Package: dataaimsr (via r-universe)

October 28, 2024

Type Package

Title AIMS Data Platform API Client

Version 1.1.0

Description AIMS Data Platform API Client which provides easy access to AIMS Data Platform scientific data and information.

Depends R (>= 3.3.0)

Imports httr, jsonlite, parsedate, dplyr, tidyr, rnaturalearth, sf, ggplot2, ggrepel, curl, rlang

Suggests httptest, rgeos, knitr, purrr, rmarkdown, testthat

License MIT + file LICENSE

Encoding UTF-8

LazyData true

VignetteBuilder knitr

URL https://ropensci.github.io/dataaimsr/, https://open-aims.github.io/data-platform/key-request, https://open-aims.github.io/data-platform/swagger/

BugReports https://github.com/ropensci/dataaimsr/issues

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

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

RemoteUrl https://github.com/ropensci/dataaimsr

RemoteRef master

RemoteSha 1a6132e76b918744ae7fde7cd1146b105785ac4b

Contents

`dataaimsr-package` *The 'dataaimsr' package.*

Description

dataaimsr is the Australian Institute of Marine Science (AIMS) Data Platform R package, and provides the user with easy access to datasets from the AIMS Data Platform API. Please see ?aims_data for more details.

References

Australian Institute of Marine Science (AIMS). (2017). AIMS Sea Water Temperature Observing System (AIMS Temperature Logger Program) https://doi.org/10.25845/5b4eb0f9bb848

Australian Institute of Marine Science (AIMS). (2017). Northern Australia Automated Marine Weather and Oceanographic Stations, https://doi.org/10.25845/5c09bf93f315d

---

`aimsdf-class` *Class* `aimsdf` *of data.frame downloaded by the* **dataaimsr** *package*

Description

Datasets downloaded by the `dataaimsr` package inherit the `aimsdf` class, which is data.frame with three attributes.

Details

See `methods(class = "aimsdf")` for an overview of available methods.

See Also

aims_data

---

aims_citation *Extracts citation attribute from object of class aimsdf*

Description

Extracts citation attribute from object of class aimsdf

Usage

```
aims_citation(df_)
```

Arguments

df_	A data.frame of class aimsdf created by function aims_data

Details

This function retrieves the citation attribute from an aimsdf object. If the input aimsdf object is a summary data.frame (see ?aims_data), then output will be an empty string.

Value

A character vector.

Author(s)

AIMS Datacentre <adc@aims.gov.au>

---

aims_data *Request data via the AIMS Data Platform API*

Description

A function that communicates with the the AIMS Data Platform via the AIMS Data Platform API

Usage

```
aims_data(target, filters = NULL, summary = NA, ...)
```

Arguments

target	A character vector of length 1 specifying the dataset. Only weather or temp_loggers are currently allowed.
filters	A list containing a set of filters for the data query (see Details).
summary	Should summary tables ("summary-by-series" or "summary-by-deployment") or daily aggregated data ("daily") be returned instead of full data (see Details)?
...	Currently unused. Additional arguments to be passed to non-exported internal functions.

Details

The AIMS Data Platform R Client provides easy access to data sets for R applications to the AIMS Data Platform API. The AIMS Data Platform requires an API Key for requests, which can be obtained at this link. It is preferred that API Keys are not stored in code. We recommend storing the environment variable AIMS_DATAPLATFORM_API_KEY permanently under the user's .Renviron file in order to load the API Key automatically.

There are two types of data currently available through the AIMS Data Platform API: Weather and Sea Water Temperature Loggers. They are searched internally via unique DOI identifiers. Only one data type at a time can be passed to the argument target.

A list of arguments for filters can be exposed for both Weather and Sea Water Temperature Loggers using function aims_expose_attributes.

Note that at present the user can inspect the range of dates for the temperature loggers data only (see usage of argument summary in the examples below). For that, the argument summary must be either the string "summary-by-series" or "summary-by-deployment". In those cases, time filters will be ignored.

Details about available dates for each dataset and time series can be accessed via Metadata on AIMS Data Platform API. We raise this caveat here because these time boundaries are very important; data are collected at very small time intervals, a window of just a few days can yield very large datasets. The query will return and error if it reaches the system's memory capacity.

