Package: appeears (via r-universe)

July 20, 2024

Title Interface to 'AppEEARS' NASA Web Services

Version 1.1

Description Programmatic interface to the NASA Application for Extracting and Exploring Analysis Ready Samples services (AppEEARS; <https://appeears.earthdatacloud.nasa.gov/>). The package provides easy access to analysis ready earth observation data in R.

URL https://github.com/bluegreen-labs/appeears, https://bluegreen-labs.github.io/appeears/

BugReports https://github.com/bluegreen-labs/appeears/issues

Depends R (>= 4.0)

Imports httr, keyring, memoise, getPass, R6, sf, jsonlite, geojsonio, rstudioapi

License AGPL-3

ByteCompile true

RoxygenNote 7.2.3

Suggests rmarkdown, covr, testthat, terra, ncdf4, knitr, rlang, dplyr, ggplot2, patchwork

VignetteBuilder knitr

Repository https://bluegreen-labs.r-universe.dev

RemoteUrl https://github.com/bluegreen-labs/appeears

RemoteRef HEAD

RemoteSha c2385fed0855a74c7fe6a43ee6016def96ecdf99

Contents

`rs_build_task` *Build a task request*

Description

Builds a valid JSON formatted API request from either a tidy data frame with point based sub-tasks, or when a region of interest (roi) is specified coordinates (latitude/longitude) will be ignored and a bounding box for an sf or SpatRaster object will be used instead.

Usage

```
rs_build_task(df, roi, format = "geotiff")
```

Arguments

`df`	a data frame with task name (task), subtask name (subtask), latitude, longitude, start (start) and end (end) date.
`roi`	a region of interest defined by a SpatRaster or sf object, the roi will override any point based data provided as latittude-longitude coordinates in the data frame
`format`	file format of the downloaded data either geotiff (the default) or netcdf4

Value

a valid AppEEARS JSON formatted task

Examples

```
# define a task as a tidy data frame
# multiple subtasks can be provided
df <- data.frame(
task = "task_name",
subtask = c("sub_task"),
latitude = c(36.206228),
longitude = c(-112.127134),
start = c("2018-01-01"),
end = c("2018-01-15"),
product = c("MCD12Q2.006"),
```

```
layer = c("Greenup")
)

# build a task
rs_build_task(df)
```

`rs_bundle` *AppEEARS list of bundled files to download*

Description

Returns a data frame of all data ready for download as one bundle

Usage

```
rs_bundle(task_id, user)
```

Arguments

`task_id`	task id for which to return the file download bundle
`user`	username used to sign up

Value

returns a nested list of files to download

Author(s)

Koen Hufkens

Examples

```
## Not run:
# get a list of datasets
rs_bundle(
user = "your_user_name",
task_id = "a_task_id"
)

## End(Not run)
```

`rs_delete` *Delete AppEEARS task from queue*

Description

Removes a task from the queue and or buffer

Usage

```
rs_delete(task_id, user, purge = FALSE)
```

Arguments

`task_id`	AppEEARS task id
`user`	username used to sign up
`purge`	if TRUE, remove all previously finished tasks from the task list (default = FALSE)

Value

returns the content of the API call

Author(s)

Koen Hufkens

Examples

```
## Not run:
# delete a single task
rs_delete(
  user = "your_user_name",
  task_id = "a_task_id"
)

# delete all finished or crashed
# jobs (if not deleted previously)
rs_delete(
  user = "your_user_name",
  purge = TRUE
)

## End(Not run)
```

rs_get_key *Retrieve NASA Earth Data password*

Description

Returns you token set by rs_set_key

Usage

```
rs_get_key(user)
```

Arguments

user	username used to sign up

Value

the password set using rs_set_key saved in the keychain

Author(s)

Koen Hufkens

See Also

rs_set_key

Examples

```
## Not run:
# set key
rs_set_key(user = "test@mail.com", password = "123")

# get key
rs_get_key(user = "test@mail.com")

## End(Not run)
```

rs_layers *AppEEARS dataset layers*

Description

Returns a data frame of available layers for an AppEEARS product

Usage

```
rs_layers(product)
```

Arguments

product	product for which to list the layers

Value

returns a data frame with the AppEEARS datasets

Author(s)

Koen Hufkens

See Also

rs_products

Examples

```
# is the server reachable
server_check <- appeears:::rs_running(
 file.path(appeears:::rs_server(),"product")
)

# get a list of datasets
if(server_check){
 layers <- rs_layers("MCD43A4.006")
 print(layers$Layer)
}
```

rs_list_task *AppEEARS list of tasks and status*

Description

Returns a data frame of all submitted tasks either in full of when providing the di

Usage

```
rs_list_task(task_id, user)
```

Arguments

task_id	task for which to list the status (if missing all tasks are listed)
user	username used to sign up

Value

returns a data frame with the AppEEARS tasks

Author(s)

Koen Hufkens

Examples

```
## Not run:
# get a list of datasets
rs_list_task()

## End(Not run)
```

rs_login *Checks AppEEARS login*

Description

Returns a valid token for a session if successful otherwise fails with an error (stop())

Usage

```
rs_login(user)
```

Arguments

user	AppEEARS username

Value

returns an AppEEARS session (bearer) token

---

rs_logout *Invalidates an AppEEARS bearer token*

Description

Given a token it will log out / delete this token, invalidating it.

