Package 'Rwtss'

June 28, 2021

Title Client for Web Time-Series Service

Version 0.9.1

Maintainer Felipe Souza <lipecaso@gmail.com>

Description Allows remote access to satellite image time series provided by the web time series service (WTSS) available at servers such as <https://brazildatacube.dpi.inpe.br/wtss/>. The functions include listing the data sets available in WTSS servers, describing the contents of a data set, and retrieving a time series based on spatial location and temporal filters.

URL https://github.com/e-sensing/Rwtss/

BugReports https://github.com/e-sensing/Rwtss/issues

ByteCompile true

LazyData true

License GPL-3

Encoding UTF-8

Depends R (>= 3.6.0)

Imports assertthat, dplyr, geosphere, ggplot2, reshape2, jsonlite, lubridate, magrittr, purrr, httr, stats, stringr, tibble, zoo

RoxygenNote 7.1.1

Suggests bfast, knitr, rmarkdown, roxygen2, testthat, spelling, vcr

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Gilberto Queiroz [aut] (<https://orcid.org/0000-0001-7534-0219>), Gilberto Camara [aut] (<https://orcid.org/0000-0002-3681-487X>), Pedro Andrade [aut] (<https://orcid.org/0000-0001-8675-4046>), Felipe Souza [aut, cre], Luiz Assis [aut]

Repository CRAN

Date/Publication 2021-06-28 14:10:02 UTC

R topics documented:

`Rwtss-package` *Rwtss*

Description

An R client to the web time series service (WTSS)

Rwtss API

Implements an R interface to a web time series service (WTSS) that offers time series of remote sensing data using a simple API. A WTSS server takes as input an Earth observation data cube, that has a spatial and a temporal dimension and can be multidimensional in terms of its attributes.

The WTSS API has four commands:

- 'wtss': given an URL, creates a connection to a WTSS service
- 'list_coverages': returns a list of coverages (cubes) available in the WTSS server.
- 'describe_coverage': returns the metadata for a given coverage.
- 'time_series': returns a time series for a spatio-temporal location.

Author(s)

Maintainer: Felipe Souza <lipecaso@gmail.com>

Authors:

- Gilberto Queiroz <gilberto.queiroz@inpe.br> (ORCID)
- Gilberto Camara <gilberto.camara@inpe.br> (ORCID)
- Pedro Andrade <pedro.andrade@inpe.br> (ORCID)
- Luiz Assis <luiz.assis@inpe.br>

See Also

Useful links:

- https://github.com/e-sensing/Rwtss/
- Report bugs at https://github.com/e-sensing/Rwtss/issues

.wtss_coverage_description

Decodes the description from a WTSS coverage

Description

creates a tibble to store the description of the WTSS coverage

Usage

```
.wtss_coverage_description(URL, cov)
```

Arguments

URL	URL of the coverage
cov	coverage response provided by WTSS service

.wtss_get_response *Get a response to the WTSS server*

Description

Sends a request to the WTSS server and gets a response

Usage

```
.wtss_get_response(request, ...)
```

Arguments

request	valid request according to the WTSS protocol
...	additional parameters that can be added in httr.

Value

response from the server

.wtss_ggplot_series *Plot one timeSeries using ggplot*

Description

Plots a set of time series using ggplot. This function is used for showing the same lat/long location in a series of time steps.

Usage

```
.wtss_ggplot_series(row, colors = "Dark2")
```

Arguments

row	A row of a sits tibble with the time series to be plotted.
colors	The set of Brewer colors to be used for plotting.

.wtss_guess_satellite *Try a best guess for the type of sensor/satellite*

Description

Based on resolution, tries to guess what is the satellite.

Usage

```
.wtss_guess_satellite(xres)
```

Arguments

xres	xres of the coverage

Value

Satellite sensor pair

Author(s)

Gilberto Camara, <gilberto.camara@inpe.br>

.wtss_list_coverages *Retrieves the list of cubes from the URL server*

Description

Use the WTSS protocol to find out available coverages

Usage

```
.wtss_list_coverages(URL)
```

Arguments

URL	URL of the WTSS service

Value

updated WTSS object.