For that same reason, from version 1.1.0 onwards, we are offering the possibility of downloading a mean daily aggregated version. For that, the user must set summary = "daily". In this particular case, query filter will be taken into account.

Value

aims_data returns a data.frame of class aimsdf.

If summary %in% c("summary-by-series", "summary-by-deployment"), the output shows the summary information for the target dataset (i.e. weather or temperature loggers) (NB: currently, summary only works for the temperature logger database). If summary is *not* passed as an additional argument, then the output contains **raw** monitoring data. If summary = "daily", then the output contains **mean daily aggregated** monitoring data. The output also contains five attributes (empty strings if summary is passed as an additional argument):

- metadataa DOI link containing the metadata record for the data series.
- citationthe citation information for the particular dataset.

- parametersThe measured parameters comprised in the output.
- typeThe type of dataset. Either "monitoring" if summary is not specified, "monitoring (daily aggregation)" if summary = "daily", or a "summary-by-" otherwise.
- targetThe input target.

Author(s)

AIMS Datacentre <adc@aims.gov.au>

See Also

aims_citation, aims_metadata, aims_parameters

Examples

```
## Not run:
library(dataaimsr)
# assumes that user already has API key saved to
# .Renviron

# start downloads:
# 1. downloads weather data from
# site Yongala
# within a defined date range
wdf_a <- aims_data("weather", api_key = NULL,
                   filters = list(site = "Yongala",
                                  from_date = "2018-01-01",
                                  thru_date = "2018-01-02"))

# 2. downloads weather data from all sites
# under series_id 64 from Davies Reef
# within a defined date range
wdf_b <- aims_data("weather", api_key = NULL,
                   filters = list(series_id = 64,
                                  from_date = "1991-10-18",
                                  thru_date = "1991-10-19"))
head(wdf_b)
range(wdf_b$time)

# 3. downloads weather data from all sites
# under series_id 64 from Davies Reef
# within defined date AND time range
wdf_c <- aims_data("weather", api_key = NULL,
                   filters = list(series_id = 64,
                                  from_date = "1991-10-18T06:00:00",
                                  thru_date = "1991-10-18T12:00:00"))
head(wdf_c)
range(wdf_c$time)

# 4. downloads all parameters from all sites
# within a defined date range
wdf_d <- aims_data("weather", api_key = NULL,
```

```
                  filters = list(from_date = "2003-01-01",
                                 thru_date = "2003-01-02"))
# note that there are multiple sites and series
# so in this case, because we did not specify a specific
# parameter, series within sites could differ by both
# parameter and depth
head(wdf_d)
unique(wdf_d[, c("site", "series_id", "series")])
unique(wdf_d$parameter)
range(wdf_d$time)

# 5. downloads chlorophyll from all sites
# within a defined date range
wdf_e <- aims_data("weather", api_key = NULL,
                  filters = list(parameter = "Chlorophyll",
                                 from_date = "2018-01-01",
                                 thru_date = "2018-01-02"))
# note again that there are multiple sites and series
# however in this case because we did specify a specific
# parameter, series within sites differ by depth only
head(wdf_e)
unique(wdf_e[, c("site", "series_id", "series", "depth")])
unique(wdf_e$parameter)
range(wdf_e$time)

# 6. downloads temperature data
# summarised by series
sdf_a <- aims_data("temp_loggers", api_key = NULL,
                  summary = "summary-by-series")
head(sdf_a)
dim(sdf_a)

# 7. downloads temperature data
# summarised by series
# for all sites that contain data
# within a defined date range
sdf_b <- aims_data("temp_loggers", api_key = NULL,
                  summary = "summary-by-series",
                  filters = list("from_date" = "2018-01-01",
                                 "thru_date" = "2018-12-31"))
head(sdf_b)
dim(sdf_b) # a subset of sdf_a

# 8. downloads temperature data
# summarised by deployment
sdf_c <- aims_data("temp_loggers", api_key = NULL,
                  summary = "summary-by-deployment")
head(sdf_c)
dim(sdf_c)

# 9. downloads temperature data
# within a defined date range, averaged by day
sdf_d <- aims_data("temp_loggers", api_key = NULL, summary = "daily",
```

```
                    filters = list(series = "DAVFL1",
                                   from_date = "2018-01-01",
                                   thru_date = "2018-01-10"))
  # note again that there are multiple sites and series
  # however in this case because we did specify a specific
  # parameter, series within sites differ by depth only
  head(sdf_d)
  unique(sdf_d[, c("site", "series_id", "series", "depth")])
  unique(sdf_d$parameter)
  range(sdf_d$time)

  ## End(Not run)
```

