# Package ‘RivRetrieve’

**Title** Retrieve Global River Gauge Data

**Version** 0.1.3

**Description** Provides access to global river gauge data from a variety of national-level river agencies. The package interfaces with the national-level agency websites to provide access to river gauge locations, river discharge, and river stage. Currently, the package is available for the following countries: Australia, Brazil, Canada, Chile, France, Japan, South Africa, the United Kingdom, and the United States.

**License** MIT + file LICENSE

**URL** [https://github.com/Ryan-Riggs/RivRetrieve](https://github.com/Ryan-Riggs/RivRetrieve)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Depends** R (>= 4.1)

**Imports** BBmisc, dataRetrieval, devtools, dplyr, htr, jsonlite, lubridate, readr, rlang, RSElenium, rvest, stringr, tibble, tidyhydat, tidyr

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Ryan Riggs [aut, cre] ([https://orcid.org/0000-0001-6834-9469](https://orcid.org/0000-0001-6834-9469)), Simon Moulds [aut] ([https://orcid.org/0000-0002-7297-482X](https://orcid.org/0000-0002-7297-482X)), Michel Wortmann [aut] ([https://orcid.org/0000-0002-1879-7674](https://orcid.org/0000-0002-1879-7674)), Louise Slater [aut] ([https://orcid.org/0000-0001-9416-488X](https://orcid.org/0000-0001-9416-488X)), George Allen [aut] ([https://orcid.org/0000-0001-8301-5301](https://orcid.org/0000-0001-8301-5301))

**Maintainer** Ryan Riggs <rriggs@tamu.edu>

**Repository** CRAN

**Date/Publication** 2023-11-14 20:40:02 UTC
Description

Provides access to global river gauge data from a variety of national-level river agencies. The package interfaces with the national-level agency websites to provide access to river gauge locations, river discharge, and river stage. Currently, the package is available for the following countries: Australia, Brazil, Canada, Chile, France, Japan, South Africa, the United Kingdom, and the United States.

Author(s)

Maintainer: Ryan Riggs <rriggs@tamu.edu> (ORCID)
Authors:

- Simon Moulds <sim.moulds@gmail.com> (ORCID)
- Michel Wortmann <michel.wortmann@ouce.ox.ac.uk> (ORCID)
- Louise Slater <louise.slater@ouce.ox.ac.uk> (ORCID)
- George Allen <geoallen@vt.edu> (ORCID)
australia

See Also

Useful links:

- https://github.com/Ryan-Riggs/RivRetrieve

Examples

```r
## Not run:
print("TODO")

## End(Not run)
```

---

australia  australia

Description

Retrieve Australian gauge data

Usage

```r
australia(
  site,
  variable = "discharge",
  start_date = NULL,
  end_date = NULL,
  sites = FALSE,
  ...
)
```

Arguments

- `site`  Australian gauge number
- `variable`  Character. Either stage or discharge.
- `start_date`  Character. Optional start date with format YYYY-MM-DD. Default is 1900-01-01.
- `end_date`  Character. End date with format YYYY-MM-DD. Default is the current date.
- `sites`  Logical. If TRUE, returns a list of measurement sites.
- `...`  Additional arguments. None implemented.

Value

data frame of discharge time-series
Examples

```r
sites <- australia(sites = TRUE)
df <- australia(sites$sites[1], "stage")
plot(df$Date, df$H, type='l')
```

Description

Retrieve Brazilian gauge data

Usage

```r
brazil(
  site,
  variable = "discharge",
  start_date = NULL,
  end_date = NULL,
  sites = FALSE,
  ...
)
```

Arguments

- `site`: Brazilian gauge number
- `variable`: Character. Either `stage` or `discharge`.
- `start_date`: Character. Optional start date with format YYYY-MM-DD. Default is 1900-01-01.
- `end_date`: Character. End date with format YYYY-MM-DD. Default is the current date.
- `sites`: Logical. If TRUE, returns a list of measurement sites.
- `...`: Additional arguments. None implemented.

