

Package ‘warp10r’

December 20, 2017

Title R Client for executing Warpscript on a Warp10 instance

Version 0.4.0

Description This R client enables to make basic ingress and egress routine with Warp 10.

Depends R (>= 3.3.2)

License Apache 2.0

Encoding UTF-8

RoxygenNote 6.0.1.9000

R topics documented:

extractGTS	1
permalink	2
postWarpscript	2
pushWarp10	3
toGtsInputFormat	4
Index	5

extractGTS	<i>Extract Geo Time Series</i>
------------	--------------------------------

Description

Extract all GTS from a JSON response and return them merged as a data.table. If too slow, consider using postWarpscript with outputType="data.table" instead.

Usage

```
extractGTS(response, withLabels = FALSE)
```

Arguments

response	response body of a post request (a character vector or a json)
withLabels	if TRUE, column names also include Labels. Default to FALSE

Value

data.table

permalink	<i>Generate a Permalink</i>
-----------	-----------------------------

Description

Generate a permalink to a quantum instance.

Usage

```
permalink(warpscript, plot = FALSE,
  endpoint = "http://localhost:8080/api/v0/exec",
  quantum = "http://localhost:8090")
```

Arguments

warpscript	code or file name ending with .mc2
plot	if TRUE, the generated link points to a plot. Default to FALSE
endpoint	egress endpoint. Default to "http://localhost:8080/api/v0/exec"
quantum	address of quantum instance. Default to "http://localhost:8090"

Value

url

postWarpscript	<i>Post Warpscript Code</i>
----------------	-----------------------------

Description

Post warpscript code to a Warp 10 instance and retrieve response as a specified type

Usage

```
postWarpscript(warpscript, outputType = "json",
  endpoint = "http://localhost:8080/api/v0/exec", withLabels = FALSE)
```

Arguments

warpscript	code or file name ending with .mc2
outputType	one of: <ul style="list-style-type: none">• "json" - returns a json (default)• "raw" - returns a character vector• "pretty" - returns a json that is pretty indented• "list" - returns a list• "data.table" - returns a single data.table with all data. Requires the first level of the stack to be a list of GTS• "gts.list" - returns a named list of one data.table per GTS. Requires the first level of the stack to be a list of GTS. The keys of the returned list are the GTS classnames. NaN values are not supported• "dataFrame" - deprecated in favors of data.table
endpoint	egress endpoint. Default to "http://localhost:8080/api/v0/exec"
withLabels	if TRUE, labels are appended to classnames when outputType is "data.table" or "gts.list"

Value

character vector or json or named list or data.table

pushWarp10	<i>Push data points</i>
------------	-------------------------

Description

Push data points to an ingress instance.

Usage

```
pushWarp10(data, token, endpoint = "http://localhost:8080/api/v0/update")
```

Arguments

data	data points in GTS input format as a character vector or a filename ending with .data or .gz
token	write token
endpoint	ingress endpoint. Default to "http://localhost:8080/api/v0/update"

toGtsInputFormat	<i>Convert a data frame</i>
------------------	-----------------------------

Description

Convert a data frame into GTS input format.

Usage

```
toGtsInputFormat(dataFrame)
```

Arguments

dataFrame	first column must be named "timestamp". Column names corresponding to GTS values must be in the <selector> form: "classname{label1=value1,label2=value2,...}" (or "classname"). Optional geo columns must be named "<selector>.lat", "<selector>.lon" or "<selector>.elev"
-----------	--

Value

data points in GTS input format as a character vector

Index

`extractGTS`, [1](#)

`permalink`, [2](#)

`postWarpscript`, [2](#)

`pushWarp10`, [3](#)

`toGtsInputFormat`, [4](#)