Package 'ecmwfr'

July 22, 2025

Title Interface to 'ECMWF' and 'CDS' Data Web Services

Version 2.0.3

Description Programmatic interface to the European Centre for Medium-Range Weather Forecasts dataset web services (ECMWF; <https://www.ecmwf.int/>) and Copernicus's Data Stores. Allows for easy downloads of weather forecasts and climate reanalysis data in R. Data stores covered include the Climate Data Store (CDS; <https://cds.climate.copernicus.eu>), Atmosphere Data Store (ADS; <https://ads.atmosphere.copernicus.eu>) and Early Warning Data Store (CEMS; <https://ewds.climate.copernicus.eu>).

URL https://github.com/bluegreen-labs/ecmwfr

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

Depends R (>= 4.2)

Imports httr, memoise, getPass, R6, keyring, tools

License AGPL-3

ByteCompile true

RoxygenNote 7.3.1

Suggests rmarkdown, covr, xml2, testthat, terra, maps, ncdf4, knitr, rlang, rstudioapi, jsonlite

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Koen Hufkens [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-5070-8109>), Reto Stauffer [ctb] (ORCID: <https://orcid.org/0000-0002-3798-5507>), Elio Campitelli [ctb] (ORCID: <https://orcid.org/0000-0002-7742-9230>), BlueGreen Labs [fnd]

Maintainer Koen Hufkens <koen.hufkens@gmail.com>

Repository CRAN

Date/Publication 2025-02-10 12:00:02 UTC

Contents

`print.ecmwfr_archetype` *Methods to deal with visualizing / printing requesting info from the archetype constructor*

Description

Methods to deal with visualizing / printing requesting info from the archetype constructor

Usage

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

Arguments

`x`	archetype object
`...`	additional parameters to pass on

`wf_archetype` *Creates an archetype function*

Description

Creates a universal MARS / CDS formatting function, in ways similar to `wf_modify_request()` but the added advantage that you could code for the use of dynamic changes in the parameters provided to the resulting custom function.

Usage

```
wf_archetype(request, dynamic_fields)
```

Arguments

request	a MARS or CDS request as an R list object.
dynamic_fields	character vector of fields that could be changed.

Details

Contrary to a simple replacement as in wf_modify_request() the generated functions are considered custom user written. Given the potential for complex formulations and formatting commands NO SUPPORT for the resulting functions can be provided. Only the generation of a valid function will be guaranteed and tested for.

Value

a function that takes ‘dynamic_fields‘ as arguments and returns a request as an R list object.

Examples

```
## Not run:

ERA <- wf_archetype(
  request = list(
    dataset_short_name = "reanalysis-era5-pressure-levels",
    product_type = "reanalysis",
    variable = "geopotential",
    year = "2024",
    month = "03",
    day = "01",
    time = "13:00",
    pressure_level = "1000",
    data_format = "grib",
    target = "download.grib"
  ),
    dynamic_fields = c("year", "day", "target")
  )
# print output of the function with below (new) parameters
str(ERA(2021, 3, "new_download.grip"))


## End(Not run)
```

---

wf_check_request *check ECMWF / CDS data requests*

Description

Check the validity of a data request by comparing the main dataset to the list provided by wf_datasets

Usage

```
wf_check_request(request)
```

Arguments

request	nested list with query parameters following the layout as specified on the ECMWF API page

Value

a data frame with the determined service and url service endpoint

Author(s)

Koen Hufkens

See Also

wf_set_key wf_transfer,wf_request, wf_transfer

---

wf_datasets *List ECMWF Data Store dataset*

Description

Returns a list of all ECMWF datasets, covering all Data Store services (i.e. CDS, ADS, CEMS). This function is used to validate the datasets queried by wf_request. For optimization reasons and limit API calls the function is cached and only called once per session (assuming that available products and their information and endpoints aren't updated on a regular sub-daily basis).

Usage

```
wf_datasets(service = c("cds", "ads", "cems"), simplify = TRUE)
```

Arguments

service	which service to use, one of webapi, cds or ads (default = webapi)
simplify	simplify the output, logical (default = TRUE). When not simplified the raw API return is provided as a nested list, for debugging purposes mostly.

Value

returns a data frame with the ECMWF Data Store datasets

Author(s)

Koen Hufkens

See Also

wf_transfer wf_request

Examples

```
## Not run:
# get a list of ECMWF Data Store datasets
wf_datasets()

## End(Not run)
```

---

wf_dataset_info *List ECMWF Data Store dataset information*

Description

Shows and returns detailed product information about a specific data set (see wf_datasets). This includes the list of sub-products in the collection as well as date and time ranges.

Usage

```
wf_dataset_info(dataset, simplify = TRUE)
```

Arguments

dataset	character, name of the data set for which the product information should be loaded
simplify	boolean, default TRUE. If TRUE the description will be returned as tidy data instead of a nested list.