Value

Data frame of discharge time-series

Examples

```r
## Not run:
df <- brazil("12650000")
plot(df$Date, df$Q, type='l')
## End(Not run)
```
Description

Retrieve Canadian gauge data

Usage

canada(
  site,
  variable = "discharge",
  start_date = NULL,
  end_date = NULL,
  sites = FALSE,
  ...
)

Arguments

site Canadian gauge number
variable Character. Either stage or discharge.
start_date Character. Optional start date with format YYYY-MM-DD. Default is 1900-01-01.
end_date Character. End date with format YYYY-MM-DD. Default is the current date.
sites Logical. If TRUE, returns a list of measurement sites.
... Additional arguments. None implemented.

Value
data frame of discharge time-series

Examples

## Not run:
#For the first time, you must run:
tidyhydat::download_hydat()
df = canada("01AD003")
plot(df$Date, df$Q, type='l')

## End(Not run)
Description

Retrieve Chilean gauge data

Usage

chile(
  site,
  variable = "discharge",
  start_date = NULL,
  end_date = NULL,
  sites = FALSE,
  ...
)

Arguments

<table>
<thead>
<tr>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>site</td>
<td>Chilean gauge number</td>
</tr>
<tr>
<td>variable</td>
<td>Character. Either stage or discharge.</td>
</tr>
<tr>
<td>start_date</td>
<td>Character. Optional start date with format YYYY-MM-DD. Default is 1900-01-01.</td>
</tr>
<tr>
<td>end_date</td>
<td>Character. End date with format YYYY-MM-DD. Default is the current date.</td>
</tr>
<tr>
<td>sites</td>
<td>Logical. If TRUE, returns a list of measurement sites.</td>
</tr>
<tr>
<td>...</td>
<td>Additional arguments. None implemented.</td>
</tr>
</tbody>
</table>

Value

data frame of discharge time-series

Examples

```r
df <- chile('01201005')
plot(df$Date, df$Q, type='l')
```
Description

Retrieve French gauge data

Usage

\[
\text{france}(\text{site, variable = "discharge", start_date = NULL, end_date = NULL, sites = FALSE, ... })
\]

Arguments

- **site**: French gauge number
- **variable**: Character. Either stage or discharge.
- **start_date**: Character. Optional start date with format YYYY-MM-DD. Default is 1900-01-01.
- **end_date**: Character. End date with format YYYY-MM-DD. Default is the current date.
- **sites**: Logical. If TRUE, returns a list of measurement sites.
- **...**: Additional arguments. None implemented.

Value

data frame of discharge time-series

Examples

\[
\text{df} \leftarrow \text{france('K027401001')}
\]

\[
\text{plot(df$Date, df$Q, type='l')}
\]
Description

get_as_stored

Usage

```r
get_as_stored(
  parameter_type,
  station_number,
  start_date,
  end_date,
  tz,
  return_fields
)
```

Arguments

- **parameter_type**: Parameter value
- **station_number**: Station number
- **start_date**: Start date (formatted as YYYY-MM-DD) or just the year (YYYY)
- **end_date**: End date (formatted as YYYY-MM-DD) or just the year (YYYY)
- **tz**: TZ
- **return_fields**: Return fields

Value

A tibble with the requested return fields. Zero row tibbles are returned if no data is available for
the requested dates. The aggregation of data is generally the mean for most variables, except for
rainfall and evaporation which is the sum over the chosen period.

Examples

```r
# Groundwater level as stored by the BoM
# PLUMB RD @ NARRABRI
## Not run:
get_as_stored(
  parameter_type = "Ground Water Level",
  station_number = "GW971623.3.3",
  start_date = "2020-03-01",
  end_date = "2020-03-01"
)
## End(Not run)
```
get_daily

Description
get_daily

Usage
get_daily(
  parameter_type,
  station_number,
  start_date,
  end_date,
  var,
  aggregation,
  tz,
  return_fields
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter_type</td>
<td>Parameter value</td>
</tr>
<tr>
<td>station_number</td>
<td>Station number</td>
</tr>
<tr>
<td>start_date</td>
<td>Start date (formatted as YYYY-MM-DD) or just the year (YYYY)</td>
</tr>
<tr>
<td>end_date</td>
<td>End date (formatted as YYYY-MM-DD) or just the year (YYYY)</td>
</tr>
<tr>
<td>var</td>
<td>The daily variable of interest. Valid inputs are mean, min, max for continuous series such as discharge and total for discrete series such as rainfall and evaporation.</td>
</tr>
<tr>
<td>aggregation</td>
<td>Whether the data is to be aggregated midnight to midnight (24HR) or from 9am-9am (09HR). The default is 24HR. 09HR is only available for mean discharge and total rainfall and evaporation.</td>
</tr>
<tr>
<td>tz</td>
<td>TZ</td>
</tr>
<tr>
<td>return_fields</td>
<td>Return fields</td>
</tr>
</tbody>
</table>

