Package ‘mapxhelper’

October 6, 2015

Title map-x helper functions.

Version 0.0.1

Date 2015-10-05

Description map-x helper functions

License GPL-3 | file LICENSE

URL https://github.com/fxi/map-x-shiny

BugReports https://github.com/fxi/map-x-shiny/issues

Imports leaflet,
RPostgreSQL

R topics documented:

`addPaletteFun` *Add palette to map-x style list*

Description

Update a style list with a palette, using the defined scale type : continuous or discrete.

Usage

```
addPaletteFun(sty, pal)
```

Arguments

`sty`	map-x style
`pal`	name of palette to use

`dbGetFilterCenter` *Get query extent, based on a pattern matching (character)*

Description

Search for a value in a column (character data type) and return the extent if something is found.

Usage

```
dbGetFilterCenter(dbInfo = NULL, table = NULL, column = NULL,
  value = NULL, geomColumn = "geom", operator = "=")
```

Arguments

`dbInfo`	Named list with dbName,host,port,user and password
`table`	Table/layer from which extract extent
`geomColumn`	set geometry column

Value

extent

`dbGetLayerCentroid` *Get layer center*

Description

Compute the union of all geometry in a given layer and return the coordinate of the centroid.

Usage

```
dbGetLayerCentroid(dbInfo = NULL, table = NULL, geomColumn = "geom")
```

Arguments

`dbInfo`	Named list with dbName,host,port,user and password
`table`	Table/layer from which extract extent
`geomColumn`	set geometry column

Value

extent

`dbWriteSpatial` *Write spatial data frame to postgis*

Description

Convert spatial data.frame to postgis table. Taken from https://philipphunziker.wordpress.com/2014/07/20/transferring-vector-data-between-postgis-and-r/

Usage

```
dbWriteSpatial(con, spatial.df, schemaname = "public", tablename,
  overwrite = FALSE, keyCol = "gid", srid = 4326, geomCol = "geom")
```

Arguments

`con`	PostgreSQL connection
`spatial.df`	Spatial data frame object
`schemaname`	Target schema table
`tablename`	Target table name
`overwrite`	Overwrite if exists
`keyCol`	Set new primary key
`srid`	Set the epsg code / SRID
`geomCol`	Set the name of the geometry column

`hot.to.df` *hot.to.df*

Description

Converts the table data passed from the client-side into a data.frame

Usage

```
hot.to.df(b)
```

Arguments

`b`	The input$hotable_id value.

`hotable` *hotable*

Description

Creates a hotable (handsontable)

Usage

```
hotable(id, width = "100%", height = "100%")
```

Arguments

`id`	The id used to refer to the table input$id or output$id

`loadUi` *Load external ui file value in shiny app*

Description

Shortcut to load external shiny ui file

Usage

```
loadUi(path)
```

Arguments

`path` Path to the file

`mapxhelper` *Map-x helper functions*

Description

All the R fonctions defined in map-x-shiny are container in this package

`mxAccordionGroup` *Create a bootstrap accordion*

Description

Create a bootstrap accordion element, based on a named list.

Usage

```
mxAccordionGroup(id, style = NULL, show = NULL, itemList)
```

Arguments

`id` Accordion group ID

`style` Additional style.

`show` Vector of item number. Collapse all item except those in this list. E.g. c(1,5) will open items 1 and 5 by default.

`itemList` Nested named list of items, containing title and content items. E.g. list("foo"=list("title"="foo","conte

Examples

```
amAccordionGroup(id='superTest',
 itemList=list(
   'a'=list('title'='superTitle',content='acontent'),
   'b'=list('title'='bTitle',content='bContent'))
 )
```

mxActionButtonState *Toggle disabling of given button, based on its id.*

Description

Action or other button can be disabled using the attribute "disabled". This function can update a button state using this method.

Usage

```
mxActionButtonState(id, session = shiny:::getDefaultReactiveDomain(),
  disable = TRUE)
```

Arguments

`id`	Id of the button.
`session`	Shiny session object.
`disable`	State of the button

mxAllow *Control ui access*

Description

Use mxConfig$roleVal list to check if the curent user's role name can access to the given numeric role.

