Description

Gets mdx view data

Usage

```
tm1_get_mdx_view(tm1_connection, mdx, RowElementAsColumn = FALSE)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
mdx	MDX of view as a string
RowElementAsColumn	if False, row elements will be attached to rownames of data frame

Examples

```
## Not run:
mdx <- "SELECT
  NON EMPTY
   {[month].[Jan],[month].[Feb],[month].[Mar]}
  ON COLUMNS,
  NON EMPTY
     {[account1].[Price],[account1].[Units]}
  ON ROWS
 FROM [SalesCube]
 WHERE
  (
   [actvsbud].[actvsbud].[Actual],
   [region].[region].[Argentina],
   [model].[model].[S Series 1.8 L Sedan]
  )"
tm1_get_mdx_view(
  tm1_connection("localhost", "8881", "admin", "apple"),
  mdx, RowElementAsColumn=FALSE)

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_mdx_view(con_obj,mdx)

## End(Not run)
```

---

tm1_get_native_view *TM1 Get Data from a Native View*

Description

Gets native view data

Usage

```
tm1_get_native_view(tm1_connection, cube, view, RowElementAsColumn= FALSE)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
cube	Name of the cube
view	Name of the view
RowElementAsColumn	if False, row elements will be attached to rownames of data frame

Examples

```
## Not run:

tm1_get_native_view(
tm1_connection("localhost", "8881", "admin", "apple"),
  "SalesCube", "Default", RowElementAsColumn=FALSE)

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_native_view(con_obj, "SalesCube", "Default")

## End(Not run)
```

`tm1_get_subset_elements`

TM1 Get Elements of a subset

Description

Gets elements of a subset

Usage

```
tm1_get_subset_elements(tm1_connection, dimension, subset)
```

Arguments

`tm1_connection`	tm1 connection object returned by the function tm1_connection
`dimension`	Name of a dimension as a string
`subset`	Name of a subset as a string

Examples

```
## Not run:
tm1_get_subset_elements(
tm1_connection("localhost", "8881", "admin", "apple"),
"region", "default")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_get_dimension_elements(con_obj, "region", "default")

## End(Not run)
```

tm1_logout *TM1 Log Out*

Description

Logs out

Usage

```
tm1_logout(tm1_connection)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection

Examples

```
## Not run:
tm1_logout(tm1_connection("localhost", "8881", "admin", "apple"))

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_logout(con_obj)

## End(Not run)
```

tm1_run_chore *TM1 Run a Chore*

Description

Runs a chore

Usage

```
tm1_run_chore(tm1_connection, chore)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
chore	Name of a chore as a string

Examples

```
## Not run:
tm1_run_chore(tm1_connection("localhost", "8881", "admin", "apple"), "test")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_run_chore(con_obj, "test")

## End(Not run)
```

`tm1_run_process` *TM1 Run a Process*

Description

Runs a process

Usage

```
tm1_run_process(tm1_connection,
  process,
  par1name, par1value,
  par2name, par2value,
  par3name, par3value)
```

Arguments

`tm1_connection`	tm1 connection object returned by the function tm1_connection
`process`	Name of a process as a string
`par1name`	Name of a parameter
`par1value`	Value of a parameter
`par2name`	Name of a parameter
`par2value`	Value of a parameter
`par3name`	Name of a parameter
`par3value`	Value of a parameter

Examples

```
## Not run:
tm1_run_process(tm1_connection("localhost", "8881", "admin", "apple"), "test")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_run_process(con_obj, "test")

## End(Not run)
```

tm1_send_data *TM1 Send Data to a Cube*

Description

Send data to a cube, Supports up-to 10 dimension for now

Usage

```
tm1_send_data(tm1_connection, value, cube,
                               element1, element2, element3, element4, element5,
                              element6, element7, element8, element9, element10,
                                increment)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
value	data value you want to send to cube
cube	Name of a cube as a string
element1	Element from 1st dimension of cube. Leave empty if there is no dimension
element2	Element from 2nd dimension of cube. Leave empty if there is no dimension
element3	Element from 3rd dimension of cube. Leave empty if there is no dimension
element4	Element from 4th dimension of cube. Leave empty if there is no dimension
element5	Element from 5th dimension of cube. Leave empty if there is no dimension
element6	Element from 6th dimension of cube. Leave empty if there is no dimension
element7	Element from 7th dimension of cube. Leave empty if there is no dimension
element8	Element from 8th dimension of cube. Leave empty if there is no dimension
element9	Element from 9th dimension of cube. Leave empty if there is no dimension
element10	Element from 10th dimension of cube. Leave empty if there is no dimension
increment	If TRUE, it will increment cube data by Value. If False, it will replace. This parameter is ignored in sending string values.

Examples

```
## Not run:
tm1_send_data(
  tm1_connection("localhost", "8881", "admin", "apple"),
  10,
  "SalesCube",
  "Actual", "Argentina", "S Series 1.8 L Sedan", "Units", "Jan")

con_obj <- tm1_connection("localhost", "8881", "admin", "apple")
tm1_send_data(con_obj,
  10,
```

```
  "SalesCube",
  "Actual", "Argentina", "S Series 1.8 L Sedan", "Units", "Jan",
  increment=TRUE)

## End(Not run)
```

---

tm1_send_dataset *TM1 Send Data Set to a Cube*

Description

Send data to a cube, Supports up-to 10 dimension for now

Usage

```
tm1_send_dataset(tm1_connection, valueset, cube, rowdim, coldim,
                                 rowdim2, rowdim3, rowdim4, rowdim5,
                                 titledim1, titleel1, titledim2, titleel2,
                                 titledim3, titleel3, titledim4, titleel4,
                                 titledim5, titleel5, titledim6, titleel6,
                                 titledim7, titleel7, titledim8, titleel8)
```

Arguments

tm1_connection	tm1 connection object returned by the function tm1_connection
valueset	data frame or matrix object holding values you want to send to cube
cube	Name of a cube as a string
rowdim	Corresponding dimension of the elements on row
coldim	Corresponding dimension of the elements on column
rowdim2	Corresponding dimension of the elements on row2
rowdim3	Corresponding dimension of the elements on row3
rowdim4	Corresponding dimension of the elements on row4
rowdim5	Corresponding dimension of the elements on row5
titledim1	Name of dimension in title
titleel1	Element of dimension in corresponding titledim
titledim2	Name of dimension in title
titleel2	Element of dimension in corresponding titledim
titledim3	Name of dimension in title
titleel3	Element of dimension in corresponding titledim
titledim4	Name of dimension in title
titleel4	Element of dimension in corresponding titledim
titledim5	Name of dimension in title

titleel5	Element of dimension in corresponding titledim
titledim6	Name of dimension in title
titleel6	Element of dimension in corresponding titledim
titledim7	Name of dimension in title
titleel7	Element of dimension in corresponding titledim
titledim8	Name of dimension in title
titleel8	Element of dimension in corresponding titledim

Examples

```
## Not run:

sdata <- tm1_connection("localhost", "8881", "admin", "apple")

#valueset
#    Argentina Brazil
#Jan         1      2
#Feb         3      4

tm1_send_dataset(
  sdata,
  valueset = valueset, cube = "SalesCube",
  rowdim = "month", coldim = "region",
  titledim1 = "actvsbud", titleel1 = "Actual",
  titledim2 = "model", titleel2 = "L Series 1.6 L Convertible",
  titledim3 = "account1", titleel3 = "Units")



## End(Not run)
```

Index