Value
A tibble with the requested return fields. Zero row tibbles are returned if no data is available for the requested dates. The aggregation of data is generally the mean for most variables, except for rainfall and evaporation which is the sum over the chosen period.
Examples

# Download daily mean aggregated over the standard day
## Not run:
get_daily(
  parameter_type = "Water Course Discharge",
  station_number = "410730",
  start_date = "2020-01-01",
  end_date = "2020-01-31",
  var = "mean",
  aggregation = "24HR"
)

## End(Not run)

# Download daily mean aggregated between 9am to 9am
## Not run:
get_daily(
  parameter_type = "Water Course Discharge",
  station_number = "410730",
  start_date = "2020-01-01",
  end_date = "2020-01-31",
  var = "mean",
  aggregation = "09HR"
)

## End(Not run)

# Download the daily max over the standard day
## Not run:
get_daily(
  parameter_type = "Water Course Discharge",
  station_number = "410730",
  start_date = "2020-01-01",
  end_date = "2020-01-31",
  var = "max",
  aggregation = "24HR"
)

## End(Not run)

---

**get_hourly**

**Description**

get_hourly
get_monthly

Usage

```r
get_hourly(
    parameter_type,
    station_number,
    start_date,
    end_date,
    tz,
    return_fields
)
```

Arguments

- parameter_type: Parameter value
- station_number: Station number
- start_date: Start date (formatted as YYYY-MM-DD) or just the year (YYYY)
- end_date: End date (formatted as YYYY-MM-DD) or just the year (YYYY)
- tz: TZ
- return_fields: Return fields

Value

A tibble with the requested return fields. Zero row tibbles are returned if no data is available for the requested dates. The aggregation of data is generally the mean for most variables, except for rainfall and evaporation which is the sum over the chosen period.

Examples

```r
# Hourly streamflow Cotter River at Gingera Gauge
## Not run:
get_hourly(
    parameter_type = "Water Course Discharge",
    station_number = "410730",
    start_date = "2020-01-01",
    end_date = "2020-01-31",
    tz,
    return_fields
)
## End(Not run)
```

### Code chunk

```r
get_monthly
```
Usage

get_monthly(
  parameter_type,
  station_number,
  start_date,
  end_date,
  tz,
  return_fields
)

Arguments

parameter_type  Parameter value
station_number   Station number
start_date       Start date (formatted as YYYY-MM-DD) or just the year (YYYY)
end_date         End date (formatted as YYYY-MM-DD) or just the year (YYYY)
tz               TZ
return_fields    Return fields

Value

A tibble with the requested return fields. Zero row tibbles are returned if no data is available for the requested dates. The aggregation of data is generally the mean for most variables, except for rainfall and evaporation which is the sum over the chosen period.

Examples

# Monthly average dry air temperature at Corin Dam
## Not run:
get_monthly(
  parameter_type = "Dry Air Temperature",
  station_number = "570947",
  start_date = "2016-01-01",
  end_date = "2016-06-01"
)
## End(Not run)

get_parameter_list 

Retrieve available parameters for stations

Description

get_parameter_list returns the parameters that can be retrieved at one or more stations.
Usage

get_parameter_list(station_number, return_fields)

Arguments

station_number  A single or multiple vector of AWRC station numbers.
return_fields   (Optional) Station parameter details to be returned. By default the return fields
                are: station_no, station_id, station_name, parametertype_id, parametertype_name,
                parametertype_unitname parametertype_shortunitname.