Usage

```
mxAllow(logged, roleName, roleLowerLimit)
```

Arguments

`logged`	Boolean. Is the user logged in ?
`roleName`	Character. Role in numeric format
`roleLowerLimit`	Numeric. Minumum role requirement

mxCatch *Catch errors*

Description

Catch errors and return alert panel in an existing div id.

Usage

```
mxCatch(title, expression, session = shiny:::getDefaultReactiveDomain(),
  debug = TRUE, panelId = "panelAlert")
```

Arguments

title	Title of the alert
session	Shiny session object
debug	Boolean. Return also message as alert.
panelId	Id of the output element

mxCheckboxIcon *Set a checkbox button with custom icon.*

Description

Create a checkbox input with a select icon.

Usage

```
mxCheckboxIcon(id, icon)
```

Arguments

id	Id of the element
icon	Name of the fontawesome icon. E.g. cog, times, wrench

mxCreateSecret *Create random secret*

Description

Get a random string of letters and hash it.

Usage

```
mxCreateSecret(n = 20)
```

Arguments

n	Number of input letter for the MD5 hash

`mxDbGetQuery` *Get query result from postgresql*

Description

Shortcut to create a connection, get the result of a query and close the connection, using a dbInfo list.

Usage

```
mxDbGetQuery(dbInfo, query)
```

Arguments

`dbInfo`	Named list with dbName,host,port,user and password
`SQL`	query

`mxDbListTable` *List existing table from postgresql*

Description

Shortcut to create a connection, get the list of table and close the connection, using a dbInfo list.

Usage

```
mxDbListTable(dbInfo)
```

Arguments

`dbInfo`	Named list with dbName,host,port,user and password

`mxDebugMsg` *Print debug message*

Description

Print a defaut debug message with date as prefix. NOTE: this function should take a global parameter "debug" and a log file.

Usage

```
mxDebugMsg(m = "")
```

Arguments

`m`	Message to be printed

`mxFileInput` *Custom file input*

Description

Default shiny fileInput has no option for customisation. This function allows to fully customize file input using the label tag.

Usage

```
mxFileInput(inputId, label, fileAccept = NULL, multiple = FALSE)
```

Arguments

`inputId`	id of the file input
`label`	Label for the input
`fileAccept`	List of accepted file type. Could be extension.
`multiple`	Boolean. Allow multiple file to be choosen. Doesn't work on all browser.

`mxGetViewsTable` *Retrieve map views table*

Description

Get a list of available map-x views in given table, e.g. mx_views

Usage

```
mxGetViewsTable(dbInfo = NULL, table = "mx_views", validated = TRUE,
  archived = FALSE, country = "AFG")
```

Arguments

`dbInfo`	Named list with dbName,host,port, user and password
`table`	Table name containing views info
`validated`	Boolean filter validated dataset. Default = TRUE
`archived`	Boolean filter to get archived data. Default =FALSE
`country`	ISO 3 code to filter country.

mxPanel *Create a modal panel*

Description

Create a modal panel with some options as custom button, close button, html content.

Usage

```
mxPanel(id = "default", title = NULL, subtitle = NULL, html = NULL,
  listActionButton = NULL, background = TRUE, defaultButtonText = "OK",
  style = NULL, class = NULL, hideCloseButton = FALSE, draggable = TRUE)
```

Arguments

`id`	Panel id
`title`	Panel title
`subtitle`	Panel subtitle
`html`	HTML content of the panel, main text
`listActionButton`	If FALSE, hide buttons. If NULL, display default close panel button, with text given in defaultButtonText. If list of buttons, list of button.
`defaultButtonText`	Text of the default button if listActionButton is NULL and not FALSE
`style`	Additional CSS style for the panel
`class`	Additional class for the panel
`hideCloseButton`	Boolean. Hide the close panel button
`draggable`	Boolean. Set the panel as draggable

mxPanelAlert *Alert panel*

Description

Create an alert panel. This panel could be send to an output object from a reactive context.