`aims_expose_attributes` *Expose available query filters*

Description

Expose available query filters which are allowed to be parsed either via argument `summary` or `filters` in `aims_data`

Usage

```
aims_expose_attributes(target)
```

Arguments

`target`	A `character` vector of length 1 specifying the dataset. Only `weather` or `temp_loggers` are currently allowed.

Details

Use this function to learn which summary modes and filters are allowed.

We are working on implementing summary visualisation methods for weather station data. So, for the moment, the options below are only available for temperature logger data. Three options are available:

- summary-by-seriesExpose summary for all available series; a series is a continuing time-series, i.e. a collection of deployments measuring the same parameter at the same site. For temperature loggers, series is synonymous with sub-site. For weather stations, it is the combination of sub-site and parameter.
- summary-by-deploymentExpose summary for all available deployments.
- dailyReturn mean daily aggregated monitoring data .

We offer a list of valid filter names:

- siteFilter by a particular site.
- subsiteFilter by a particular subsite.
- seriesFilter by a particular series.
- series_idA unique identifier for the series - it should be unique within a dataset. An alternative to looking up a series by name.
- parameterParameter of interest. Only relevant for weather station data because temperature logger is always water temperature.
- min_latMinimum latitude; used to filter by a lat-lon box.
- max_latMaximum latitude; used to filter by a lat-lon box.
- min_lonMinimum longitude; used to filter by a lat-lon box.
- max_lonMaximum longitude; used to filter by a lat-lon box.
- from_dateFilter from time (string of format YYYY-MM-DD).
- thru_dateFilter until time (string of format YYYY-MM-DD).

Some additional options for the actual download, which should be passed as additional arguments to the function, are:

- sizeSet a page size for large queries (only for the `data` and `data-no-key` endpoints).
- cursorUsed for pagination on / data").
- versionRequest the data as recorded at a particular time (a version history).

Value

A `list` of two `character` vectors: one detailing summary modes, another detailing filters.

Author(s)

AIMS Datacentre <adc@aims.gov.au>

Examples

```
## Not run:
library(dataaimsr)
aims_expose_attributes("weather")
aims_expose_attributes("temp_loggers")

## End(Not run)
```

`aims_filter_values` *Retrieve vector of existing filter values*

Description

This is a utility function which allows to user to query about the existing possibilities of a given filter name

Usage

```
aims_filter_values(target, filter_name)
```

Arguments

`target`	A `character` vector of length 1 specifying the dataset. Only `weather` or `temp_loggers` are currently allowed.
`filter_name`	A `character` string containing the name of the filter. Must be "site", "subsite", "series", or "parameter". See details.

Details

For a full description of each valid filter_name see ?`aims_expose_attributes`. In the temperature logger dataset, "subsite" is equivalent to "series"; moreover, note that there is only one parameter being measured (i.e. water temperature), so the "parameter" filter contains one single value.

Value

Either a `data.frame` if `filter_name = "series"`, else a `character` vector.

Author(s)

AIMS Datacentre <adc@aims.gov.au>

See Also

`aims_data`, `aims_expose_attributes`

Examples

```
## Not run:
library(dataaimsr)
aims_filter_values("weather", filter_name = "site")
aims_filter_values("temp_loggers", filter_name = "subsite")

## End(Not run)
```

aims_metadata *Extracts metadata attribute from object of class aimsdf*

Description

Extracts metadata attribute from object of class aimsdf

Usage

```
aims_metadata(df_)
```

Arguments

df_	A data.frame of class aimsdf created by function aims_data

Details

This function retrieves the metadata attribute from an aimsdf object. If the input aimsdf object is a summary data.frame (see ?aims_data), then output will be an empty string.

Value

A character vector.