Details

The default return fields have been selected to generally return the most useful fields while reducing
duplication of metadata. The full list of return fields:

- station_no',
- station_id
- station_name
- stationparameter_id
- stationparameter_no
- stationparameter_name
- stationparameter_longname
- site_id
- site_no
- site_name
- parametertype_id
- parametertype_name
- parametertype_longname
- parametertype_unitname
- parametertype_shortunitname

Value

A tibble with columns for each of the return fields.

Examples

# Return parameters for a single station
## Not run:
get_parameter_list(station_number = "410730")

## End(Not run)
# Return available parameters for multiple stations
## Not run:
get_parameter_list(station_number = c("410730", "570946"))

## End(Not run)
get_station_list

Retrieve water observation stations

Description

get_station_list queries Water Data Online and returns station details. Queries can be input with the desired parameter_type to find all the stations on record. If you already have a vector of station numbers, you can pass the vector to station_number and return the details of those stations. return_fields can be customised to return various data about the stations.

Usage

get_station_list(parameter_type, station_number, bbox, return_fields)

Arguments

parameter_type  The parameter for the station request (e.g. Water Course Discharge, Storage Level)
station_number  Optional: a single or multiple vector of AWRC station numbers.
bbox  Optional: a bounding box to get stations in a region of interest. Takes a vector ordered xmin, ymin, xmax, ymax.
return_fields  Station details to be returned. By default the columns returned are station name, number, ID, latitude and longitude. Can be customised with a vector of parameters.

Details

Possible return fields for get_station_list() are:

- station_name
- station_longname
- station_no
- station_id
- station_latitude
- station_longitude
- station_carteasting
- station_cartnorthing
- stationparameter_name
- station_georefsystem
- catchment_no
- catchment_id
- catchment_name
- site_no
get_station_list

- site_id
- site_name
- parametertype_id
- parametertype_name
- object_type
- custom_attributes

Value

With the default return fields, a tibble with columns station_name, station_no, station_id, station_latitude, station_longitude.

Examples

# Get all Water Course Discharge Stations
## Not run:
get_station_list()

## End(Not run)

# Just the details for Cotter River at Gingera
## Not run:
get_station_list(station_number = "410730")

## End(Not run)

# Rainfall stations
## Not run:
get_station_list(parameter_type = "Rainfall")

## End(Not run)

# Vector of return_fields
return_fields <- c(
  "station_name",
  "station_longname",
  "station_no",
  "station_id",
  "station_latitude",
  "station_longitude",
  "station_carteasting",
  "station_cartnorthing",
  "stationparameter_name",
  "station_georefsystem",
  "catchment_no",
  "catchment_id",
  "catchment_name",
  "site_no",
  "site_id",
  "site_name",
  "parametertype_id",
  "parametertype_name",
  "object_type",
  "custom_attributes"
get_timeseries

## Not run:
get_station_list("Water Course Discharge", "410730", return_fields)

## End(Not run)

Description

Get timeseries data from Water Data online

Usage

```r
get_timeseries(
  parameter_type, 
  station_number, 
  start_date, 
  end_date, 
  tz, 
  return_fields, 
  ts_name
)
```

Arguments

- **parameter_type**: The water data parameter type (e.g. Water Course Discharge). See `parameters()` for a full list.
- **station_number**: The AWRC station number.
- **start_date**: Start date formatted as a string or date class (YYYY-MM-DD).
- **end_date**: End date formatted as a string or date class (YYYY-MM-DD).
- **tz**: Optional: the desired time zone for the output timeseries. Input must be an Olson Name (see `OlsonNames()`). By default the timeseries are returned in non-DST time zones (AEST, ACST or AWST) depending on the station location.
- **return_fields**: Optional: columns to be returned from Water Data Online. By default Times-tamp, Value and Quality Code are returned.
- **ts_name**: The timeseries name (e.g. DMQaQc.Merged.DailyMean.24HR) that is desired.

Details

This function can be used if you want to retrieve a specific timeseries that is not the default quality checked one.

Common valid return fields are:
get_timeseries

- Timestamp
- Value
- Quality Code
- Interpolation Type

Other valid return fields (depending on the parameter requested) may be:

- Absolute Value
- AV Interpolation
- AV Quality Code
- Runoff Value
- RV Interpolation
- RV Quality Code
- Aggregation
- Accuracy

If the request is not valid it will fail.