Usage

```
mxPanelAlert(title = c("error", "warning", "message"), subtitle = NULL,
  message = NULL, listActionButton = NULL)
```

Arguments

`title`	Title of the alert. Should be "error", "warning" or "message"
`subtitle`	Subtitle of the alert
`message`	html or text message for the alert
`listActionButtons`	List of action button for the panel

`mxSelectInput` *Custom select input*

Description

Custom select input without label.

Usage

```
mxSelectInput(inputId, choices = NULL, selected = NULL)
```

Arguments

`inputId`	Element id
`choices`	List of options
`select`	Value selected by default

`mxSetCookie` *Save named list of value into cookie*

Description

Note : don't use this for storing sensitive data, unless you have a trusted network.

Usage

```
mxSetCookie(session = getDefaultReactiveDomain(), cookie = NULL,
  nDaysExpires = NULL, deleteAll = FALSE)
```

Arguments

`session`	Shiny session object. By default: default reactive domain.
`cookie`	Named list holding paired cookie value. e.g. (list(whoAteTheCat="Alf"))
`nDaysExpires`	Integer of days for the cookie expiration

mxSetMapPanelMode *Set map panel mode.*

Description

Map-x panel use multiple panel mode : config,creator,explorer,toolbox. This function set and save the panel mode.

Usage

```
mxSetMapPanelMode(session = shiny::getDefaultReactiveDomain(),
  mode = c("mapViewsConfig", "mapViewsCreator", "mapStoryCreator",
  "mapViewsExplorer", "mapViewsToolbox"), title = NULL)
```

Arguments

session	Shiny session
mode	Map panel mode. In mapViewCreator, mapStoryCreator, mapExplorer
title	Optionnal title to be returned.

Value

title string

mxSetStyle *Apply map-x style to existing vector tiles*

Description

When leafletvt handle a vector tile source, a lealflet object is stored in leafletvtId, but no style is applied. Default is transparent. We add a style function after that the layer is fully loaded using this function. The style function is also stored alongside the leaflet object in leafletId under the name "vtStyle".

Usage

```
mxSetStyle(session = shiny:::getDefaultReactiveDomain(), style)
```

Arguments

session	Shiny session object
style	map-x style

`mxSliderOpacity` *Set ioRange slider for opacity*

Description

Set ioRange slider for opacity

Usage

```
mxSliderOpacity(id, opacity)
```

Arguments

`id`	Id of the slider
`opacity`	Default opacity

`mxTimeSlider` *Set ioRange slider for time slider*

Description

Set ioRange slider for time slider

Usage

```
mxTimeSlider(id, min, max)
```

Arguments

`id`	Id of the slider
`min`	Minimum js unix date in milisecond
`max`	Maxmimum js unix date in milisecond

`mxUiAccess` *Control ui access*

Description

UI manager based on login info

Usage

```
mxUiAccess(logged, roleNum, roleLowerLimit, uiDefault, uiRestricted)
```

Arguments

`logged`	Boolean. Is the user logged in ?
`roleNum`	Numeric. Role in numeric format
`roleLowerLimit`	Numeric. Minumum role requirement
`uiDefault`	TagList. Default ui.
`uiRestricted`	TagList. Restricted ui.

`mxUiEnable` *Control visbility of elements*

Description

Display or hide element by id, without removing element AND without having element's space empty in UI. This function add or remove mxHide class to the element.

Usage

```
mxUiEnable(session = shiny:::getDefaultReactiveDomain(), id = NULL,
  enable = TRUE)
```

Arguments

`session`	Shiny session
`id`	Id of element to enable/disable
`enable`	Boolean. Enable or not.

`mxUpdateChartRadar` *Create a chartRadar in a canvas element.*

Description

Search the dom for an id a get drawing context, create a new chart object and config it with data.

