Package ‘WISKIr’

April 6, 2017

Title Acquires data from a WISKI web server

Version 2.1.4

Date 2016-04-06

Author Kevin Shook, Centre for Hydrology, University of Saskatchewan

Maintainer Kevin Shook <kevin.shook@usask.ca>

Description Functions to acquire a) the set of time series for a station, b) the metadata for a specific times series, c) the values for a specified time series over a specified range of dates.

Depends R (>= 3.3)

Imports stringr (>= 1.0)

License GPL-3

URL www.usask.ca/hydrology

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

WISKIr-package 2
findWISKIgroups 2
findWISKIstations 3
findWISKItimeseries 4
getWISKImetadata 5
getWISKIvalues 6
WISKItoObs 7

Index **8**

WISKIr-package *Contains functions to extract data from a Kisters WISKI web server.*

Description

Using the functions, data values can be downloaded directly from a WISKI web server. This allows you to automate data aquisition in your scripts. It is a good idea to use the function `findWISKItimeseries` first. It allows you to search by the name of a station, and gives you the `id` numbers for for all of its time series. When you have the `id` number of a time series, you can obtain its metadata with the command `getWISKImetadata`. The actual values are downloaded using the command `getWISKIvalues`. The WISKI values can be converted to a **CRHMr** obs dataframe using the command `WISKItoObs`.

References

To cite **WISKIr** in publications, use the command `citation('WISKIr')` to get the current version of the citation.

findWISKIgroups *Finds all groups on the Wiski server*

Description

Finds all groups on the Wiski server

Usage

```
findWISKIgroups(site.url = "http://giws.usask.ca:8080/")
```

Arguments

site.url	Optional. A character string containing the url of the WISKI web server. Defaults to the Global Institute for Water Security (GIWS) server: `'http://giws.usask.ca:8080/'`. As this package is intended for use by the GIWS hydrological community, it is usually unnecessary to specify the web server.

Value

If unsuccessful, returns `FALSE`. If successful, returns a dataframe containing

group_id	ID number of the group
group_name	WISKI name of the group
gorup_type	Type of the group

Author(s)

Kevin Shook

See Also

`findWISKIstations` `findWISKItimeseries`

Examples

```
groups <- findWISKIgroups()
```

`findWISKIstations` *Finds all stations on the Wiski server*

Description

Finds all stations on the Wiski server

Usage

```
findWISKIstations(stationName = "*",
  site.url = "http://giws.usask.ca:8080/")
```

Arguments

`stationName` Required. Name of station to search for. May contain wildcards. The defaul is '`*`', which will search for all of the stations on the server.

`site.url` Optional. A character string containing the url of the WISKI web server. Defaults to the Global Institute for Water Security (GIWS) server: `'http://giws.usask.ca:8080/'`. As this package is intended for use by the GIWS hydrological community, it is usually unnecessary to specify the web server.

Value

If unsuccessful, returns `FALSE`. If successful, returns a dataframe containing

`station_name` Name of station, specified by `stationName`

`station_no` WISKI number of station

`station_id` WISKI station ID number

`station_latitude` Latitude of station

`station_longitude` Longitude of station

Author(s)

Kevin Shook

See Also

findWISKItimeseries findWISKIgroups

Examples

```
stations <- findWISKIstations()
```

findWISKItimeseries *Find all timeseries on WISKI web server for a specified station*

Description

Queries a WISKI web server to get a list of available time series which have the specified station name.

Usage

```
findWISKItimeseries(stationName, site.url = "http://giws.usask.ca:8080/")
```

Arguments

stationName	Required. A character string containing the name of the station. May contain wildcard characters.
site.url	Optional. A character string containing the url of the WISKI web server. Defaults to the Global Institute for Water Security (GIWS) server: 'http://giws.usask.ca:8080/'. As this package is intended for use by the GIWS hydrological community, it is usually unnecessary to specify the web server.

Value

If unsuccessful, returns FALSE. If successful, returns a dataframe containing

station_name	Name of station, specified by stationName
station_no	WISKI number of station
station_id	WISKI station ID number
ts_id	WISKI time series ID number
ts_name	WISKI time series name
parametertype_id	WISKI ID number of parameter type of time series
parametertype_name	WISKI name of parameter type of time series

Author(s)

Kevin Shook

See Also

getWISKImetadata getWISKIvalues

Examples

```
FiseraTimeseries <- findWISKItimeseries('Fisera*')
print(FiseraTimeseries)
```

---

getWISKImetadata *Gets metadata of a WISKI time series*

Description

Returns metadata of a specified time series.