.wtss_parse_json *Parse a JSON response from the WTSS server*

Description

Parse a JSON response from the WTSS service

Usage

```
.wtss_parse_json(response)
```

Arguments

response	valid JSON response from the WTSS service

Value

parsed JSON document

.wtss_process_request *Process a request to the WTSS server*

Description

Process a request

Usage

```
.wtss_process_request(request)
```

Arguments

request	valid request to the WTSS service

Value

parsed JSON document

.wtss_remove_trailing_dash *Remove trailing dashes from a WTSS server address*

Description

The WTSS URL cannot have a trailing dash. This functions checks and removes it, if present.

Usage

```
.wtss_remove_trailing_dash(URL)
```

Arguments

URL	A WTSS URL

Value

URL without trailing dash

.wtss_send_request *Send a request to WTSS server*

Description

Sends a request to the WTSS server and times out after 10 tries

Usage

```
.wtss_send_request(request, ...)
```

Arguments

request	valid request according to the WTSS protocol
...	additional parameters that can be added in httr.

Value

response from the server

.wtss_tibble *Create a sits tibble to store the time series information*

Description

This function returns an empty tibble that contains the satellite image time series and its metadata. The columns are <longitude, latitude, start_date, end_date, label, cube, time_series>. WTSS functions produce a tibble as output.

Usage

```
.wtss_tibble()
```

Value

A tibble.

Author(s)

Gilberto Camara, <gilberto.camara@inpe.br>

.wtss_time_series_processing *Processing a Time Series Result from WTSS*

Description

Processing a Time Series Result from WTSS

Usage

```
.wtss_time_series_processing(items)
```

Arguments

items	Items retrieved from WTSS server

Value

tibble with a time series

`.wtss_to_tibble` *Import time series in the zoo format to a tibble*

Description

Converts data from an instance of a zoo series to a sits tibble.

Usage

```
.wtss_to_tibble(
  ts,
  name,
  bands,
  longitude,
  latitude,
  start_date,
  end_date,
  cov_desc
)
```

Arguments

`ts`	list of time series retrieved by WTSS
`name`	Name of the coverage where data comes from.
`bands`	Bands to be retrieved from the time series.
`longitude`	Longitude of the chosen location.
`latitude`	Latitude of the chosen location.
`start_date`	Starting date of the time series
`end_date`	End date of the time series
`cov_desc`	Description of the WTSS coverage

Value

Time series in sits tibble format.

Author(s)

Gilberto Camara, <gilberto.camara@inpe.br>

describe_coverage *Retrieves the list of cubes from the URL server*

Description

Contacts the WTSS server to describe one coverage

Usage

```
describe_coverage(URL, name, .print = TRUE)
```

Arguments

URL	URL of the server
name	name of coverage
.print	Print the coverage description

Value

tibble with coverage description

Examples

```
## Not run:
# Using external server
describe_coverage("https://brazildatacube.dpi.inpe.br/wtss/",
                  "LC8_30_16D_STK-1")

## End(Not run)
```

list_coverages *List the coverages available in the WTSS service*

Description

Lists coverages available in the WTSS service

Usage

```
list_coverages(URL)
```

Arguments

URL	URL of the server

Value

vector with coverage name

Examples

```
## Not run:
# Using external server
list_coverages("https://brazildatacube.dpi.inpe.br/wtss/")

## End(Not run)
```

---

ndvi_ts *Example time series from MOD13Q1 product.*

Description

A dataset containing a wtss tibble, with extracted time series.

Usage

```
data("ndvi_ts")
```

Format

A `wtss` tibble with 388 samples. A `wtss` tibble contains data and metadata. The first six columns contain the metadata: satellite, sensor, spatial and temporal information, and the coverage from where the data has been extracted. The spatial location is given in longitude and latitude coordinates for the "WGS84" ellipsoid. The ‘time_series‘ column contains the time series data for each spatiotemporal location.

---

plot *Generic interface for ploting time series*

Description

Given a tibble with a set of time series, plot them.

Usage

```
## S3 method for class 'wtss'
plot(x, y, ..., colors = "Dark2")
```

Arguments

x	object of class "wtss"
y	ignored
...	further specifications for plot.
colors	Color pallete to be used (based on Color Brewer - default is "Dark2").

Value

Input tibble (useful for chaining functions).