Usage

```
mxUpdateChartRadar(session = shiny::getDefaultReactiveDomain(), main,
  compMain, id, idLegend, labels, values, compValues)
```

Arguments

`session`	Shiny reactive session
`main`	Main label
`compMain`	Comparative value label
`id`	Id of the canvas
`idLegend`	Id of the legend
`labels`	Labels for value and comparative values
`compValues`	Comparative values
`value`	Values

mxUpdatePanel *Update existing panel*

Description

Use output object to output the panel in a known id. E.g. for updating uiOutput("panelTest"), use mxUpdatePanel with panelId "panelTest"

Usage

```
mxUpdatePanel(panelId = NULL, session = shiny:::getDefaultReactiveDomain(),
  ...)
```

Arguments

panelId	Id of the existing panel
session	Shiny reactive object of the session
...	Other mxPanel options

mxUpdateText *Update text by id*

Description

Search for given id and update content.

Usage

```
mxUpdateText(session = shiny:::getDefaultReactiveDomain(), id, text)
```

Arguments

session	Shiny session
id	Id of the element
text	New text

pwdInput *Password input*

Description

Create a password input.

Usage

```
pwdInput(inputId, label)
```

Arguments

`inputId`	Input id
`label`	Label to display

randomName *Random name generator*

Description

Create a random name with optional prefix and suffix.

Usage

```
randomName(prefix = NULL, suffix = NULL, n = 20, sep = "_")
```

Arguments

`prefix`	Prefix. Default = NULL
`suffix`	Suffix. Default = NULL
`n`	Number of character to include in the random string

Value

Random string of letters, with prefix and suffix

remoteCmd *Send command on remote server through ssh*

Description

Allow sending command on a remote server, e.g. Vagrant machine, using ssh.

Usage

```
remoteCmd(host = NULL, user = NULL, port = NULL, cmd = NULL,
  vagrant = TRUE, sshConfig = "settings/sshConfig")
```

Arguments

`host`	Host
`user`	User
`port`	Port
`cmd`	Command to send
`vagrant`	Boolean. If TRUE, use ssh config file. E.g. vagrant ssh-config > sshConfig

renderHotable *renderHotable*

Description

Renders the hotable.

Usage

```
renderHotable(expr, env = parent.frame(), quoted = FALSE, options = NULL,
  readOnly = NULL, fixedCols = 1, stretched = c("all", "last", "none"))
```

Arguments

`expr`	The computation that leads to an output
`env`	The R environment in which to create the dataset
`quoted`	Pass to the exprToFunction
`options`	Pass to the exprToFunction
`readOnly`	A vector of TRUE/FALSE values to indicate which of the columns should be readonly. If numeric vector, select col number to set as readOnly.
`fixedCols`	A vector of integer of columns number to fix.

`setLayerOpacity` *Set opacity.*

Description

Set given layer opacity.

Usage

```
setLayerOpacity(session = shiny:::getDefaultReactiveDomain(),
  layer = "leafletvtId", group = NULL, opacity = 1)
```

Arguments

`session`	Shiny session
`layer`	Leaflet.MapboxVectorTile layer group object name
`group`	Group
`opacity`	Opacits

`setLayerZIndex` *Set zIndex.*

Description

Set zIndex on a leafletvt layer. NOTE: leaflet seems to reset this after layer changes.

Usage

```
setLayerZIndex(session = getDefaultReactiveDomain(), layer = "leafletvtId",
  group = NULL, zIndex = 15)
```

Arguments

`session`	Shiny session
`layer`	Leaflet.MapboxVectorTile layer group object name
`zIndex`	zIndex of the group

subPunct *Substitute ponctiation and non-ascii character*

Description

Take a string and convert to ascii string with optional transliteration ponctuation convertion.

Usage

```
subPunct(str, sep = "_", rmTrailingSep = T, rmLeadingSep = T,
  rmDuplicateSep = T, useTransliteration = T)
```

Arguments

str	String to evaluate
sep	Replace separator
rmTrailingSep	Logical argument : no trailing separator returned
rmLeadingSep	Logical argument : no leading separator returned
rmDuplicateSep	Logical argument : no consecutive separator returned

toggleClass *Toggle html element by class.*

Description

Toggle html hide parameter by class.

Usage

```
toggleClass(session = shiny:::getDefaultReactiveDomain(), class = "")
```

Arguments

session	Shiny session
class	CSS class to hide

`usrInput` *User name input*

Description

Create a username input

Usage

```
usrInput(inputId, label)
```

Arguments

`inputId`	Input id
`label`	Label to display

Index