Author(s)

AIMS Datacentre <adc@aims.gov.au>

aims_parameters *Extracts parameters attribute from object of class aimsdf*

Description

Extracts parameters attribute from object of class aimsdf

Usage

```
aims_parameters(df_)
```

Arguments

df_	A data.frame of class aimsdf created by function aims_data

Details

This function retrieves the parameters attribute from an aimsdf object. If the input aimsdf object is a summary data.frame (see ?aims_data), then output will be an empty string.

Value

A character vector.

Author(s)

AIMS Datacentre <adc@aims.gov.au>

---

is_aimsdf *Checks if argument is a* aimsdf *object*

Description

Checks if argument is a aimsdf object

Usage

```
is_aimsdf(x)
```

Arguments

x	An R object

---

next_page_data *Further data requests via the AIMS Data Platform API*

Description

Similar to page_data, but for cases #' where there are multiple URLs for data retrieval

Usage

```
next_page_data(url, api_key = NULL, ...)
```

Arguments

url	A data retrieval URL
api_key	An AIMS Data Platform API Key
...	Additional arguments to be passed to internal function update_format

Details

The AIMS Data Platform R Client provides easy access to data sets for R applications to the AIMS Data Platform API. The AIMS Data Platform requires an API Key for requests, which can be obtained at this link. It is preferred that API Keys are not stored in code. We recommend storing the environment variable `AIMS_DATAPLATFORM_API_KEY` permanently under the user's `.Renviron` file in order to load the API Key automatically.

There are two types of data currently available through the AIMS Data Platform API: Weather and Sea Water Temperature Loggers. They are searched internally via unique DOI identifiers. Only one data type at a time can be passed to the argument `target`.

A list of arguments for `filters` can be exposed for both Weather and Sea Water Temperature Loggers using function `aims_expose_attributes`.

Note that at present the user can inspect the range of dates for the temperature loggers data only (see usage of argument `summary` in the examples below). For that, the argument `summary` must be either the string `"summary-by-series"` or `"summary-by-deployment"`. In those cases, time filters will be ignored.

Details about available dates for each dataset and time series can be accessed via Metadata on AIMS Data Platform API. We raise this caveat here because these time boundaries are very important; data are collected at very small time intervals, a window of just a few days can yield very large datasets. The query will return and error if it reaches the system's memory capacity.

For that same reason, from version 1.1.0 onwards, we are offering the possibility of downloading a mean daily aggregated version. For that, the user must set `summary = "daily"`. In this particular case, query filter will be taken into account.

Value

`aims_data` returns a `data.frame` of class `aimsdf`.

If `summary %in% c("summary-by-series", "summary-by-deployment")`, the output shows the summary information for the target dataset (i.e. weather or temperature loggers) (NB: currently, `summary` only works for the temperature logger database). If `summary` is *not* passed as an additional argument, then the output contains **raw** monitoring data. If `summary = "daily"`, then the output contains **mean daily aggregated** monitoring data. The output also contains five attributes (empty strings if `summary` is passed as an additional argument):

- `metadata`a DOI link containing the metadata record for the data series.
- `citation`the citation information for the particular dataset.
- `parameters`The measured parameters comprised in the output.
- `type`The type of dataset. Either "monitoring" if `summary` is not specified, "monitoring (daily aggregation)" if `summary = "daily"`, or a "summary-by-" otherwise.
- `target`The input target.

Author(s)

AIMS Datacentre <adc@aims.gov.au>

See Also

aims_filter_values, page_data, aims_data

---

page_data *Request data via the AIMS Data Platform API*

Description

A function that communicates with the the AIMS Data Platform via the AIMS Data Platform API

Usage

```
page_data(
  doi,
  filters = NULL,
  api_key = NULL,
  summary = NA,
  aims_version = NA,
  verbose = FALSE
)
```

Arguments

doi	A Digital Object Identifier for a chosen AIMS data series
filters	A list containing a set of filters for the data query (see Details).
api_key	An AIMS Data Platform API Key
summary	Should summary tables ("summary-by-series" or "summary-by-deployment") or daily aggregated data ("daily") be returned instead of full data (see Details)?
aims_version	A character string defining the version of database. Must be "/v1.0" or "-v2.0". If none is provided, then "-v2.0" (the most recent) is used.
verbose	Should links be printed to screen? Used for debugging only

Details

The AIMS Data Platform R Client provides easy access to data sets for R applications to the AIMS Data Platform API. The AIMS Data Platform requires an API Key for requests, which can be obtained at this link. It is preferred that API Keys are not stored in code. We recommend storing the environment variable AIMS_DATAPLATFORM_API_KEY permanently under the user's .Renviron file in order to load the API Key automatically.