Value

A tibble with columns with the requested return_fields. A zero row tibble is returned if no data is returned from the query. The columns of the tibble are returned as character classes and have not been formatted to more appropriate correct classes (this happens in other functions).

Examples

# Accessible dam storage, as shown on the BoM Water Storage dashboard
## Not run:
```r
get_timeseries(
  parameter_type = "Storage Volume",
  "G8150011",
  "2020-01-01",
  "2020-01-31",
  ts_name = "PR02AVQaQc.Merged.DailyMean.24HR",
  tz = NULL,
  return_fields = c("Timestamp", "Value", "Quality Code")
)
```

## End(Not run)
get\_timeseries\_id \hspace{1cm} Retrieve the timeseries ID

Description

get\_timeseries\_id retrieves the timeseries ID that can be used to obtain values for a parameter type, station and timeseries combination.

Usage

get\_timeseries\_id(parameter\_type, station\_number, ts\_name)

Arguments

parameter\_type \hspace{1cm} The parameter of interest (e.g. Water Course Discharge).
station\_number \hspace{1cm} The AWRC station number.
pt_name \hspace{1cm} The BoM time series name (e.g. DMQaQc.Merged.DailyMean.24HR).

Value

Returns a tibble with columns station\_name, station\_no, station\_id, ts\_id, ts\_name, parameter\_type\_id, parameter\_type\_name.

get\_timeseries\_values \hspace{1cm} Retrieve timeseries values

Description

get\_timeseries\_values returns the timeseries values between a start and end date for given timeseries ID.

Usage

get\_timeseries\_values(ts\_id, start\_date, end\_date, return\_fields)

Arguments

ts\_id \hspace{1cm} The timeseries ID for the values of interest. Can be found using the function get\_timeseries\_id.
start\_date \hspace{1cm} The start date formatted as 'YYYY-MM-DD'.
end\_date \hspace{1cm} The end date formatted as 'YYYY-MM-DD'.
return\_fields \hspace{1cm} A vector of the variables that are to be returned.
get_yearly

Value

A tibble with columns with the requested return_fields. A zero row tibble is returned if no data is returned from the query. The columns of the tibble are returned as character classes and have not been formatted to more appropriate correct classes (this happens in other functions).

get_yearly

Description

get_yearly

Usage

get_yearly(
  parameter_type,
  station_number,
  start_date,
  end_date,
  tz,
  return_fields
)

Arguments

  parameter_type  Parameter value
  station_number  Station number
  start_date      Start date (formatted as YYYY-MM-DD) or just the year (YYYY)
  end_date        End date (formatted as YYYY-MM-DD) or just the year (YYYY)
  tz              TZ
  return_fields   Return fields

Value

A tibble with the requested return fields. Zero row tibbles are returned if no data is available for the requested dates. The aggregation of data is generally the mean for most variables, except for rainfall and evaporation which is the sum over the chosen period.

Examples

  # Download annual rainfall for Cotter Hut
  ## Not run:
  get_yearly(
    parameter_type = "Rainfall",
    station_number = "570946",
    start_date = 2016,
Description

Retrieve Japanese gauge data

Usage

japan(
  site,
  variable = "discharge",
  start_date = NULL,
  end_date = NULL,
  sites = FALSE,
  ...
)

Arguments

site                      Japanese gauge number
variable                  Character. Either stage or discharge.
start_date                Character. Optional start date with format YYYY-MM-DD. Default is 1900-01-01.
end_date                  Character. End date with format YYYY-MM-DD. Default is the current date.
sites                     Logical. If TRUE, returns a list of measurement sites.
...                       Additional arguments. None implemented.

Value

data frame of discharge time-series

Examples

## Not run:
start_date <- as.Date("2019-01-01")
end_date <- as.Date("2022-12-31")
df <- japan("301011281104010", "discharge", start_date, end_date)
plot(df$Date, df$Q, type='l')

## End(Not run)
**make_bom_request**

*Query the BoM WISKI API*

**Description**

This function queries the Bureau of Meteorology Water Data KISTERS API. A parameter list is passed to make request and the JSON return is parsed depending on what is requested. This function can be used if you want to build your own JSON queries.