Author(s)

Gilberto Camara, <gilberto.camara@inpe.br>

Examples

```
# Access to external service
# Read one time series from the WTSS server
# plot one time series
wtss_server <- "http://www.esensing.dpi.inpe.br/wtss"
ts   <- Rwtss::time_series(wtss_server, name = "MOD13Q1",
                attributes = c("ndvi","evi"),
                longitude = -45.00, latitude  = -12.00,
                start_date = "2000-02-18", end_date = "2016-12-18")
plot(ts)
```

---

time_series *Get time series*

Description

Retrieves the time series for a pair of coordinates

Usage

```
time_series(
  URL,
  name,
  attributes = NULL,
  longitude,
  latitude,
  start_date = NULL,
  end_date = NULL,
  token = NULL,
  ...
)
```

Arguments

URL	URL of the server
name	Coverage name.
attributes	Vector of band names.
longitude	Longitude in WGS84 coordinate system.
latitude	Latitude in WGS84 coordinate system.
start_date	Start date in the format yyyy-mm-dd or yyyy-mm depending on the coverage.
end_date	End date in the format yyyy-mm-dd or yyyy-mm depending on the coverage.
token	A character with token to be add in URL.
...	Additional parameters that can be added in httr.

Value

time series in a tibble format (NULL)

Author(s)

Gilberto Camara

Examples

```
## Not run:
# connect to a WTSS server
wtss_server <- "https://brazildatacube.dpi.inpe.br/wtss/"
# retrieve a time series
ndvi_ts <- Rwtss::time_series(wtss_server,
                              "LC8_30_16D_STK-1",
                              attributes = "NDVI",
                              latitude = -14.31,
                              longitude = -51.16,
                              token = "YOUR-BDC-TOKEN")
# plot the time series
plot(ndvi_ts)

## End(Not run)
```

---

wtss_to_ts *Export data to be used to the ts format*

Description

Converts data from a wtss tibble to a time series "ts". A WTSS tibble contains data retrieved from a WTSS server. These data sets are time series with irregular intervals. Given that of many functions that use the R "ts" format, this function converts a time series (a tibble with data and metadata) to the "ts" format. Since "ts" requires regular time series, it interpolates the original irregular time series to a regular time series. To do this, the user needs to specify a period which is recognised by the "ts" format. This period can be either "month", "week", "day", "months", "weeks", "days" or 12, 52, 365. This function creates a new time series with the required frequency and intepolates the missing values using spline interpolation from the "zoo" package (zoo::na.spline).

Usage

```
wtss_to_ts(data, band = NULL, period = "week")
```

Arguments

data	A sits tibble with time series.
band	Name of the band to be exported (optional if series has only one band)
period	One of c("month", "week", "day"), c("months", "weeks", "days") or c(12, 52, 365)

Value

A time series in the ts format.

Author(s)

Gilberto Camara, <gilberto.camara@inpe.br>

Examples

```
# connect to a WTSS server
wtss <- "http://www.esensing.dpi.inpe.br/wtss"
# retrieve a time series
ts_wtss  <- Rwtss::time_series(wtss, "MOD13Q1", c("ndvi","evi"),
                longitude = -45.00, latitude  = -12.00,
                start_date = "2000-02-18", end_date = "2016-12-18")
# convert to ts
ts <- Rwtss::wtss_to_ts(ts_wtss, band = "ndvi")
```

`wtss_to_zoo` *Export data to be used to the zoo format*

Description

Converts data from a tibble to a list of a zoo series.

Usage

```
wtss_to_zoo(data, band = NULL)
```

Arguments

`data`	A tibble with time series.
`band`	Name of the band to be exported (if NULL all bands are exported).

Value

List of time series in zoo format.

Author(s)

Gilberto Camara, <gilberto.camara@inpe.br>

Examples

```
# retrieve a time series
ts_wtss  <- Rwtss::time_series("http://www.esensing.dpi.inpe.br/wtss",
                "MOD13Q1", c("ndvi","evi"),
                longitude = -45.00, latitude  = -12.00,
                start_date = "2000-02-18", end_date = "2016-12-18")
# convert to zoo
zoo.lst <- Rwtss::wtss_to_zoo(ts_wtss)
```

`%>%` *Pipe*

Description

Magrittr compound assignment pipe-operator.

Arguments

`lhs, rhs`	A visualisation and a function to apply to it.

Index