There are two types of data currently available through the AIMS Data Platform API: Weather and Sea Water Temperature Loggers. They are searched internally via unique DOI identifiers. Only one data type at a time can be passed to the argument target.

A list of arguments for filters can be exposed for both Weather and Sea Water Temperature Loggers using function aims_expose_attributes.

Note that at present the user can inspect the range of dates for the temperature loggers data only (see usage of argument `summary` in the examples below). For that, the argument `summary` must be either the string `"summary-by-series"` or `"summary-by-deployment"`. In those cases, time filters will be ignored.

Details about available dates for each dataset and time series can be accessed via Metadata on AIMS Data Platform API. We raise this caveat here because these time boundaries are very important; data are collected at very small time intervals, a window of just a few days can yield very large datasets. The query will return and error if it reaches the system's memory capacity.

For that same reason, from version 1.1.0 onwards, we are offering the possibility of downloading a mean daily aggregated version. For that, the user must set `summary = "daily"`. In this particular case, query filter will be taken into account.

Value

`aims_data` returns a `data.frame` of class `aimsdf`.

If `summary %in% c("summary-by-series", "summary-by-deployment")`, the output shows the summary information for the target dataset (i.e. weather or temperature loggers) (NB: currently, `summary` only works for the temperature logger database). If `summary` is *not* passed as an additional argument, then the output contains **raw** monitoring data. If `summary = "daily"`, then the output contains **mean daily aggregated** monitoring data. The output also contains five attributes (empty strings if `summary` is passed as an additional argument):

- `metadata`a DOI link containing the metadata record for the data series.
- `citation`the citation information for the particular dataset.
- `parameters`The measured parameters comprised in the output.
- `type`The type of dataset. Either "monitoring" if `summary` is not specified, "monitoring (daily aggregation)" if `summary = "daily"`, or a "summary-by-" otherwise.
- `target`The input target.

Author(s)

AIMS Datacentre <adc@aims.gov.au>

See Also

`aims_expose_attributes`, `aims_filter_values`, `aims_data`

`plot.aimsdf` *plot.aimsdf*

Description

Plotting options for aimsdf objects

Usage

```
## S3 method for class 'aimsdf'
plot(x, ..., ptype, pars)
```

Arguments

x	An object of class aimsdf as returned by aims_data.
...	Not used.
ptype	Type of plot. Can either be "time_series" or "map".
pars	Which parameters to plot? Only relevant if ptype is "time_series"

Details

Currently plots cannot be customised. Summary datasets can only be represented by maps.

Value

An object of class ggplot.

Examples

```
## Not run:
library(dataaimsr)
wdf <- aims_data("weather", api_key = NULL,
                 filters = list(site = "Yongala",
                                from_date = "2018-01-01",
                                thru_date = "2018-01-02"))
plot(wdf, ptype = "map")
plot(wdf, ptype = "time_series")
# summary-by- datasets can only return maps
sdf <- aims_data("temp_loggers", api_key = NULL,
                 summary = "summary-by-deployment")
plot(sdf, ptype = "map")

## End(Not run)
```

---

print.aimsdf *print.aimsdf*

Description

print.aimsdf

Usage

```
## S3 method for class 'aimsdf'
print(x, ...)
```

Arguments

x	An object of class aimsdf as returned by aims_data.
...	Not used.

Value

A list containing a summary of the model fit as returned a brmsfit for each model.

---

summary.aimsdf *summary.aimsdf*

Description

summary.aimsdf

Usage

```
## S3 method for class 'aimsdf'
summary(object, ...)
```

Arguments

object	An object of class aimsdf as returned by aims_data.
...	Unused.

Value

A list containing summary info from the input data.frame.

Index