Package: CopernicusClimate (via r-universe)

October 11, 2026

Title Search Download and Handle Data from Copernicus Climate Data Service

Version 0.0.6.0001

Description Subset and download data from EU Copernicus Climate Data Service: <https://cds.climate.copernicus.eu/>. Import information about the Earth's past, present and future climate from Copernicus into R without the need of external software.

Depends R (>= 4.1.0)

Imports dplyr, httr2, jsonlite, methods, purrr, rlang, stringr, tibble, tidyr

Suggests clipr, ggplot2, knitr, RefManageR, rmarkdown, sf, stars, testthat (>= 3.0.0)

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Collate 'CopernicusClimate-package.R' 'helpers.R' 'catalogue.R' 'cite.R' 'jobs.R' 'licenses.R' 'python.R' 'retrieve.R' 'stars.R' 'token.R'

Config/testthat/edition 3

Language en-GB

URL https://pepijn-devries.github.io/CopernicusClimate/, https://github.com/pepijn-devries/CopernicusClimate/

BugReports https://github.com/pepijn-devries/CopernicusClimate/issues

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libicu-dev libssl-dev

Repository https://pepijn-devries.r-universe.dev

Date/Publication 2026-08-12 22:37:34 UTC

RemoteUrl https://github.com/pepijn-devries/CopernicusClimate

RemoteRef main

RemoteSha 7e72c93919abc2586dba91d5c2b4983619a4b7ad

Contents

cds_accept_licence . . . 2
cds_accepted_licences . . . 3
cds_account_metrics . . . 4
cds_build_request . . . 4
cds_catalogue_vocabulary . . . 5
cds_check_authentication . . . 6
cds_cite_dataset . . . 7
cds_dataset_form . . . 7
cds_delete_job . . . 8
cds_download_jobs . . . 9
cds_estimate_costs . . . 10
cds_get_account . . . 11
cds_get_token . . . 12
cds_job_results . . . 13
cds_list_datasets . . . 14
cds_list_jobs . . . 15
cds_list_licences . . . 16
cds_python_to_r . . . 16
cds_starred . . . 18
cds_submit_job . . . 19

Index **21**

`cds_accept_licence` *Accept a specific licence*

Description

Call this function if you wish to accept a specific license.

Usage

```
cds_accept_licence(license, revision, ..., token = cds_get_token())
```

Arguments

`license`	The license id you wish to accept.
`revision`	The revision number of the license you are accepting. Should always be the latest revision.
`...`	Ignored
`token`	An API key to be used for authentication. Will use `cds_get_token()` by default.

Value

Returns a `data.frame` containing the accepted license

See Also

Other licences: cds_accepted_licences(), cds_list_licences()

Examples

```
if (interactive() && cds_token_works()) {
  cds_accept_licence("cc-by", 1)
}
```

---

cds_accepted_licences *List accepted licences*

Description

In order to use specific features of the Climate Data Service, you may first need to accept specific licences. This function will list all licenses you have accepted.

Usage

```
cds_accepted_licences(
  scope = c("all", "dataset", "portal"),
  ...,
  token = cds_get_token()
)
```

Arguments

scope	Scope of the licenses to be listed should be one of "all" (default), "dataset" or "portal".
...	Ignored
token	An API key to be used for authentication. Will use cds_get_token() by default.

Value

Returns a data.frame listing the accepted licenses.

See Also

Other licences: cds_accept_licence(), cds_list_licences()

Examples

```
if (interactive() && cds_token_works()) {
  cds_accepted_licences("portal")
}
```

cds_account_metrics *Get Prometheus metrics for account*

Description

Obtain account metrics that can be interpreted with Prometheus

Usage

```
cds_account_metrics(token = cds_get_token(), ...)
```

Arguments

token	An API key to be used for authentication. Will use cds_get_token() by default.
...	Ignored

Value

Returns text that can be interpreted with Prometheus

See Also

Other authentication-functions: cds_check_authentication(), cds_get_account(), cds_get_token()

Examples

```
if (interactive() && cds_token_works()) {
  cds_account_metrics()
}
```

cds_build_request *Prepare a request for downloading a dataset*

Description

This function is used by cds_estimate_costs() and cds_submit_job() to subset a dataset before downloading. It will also help you to explore which parameters are available for subsetting.

Usage

```
cds_build_request(dataset, ...)
```

Arguments

dataset	The dataset name to be used for setting up a request.
...	Parameters for subsetting the dataset. Use cds_dataset_form() to inquiry which parameters and parameter values are available for a specific dataset. If left blank it will take default parameter values.

Value

Returns a named list, which can be used to submit a job (cds_submit_job()) or inquiry its cost (cds_estimate_costs()).

See Also

Other helper-functions: cds_cite_dataset(), cds_dataset_form(), cds_estimate_costs(), cds_python_to_r()

Examples

```
if (interactive()) {
  cds_build_request(
    dataset        = "reanalysis-era5-pressure-levels",
    variable       = "geopotential",
    product_type   = "reanalysis",
    area           = c(n = 55, w = -1, s = 50, e = 10),
    year           = "2024",
    month          = "03",
    day            = "01",
    pressure_level = "1000",
    data_format    = "netcdf"
  )
}
```

---

cds_catalogue_vocabulary

List catalogue vocabulary

Description

The catalogue uses a specific vocabulary for keywords. This function produces an overview.

Usage

```
cds_catalogue_vocabulary(...)
```

Arguments

...	Ignored

Value

Returns a `data.frame` of keyword vocabulary used by the catalogue.