Value

Downloads a tidy data frame with product descriptions from CDS. If simplify = FALSE a list with product details will be returned.

Author(s)

Reto Stauffer, Koen Hufkens

See Also

wf_datasets.

Examples

```
## Not run:
 # Return information
 info <- wf_dataset_info("reanalysis-era5-single-levels")
 names(info)

## End(Not run)
```

`wf_delete` *Delete ECMWF Data Store request*

Description

Deletes a staged download from the queue when not using R6 methods.

Usage

```
wf_delete(url, user = "ecmwfr", verbose = TRUE)
```

Arguments

`url`	url to query
`user`	user, generally not set (default = "ecmwfr"), used by `wf_set_key`
`verbose`	show feedback on processing

Author(s)

Koen Hufkens

See Also

`wf_set_key` `wf_transfer` `wf_request`

Examples

```
## Not run:

# demo query using a valid request (not shown)
file <- wf_request(request = request)

# delete request
job_url <- file$get_url()
wf_delete(url = job_url)

## End(Not run)
```

wf_get_key *Get secret ECMWF / CDS token*

Description

Returns you token set by wf_set_key

Usage

```
wf_get_key(user = "ecmwfr")
```

Arguments

user	user (email address) used to sign up for the ECMWF data service

Value

the key set using wf_set_key saved in the keychain

Author(s)

Koen Hufkens

See Also

wf_set_key

Examples

```
## Not run:
# set key
wf_set_key(key = "123")

# get key
wf_get_key()

## End(Not run)
```

wf_request *ECMWF Data Store (DS) 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 wf_transfer. Note that the function will do some basic checks on the request input to identify possible problems.

Usage

```
wf_request(
  request,
  user = "ecmwfr",
  transfer = TRUE,
  path = tempdir(),
  time_out = 3600,
  retry = 30,
  job_name,
  verbose = TRUE
)

wf_request_batch(
  request_list,
  workers = 2,
  user = "ecmwfr",
  path = tempdir(),
  time_out = 3600,
  retry = 5,
  total_timeout = length(request_list) * time_out/workers
)
```

Arguments

request	nested list with query parameters following the layout as specified on the ECMWF APIs page
user	user (default = "ecmwf") provided by the ECMWF data service, used to retrieve the token set by wf_set_key
transfer	logical, download data TRUE or FALSE (default = TRUE)
path	path were to store the downloaded data
time_out	how long to wait on a download to start (default = 3600 seconds).
retry	polling frequency of submitted request for downloading (default = 30 seconds).
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 ECMWF services are limited to 20 concurrent requests (default = 2).
total_timeout	overall timeout limit for all the requests in seconds.

Value

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

Author(s)

Koen Hufkens

See Also

wf_set_key wf_transfer

Examples

```
## Not run:
# set key
wf_set_key(key = "123")

request <- list(
  dataset_short_name = "reanalysis-era5-pressure-levels",
  product_type = "reanalysis",
  variable = "geopotential",
  year = "2024",
  month = "03",
  day = "01",
  time = "13:00",
  pressure_level = "1000",
  data_format = "grib",
  target = "download.grib"
)

# demo query
wf_request(request = request)

# Run as an RStudio Job. When finished, will create a
# variable named "test" in your environment with the path to
# the downloaded file.
wf_request(request = request, job_name = "test")

## End(Not run)
```

wf_set_key *Set secret ECMWF token*

Description

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

Usage

```
wf_set_key(key, user = "ecmwfr")
```

Arguments

key	token provided by ECMWF
user	user (email address) used to sign up for the ECMWF data service, if only a single user is needed it defaults to ("ecmwfr").

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

wf_get_key

Examples

```
## Not run:
# set key
wf_set_key(key = "123")

# get key
wf_get_key()

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


## End(Not run)
```

wf_transfer *ECMWF 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

```
wf_transfer(
  url,
  user = "ecmwfr",
  path = tempdir(),
  filename = tempfile("ecmwfr_", tmpdir = ""),
  verbose = TRUE
)
```

Arguments

url	R6 wf_request) query output or API endpoint
user	user (email address) used to sign up for the ECMWF data service, used to retrieve the token set by wf_set_key.
path	path were to store the downloaded data
filename	filename to use for the downloaded data
verbose	show feedback on data transfers

Details

Normal workflows would use the methods included in returned objects. This is for legacy support and custom scripting only.

Value

a (netCDF) file of data on disk as specified by a wf_request

Author(s)

Koen Hufkens

See Also

wf_set_key wf_request

Examples

```
## Not run:
# request data and grab url and try a transfer
# (request not provided)
r <- wf_request(request, transfer = FALSE)

# check transfer, will download if available
wf_transfer(r$get_url())

## End(Not run)
```

Index