Package: climateR.catalogs (via r-universe)

December 4, 2024

Version 0.1.0

License MIT

Description A collection of tools for maniputatling hydrologic and hydraulic networks

Imports arrow, dplyr, here, httr, glue, targets, tarchetypes, tidyr, terra, sf, rvest, jsonlite, logger, lubridate, ncmeta, pacman, RNetCDF, data.table, RCurl, readr, stringr, AOI, climateR (>= 0.3.1)

Remotes mikejohnson51/AOI, mikejohnson51/climateR

Roxygen list(markdown = TRUE, r6 = TRUE)

RoxygenNote 7.2.3

Encoding UTF-8

Collate 'class-data_source.R' 'meta-data_source.R' 'meta-targets.R' 'utils.R'

Config/pak/sysreqs cmake libgdal-dev gdal-bin libgeos-dev git libglpk-dev make libicu-dev libpng-dev libxml2-dev libnetcdf-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev zlib1g-dev

Repository https://mikejohnson51.r-universe.dev

RemoteUrl https://github.com/mikejohnson51/climateR-catalogs

RemoteRef HEAD

RemoteSha d9331a007f5ac7673fc1a8f62243831dd24554f8

Contents

data_source *climateR Catalog Data Source*

Description

This is a generic base class for describing catalog data sources. Each data source must provide a `pull` and `tidy` method.

Value

`(raw())`
The data member of this data source in Arrow IPC Stream format

`(character(1))`
The given path, invisibly.

Public fields

`.id (character(1))`

`.data (arrow::Table)`

`.pull (function)`

`.tidy (function)`

`.finished (logical(1))`
`TRUE` if `$tidy()` has been called successfully, otherwise `FALSE`.

Active bindings

`id (character(1))`
Identifier of this data source.

`result (arrow::Table)`
Result of this data source after `$tidy()` is called. (**Read-only**)

`.id (character(1))`

`.data (arrow::Table)`

`.pull (function)`

`.tidy (function)`

`.finished (logical(1))`
`TRUE` if `$tidy()` has been called successfully, otherwise `FALSE`.

Methods

Public methods:

- `data_source$new()`
- `data_source$print()`
- `data_source$pull()`

- data_source$tidy()
- data_source$to_ipc_stream()
- data_source$from_ipc_stream()
- data_source$to_ipc_file()
- data_source$from_ipc_file()
- data_source$clone()

Method `new()`**:** Create a new catalog data source.

Usage:

```
data_source$new(id, pull, tidy)
```

Arguments:

`id` (`character(1)`)
: Identifier for this data source.

`pull` (`function`)
: Pull method for this class.

  The pull function may require any amount of parameters.

  The pull function must return one of: `arrow::Table`, `data.table::data.table`, or a `data.frame`.

`tidy` (`function`)
: Tidy method for this class. See the `tidy` method for details.

  The tidy function must require at least 1 argument as its first argument that takes in, and also **returns**, one of: `arrow::Table`, `data.table::data.table`, or a `data.frame`

Method `print()`**:**

Usage:

```
data_source$print(...)
```

Method `pull()`**:** Pull a catalog data source from its endpoint.
This method is user-defined at object creation.

Usage:

```
data_source$pull(...)
```

Arguments:

`...` (`any`)
: User-defined parameters that may be used.

Method `tidy()`**:** Tidy a raw catalog data source into the catalog schema.

Usage:

```
data_source$tidy(...)
```

Arguments:

`...` (`any`)
: User-defined parameters that may be used.

Method `to_ipc_stream()`**:** Marshal this data source to Arrow IPC Stream format

Usage:

```
data_source$to_ipc_stream()
```

Method `from_ipc_stream()`**:** Unmarshals an Arrow IPC Stream to a data source

Usage:

```
data_source$from_ipc_stream(stream)
```

Arguments:

`stream (raw())`
: The given Arrow IPC Stream

Method `to_ipc_file()`**:** Output this data source to Arrow IPC File format

Usage:

```
data_source$to_ipc_file(path)
```

Arguments:

`path (character(1))`
: Path to file to write to. Should have extension '.arrow'.

Method `from_ipc_file()`**:** Read a data source from Arrow IPC File format

Usage:

```
data_source$from_ipc_file(path)
```

Arguments:

`path (character(1))`
: Path to Arrow IPC file.

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

Usage:

```
data_source$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Note

If all parameters are missing, then an empty data_source is created. This is only useful for reading from IPC.

`new_data_source` *Create a new climateR.catalogs data source plugin*

Description

Create a new climateR.catalogs data source plugin

Usage

```
new_data_source(name, dir)
```

Arguments

`name`	Name of the data source
`dir`	Directory to output

Index