Usage

```
rs_logout(token)
```

Arguments

token	a Bearer token as returned by rs_login()

Value

returns if the session has closed TRUE/FALSE

---

rs_products *AppEEARS dataset list*

Description

Returns a data frame of available data products

Usage

```
rs_products()
```

Value

returns a data frame with the AppEEARS datasets

Author(s)

Koen Hufkens

See Also

rs_set_key rs_transfer rs_request

Examples

```
# is the server reachable
server_check <- appeears:::rs_running(
 file.path(appeears:::rs_server(),"product")
)

# get a list of datasets
if(server_check){
 products <- rs_products()
}
```

`rs_quality` *List or translate AppEEARS quality metrics*

Description

Returns a data frame of all quality layers, or the translation of a quality layer value into plain language.

Usage

```
rs_quality(product, layer, value)
```

Arguments

`product`	AppEEARS product name
`layer`	name of a product quality control layer
`value`	quality control value to translate

Value

returns a data frame of all AppEEARS quality layers, or those associated with a product. When a value is provided this quality flag will be translated from bitwise representation to plain language.

Author(s)

Koen Hufkens

Examples

```
## Not run:
# get a list of quality layers for all data products
rs_quality()

## End(Not run)
```

`rs_request` *AppEEARS data request and download*

Description

Stage a data request, and optionally download the data to disk. Alternatively you can only stage requests, logging the request URLs to submit download queries later on using `rs_transfer`.

Usage

```
rs_request(
  request,
  user,
  transfer = TRUE,
  path = tempdir(),
  time_out = 3600,
  job_name,
  verbose = TRUE
)

rs_request_batch(
  request_list,
  workers = 2,
  user,
  path = tempdir(),
  time_out = 7200,
  total_timeout = length(request_list) * time_out/workers,
  verbose = TRUE
)
```

Arguments

`request`	nested list with query parameters following the layout as specified on the AppEEARS APIs page
`user`	user (email address or ID) provided by the AppEEARS data service, used to retrieve the token set by `rs_set_key`
`transfer`	logical, download data TRUE or FALSE (default = TRUE)
`path`	path were to store the downloaded data
`time_out`	individual time out for each request
`job_name`	optional name to use as an RStudio job and as output variable name. It has to be a syntactically valid name.
`verbose`	show feedback on processing
`request_list`	a list of requests that will be processed in parallel.
`workers`	maximum number of simultaneous request that will be submitted to the service. Most services are limited to ~20 concurrent requests (default = 2).

`total_timeout` overall timeout limit for all the requests in seconds. (note that the overall timeout on a session is 48h)

Value

the path of the downloaded (requested file) or the an R6 object with download/transfer information

Author(s)

Koen Hufkens

See Also

`rs_set_key` `rs_transfer`

Examples

```
## Not run:
# specifiy a task/request as a
# data frame
df <- data.frame(
 task = "grand canyon",
 subtask = c("test1", "test2"),
 latitude = c(36.206228, 36.206228),
 longitude = c(-112.127134, -112.127134),
 start = c("2018-01-01","2018-01-01"),
 end = c("2018-01-15","2018-01-15"),
 product = c("MOD11A2.061","MCD12Q2.006"),
 layer = c("LST_Day_1km","Dormancy")
)

# build a proper JSON query
task <- rs_build_task(df = df)

# request the task to be executed
rs_request(
 request = task,
 user = "earth_data_user",
 transfer = TRUE,
 path = "~/some_path",
 verbose = TRUE
)

## End(Not run)
```

rs_set_key *Set NASA Earth Data password*

Description

Saves the token to your local keychain under a service called "appeears".

Usage

```
rs_set_key(user, password)
```

Arguments

user	user used to sign up for the AppEEARS data service (this is not the email address, but the true user name!)
password	used to sign up for AppEEARS

Details

In systems without keychain management set the option keyring_backend to ‘file‘ (i.e. options(keyring_backend = "file")) in order to write the keychain entry to an encrypted file. This mostly pertains to headless Linux systems. The keychain files can be found in ~/.config/r-keyring.

Value

It invisibly returns the user.

Author(s)

Koen Hufkens

See Also

rs_get_key

Examples

```
## Not run:
# set key
rs_set_key(user = "test", password = "123")

# get key
rs_get_key(user = "test")

# leave user and key empty to open a browser window to the service's website
# and type the key interactively
rs_get_key()


## End(Not run)
```

rs_transfer *AppEEARS data transfer function*

Description

Returns the contents of the requested url as a NetCDF file downloaded to disk or the current status of the requested transfer.

Usage

```
rs_transfer(task_id, user, path = tempdir(), verbose = TRUE)
```

Arguments

task_id	R6 rs_request) query output or task id
user	user (email address) used to sign up for the AppEEARS data service, used to retrieve the token set by rs_set_key.
path	path were to store the downloaded data
verbose	show feedback on data transfers

Value

data on disk as specified by a rs_request

Author(s)

Koen Hufkens

See Also

rs_set_key rs_request

Examples

```
## Not run:
# set key
rs_set_key(user = "test", password = "123")

# request data and grab url and try a transfer
r <- rs_request(request, "test", transfer = FALSE)

# check transfer, will download if available
rs_transfer(r$get_task_id(), user = "test")

## End(Not run)
```

Index