**Usage**

```r
make_bom_request(params)
```

**Arguments**

- `params` A named list of parameters.

**Value**

A tibble is returned with the columns depending on the request. For `get_timeseries` requests, a tibble with zero rows is returned if there is no data available for that query.

---

**original**

*Get original data*

**Description**

Get original data

**Usage**

```r
original(x, ...)
```

**Arguments**

- `x` Tibble.

- `...` Additional arguments. None implemented.

**Value**

`list`
### parameters

**Description**

`parameters` returns a vector of parameters that can be retrieved from Water Data Online.

**Usage**

```r
parameters(pars)
```

**Arguments**

- **pars**
  
  Optional: if empty all available parameters will be returned. Alternatively, a vector of the continuous or discrete parameters can be requested.

**Value**

A vector of parameters.

**Examples**

```r
parameters()
parameters("continuous")
parameters("discrete")
```

### plot.rr_tbl

**Description**

Plot values

**Usage**

```r
## S3 method for class 'rr_tbl'
plot(x, ...)
```

**Arguments**

- **x**
  
  Tibble.

- **...**
  
  Additional arguments. None implemented.

**Value**

`ggplot2`
Description

Retrieve South African gauge data

Usage

```r
southAfrica(
  site,
  variable = "stage",
  start_date = NULL,
  end_date = NULL,
  sites = FALSE,
  ...  
)
```

Arguments

- **site**  
  South African gauge number

- **variable**  
  Character. Either `stage` or `discharge`.

- **start_date**  
  Character. Optional start date with format YYYY-MM-DD. Default is 1900-01-01.

- **end_date**  
  Character. End date with format YYYY-MM-DD. Default is the current date.

- **sites**  
  Logical. If TRUE, returns a list of measurement sites.

- **...**  
  Additional arguments. None implemented.

Value

data frame of discharge time-series

Examples

```r
site <- "X3H023"
start_date <- as.Date("2000-01-01")
end_date <- as.Date("2010-01-01")
x <- southAfrica(site, "stage", start_date, end_date)
```
### Description

Retrieve UK gauge data

#### Usage

```r
uk(site, variable, start_date = NULL, end_date = NULL, sites = FALSE, ...)
```

#### Arguments

- **site**: UK gauge number
- **variable**: Character. Either stage or discharge.
- **start_date**: Character. Optional start date with format YYYY-MM-DD. Default is 1900-01-01.
- **end_date**: Character. End date with format YYYY-MM-DD. Default is the current date.
- **sites**: Logical. If TRUE, returns a list of measurement sites.
- **...**: Additional arguments. None implemented.

#### Value

data frame of discharge time-series

#### Examples

```r
site <- "http://environment.data.gov.uk/hydrology/id/stations/3c5cba29-2321-4289-a1fd-c355e135f4cb"
x <- uk(site, variable = "discharge")
plot(x$Date, x$Q, type='l')
```

---

### Description

Retrieve USA gauge data
Usage

usa(
    site,
    variable = "stage",
    start_date = NULL,
    end_date = NULL,
    sites = FALSE,
    ...
)

Arguments

site    USA gauge number
variable Character. Either stage or discharge.
start_date Character. Optional start date with format YYYY-MM-DD. Default is 1900-01-01.
end_date Character. End date with format YYYY-MM-DD. Default is the current date.
sites   Logical. If TRUE, returns a list of measurement sites.
...     Additional arguments. None implemented.

Value
data frame of discharge time-series

Examples

df <- usa("02471078", variable="discharge")
plot(df$Date, df$Q, type='l')
**Index**

australia, 3
brazil, 4
canada, 5
chile, 6
france, 7
get_as_stored, 8
get_daily, 9
get_hourly, 10
get_monthly, 11
get_parameter_list, 12
get_station_list, 14
get_timeseries, 16
get_timeseries_id, 18
get_timeseries_values, 18
get_yearly, 19
japan, 20
make_bom_request, 21
original, 21
parameters, 22
parameters(), 16
plot.rr_tbl, 22
RivRetrieve-package, 2
southAfrica, 23
uk, 24
usa, 24