Usage

```
getWISKImetadata(timeSeries, site.url = "http://giws.usask.ca:8080/")
```

Arguments

timeSeries	Required. Character string containing the WISKI time series ID number, which is returned by findWISKItimeseries. Cannot contain wild card characters.
site.url	Optional. A character string containing the url of the WISKI web server. Defaults to the Global Institute for Water Security (GIWS) server: 'http://giws.usask.ca:8080/'. As this package is intended for use by the GIWS hydrological community, it is usually unnecessary to specify the web server.

Value

If unsucessful, returns FALSE. If successful, returns a dataframe containing the metadata of the specified time series.

Author(s)

Kevin Shook

See Also

getWISKIvalues findWISKItimeseries

Examples

```
# Finds metadata for Fisera Ridge air temperatures (Original time series)
FiseraTmetadata <- getWISKImetadata('9296042')
print(FiseraTmetadata)
```

getWISKIvalues *Gets values of a WISKI time series*

Description

Returns times, values, and quality codes for a specified interval of a specified time series.

Usage

```
getWISKIvalues(timeSeries = "", startDate = "1900-01-01", endDate = "",
  timezone = "", site.url = "http://giws.usask.ca:8080/")
```

Arguments

`timeSeries`	Required. Character string containing the WISKI time series ID number, which is returned by `findWISKItimeseries`. Cannot contain wild card characters.
`startDate`	Optional. Character vector of the starting date of data being queried. Must be in the form `'yyyy-mm-dd'`. The default value is `'1900-01-01'`.
`endDate`	Optional. Character vector of the ending date of data being queried. Must be in the form `'yyyy-mm-dd'`. The default value is today's date.
`timezone`	Required. Time zone for the data. Can be any time zone string sed by `Java` such as '`GMT+5`', or '`MST`'.
`site.url`	Optional. A character string containing the url of the WISKI web server. Defaults to the Global Institute for Water Security (GIWS) server: `'http://giws.usask.ca:8080/'`. As this package is intended for use by the GIWS hydrological community, it is usually unnecessary to specify the web server.

Value

If unsuccessful, returns `FALSE`. If successful, returns a dataframe with three variables:

`time`	R time value
`variable name`	time series values
`QualityCode`	time series quality codes

Author(s)

Kevin Shook

See Also

`getWISKImetadata` `findWISKItimeseries`

Examples

```
# get values for Fisera Ridge air temperatures (Original time series)
FiseraTvalues <- getWISKIvalues('9296042','2013-01-01','2013-06-30', timezone='CST')
summary(FiseraTvalues)
```

WISKItoObs *Converts a WISKI time series to an obs dataframe*

Description

Converts the WISKI date/time to a CRHM datetime. If possible, it will also convert the variable name to a CRHM variable name. Note that this may be incorrect, as it is attempting to infer the variable from its units.

Usage

```
WISKItoObs(WISKItimeseries, timezone = "", varName = "")
```

Arguments

`WISKItimeseries`
: Required. Dataframe containing the WISKI time series, as returned by getWISKIvalues.

`timezone`
: Required. The name of the timezone of the data as a character string. This should be the timezone of your data, but omitting daylight savings time. Note that the timezone code is specific to your OS. To avoid problems, you should use a timezone without daylight savings time. Under Linux, you can use ‘`CST`’ and ‘`MST`’ for Central Standard or Mountain Standard time, respectively. Under Windows or OSX, you can use ‘`etc/GMT+6`’ or ‘`etc/GMT+7`’ for Central Standard and Mountain Standard time. DO NOT use ‘`America/Regina`’ as the time zone, as it includes historical changes between standard and daylight savings time.

`varName`
: Optional. Name for the variable. If not specified (the default) then a name will be guessed from the units. Note that many different variables have the same units, so the guess might not be very good!

Author(s)

Kevin Shook

See Also

`getWISKIvalues`

Examples

```
FiseraTvalues <- getWISKIvalues('9296042','2013-01-01','2013-06-30', timezone='CST')
FiseraTobs <- WISKItoObs(FiseraTvalues, timezone='Etc/GMT+7')
```

Index

findWISKIgroups, 2, *4*
findWISKIstations, *3*, 3
findWISKItimeseries, *2–4*, 4, *5, 6*

getWISKImetadata, *2*, *5*, 5, *6*
getWISKIvalues, *2*, *5*, 6, *7*

WISKIr-package, 2
WISKItoObs, *2*, 7