See Also

Other exploration-functions: `cds_list_datasets()`, `cds_starred()`

Examples

```
if (interactive()) {
  cds_catalogue_vocabulary()
}
```

`cds_check_authentication`

Check if authentication works with a specific token

Description

Checks if the specified API key can be used for authentication at the Climate Data Service.

Usage

```
cds_check_authentication(token = cds_get_token(), ...)

cds_token_works(token = cds_get_token(), ...)
```

Arguments

`token`	An API key to be used for authentication. Will use `cds_get_token()` by default.
`...`	Ignored

Value

`cds_check_authentication()` will return some account information when successful but throws an error if it is not. In contrast `cds_token_works()` returns a `logical` value and will not throw an error upon failure.

See Also

Other authentication-functions: `cds_account_metrics()`, `cds_get_account()`, `cds_get_token()`

Examples

```
if (interactive() && cds_token_works()) {
  cds_check_authentication()
}
```

cds_cite_dataset *Cite a dataset*

Description

Use this function to obtain citation details for a specific dataset

Usage

```
cds_cite_dataset(dataset, ...)
```

Arguments

dataset	The name of a dataset to be cited.
...	Ignored

Value

Returns a BibEntry-class object, with citation details for the requested dataset

See Also

Other helper-functions: cds_build_request(), cds_dataset_form(), cds_estimate_costs(), cds_python_to_r()

Examples

```
if (interactive()) {
  cds_cite_dataset("reanalysis-era5-pressure-levels")
}
```

cds_dataset_form *Obtain an overview of options to subset a dataset*

Description

This function provides an overview of parameters that can be used to subset a dataset. It can help you set up a request with cds_submit_job() or cds_build_request().

Usage

```
cds_dataset_form(dataset, ...)
```

Arguments

dataset	A name of a dataset to explore
...	Ignored

Value

A data.frame with aspects of the dataset that can be subsetted, when defining a job. See also cds_submit_job() and cds_build_request()

See Also

Other helper-functions: cds_build_request(), cds_cite_dataset(), cds_estimate_costs(), cds_python_to_r()

Examples

```
if (interactive() && cds_token_works()) {
  cds_dataset_form("reanalysis-era5-pressure-levels")
}
```

---

cds_delete_job *Delete/cancel jobs submitted to the Climate Data Service*

Description

When you regret submitting a job, you can cancel it by calling this function.

Usage

```
cds_delete_job(job_id, ..., token = cds_get_token())
```

Arguments

job_id	Hexadecimal code used as identifier of a job. Identifies the job to be cancelled.
...	Ignored
token	An API key to be used for authentication. Will use cds_get_token() by default.

Value

Returns a data.frame with information about the cancelled job.

See Also

Other job-functions: cds_download_jobs(), cds_job_results(), cds_list_jobs(), cds_submit_job()

Examples

```
if (interactive() && cds_token_works()) {
  job <- cds_submit_job(
      dataset        = "reanalysis-era5-pressure-levels",
      variable       = "geopotential",
      product_type   = "reanalysis",
      area           = c(n = 55, w = -1, s = 50, e = 10),
      year           = "2024",
      month          = "03",
      day            = "01",
      pressure_level = "1000",
      data_format    = "netcdf"
    )
  cds_delete_job(job$jobID, tempdir())
}
```

`cds_download_jobs` *Download specific jobs*

Description

After submitting one or more jobs with `cds_submit_job()`, you can download the resulting files with `cds_download_jobs()`. See `vignette("download")` for more details.

Usage

```
cds_download_jobs(job_id, destination, names, ..., token = cds_get_token())
```

Arguments

`job_id`	If a specific job identifier is listed here, only the files resulting from those jobs are downloaded. If left blank, all successful jobs are downloaded.
`destination`	Destination path to store downloaded files.
`names`	File names for the downloaded files. If missing, the cryptic hexadecimal file name is taken from the job.
`...`	Ignored
`token`	An API key to be used for authentication. Will use `cds_get_token()` by default.

Value

A `data.frame` of all downloaded files. Contains a column `local` with the path to the locally stored files.

See Also

Other job-functions: `cds_delete_job()`, `cds_job_results()`, `cds_list_jobs()`, `cds_submit_job()`

Examples

```
if (interactive() && cds_token_works()) {
  job <- cds_submit_job(
      dataset        = "reanalysis-era5-pressure-levels",
      variable       = "geopotential",
      product_type   = "reanalysis",
      area           = c(n = 55, w = -1, s = 50, e = 10),
      year           = "2024",
      month          = "03",
      day            = "01",
      pressure_level = "1000",
      data_format    = "netcdf"
    )
  cds_download_jobs(job$jobID, tempdir())
}
```

---

cds_estimate_costs *Check the cost of a request against your quota*

Description

Each account has a limit to the amount of data that can be downloaded. Use this function to check if a request exceeds your quota.

Usage

```
cds_estimate_costs(dataset, ..., token = cds_get_token())
```

Arguments

dataset	A dataset name to be inspected
...	Parameters passed on to cds_build_request()
token	An API key to be used for authentication. Will use cds_get_token() by default.

Value

Returns a named list indicating the available quota and the estimated cost for a request specified with ...-arguments.

See Also

Other helper-functions: cds_build_request(), cds_cite_dataset(), cds_dataset_form(), cds_python_to_r()

Examples

```
if (interactive() && cds_token_works()) {
  cds_estimate_costs(
    dataset        = "reanalysis-era5-pressure-levels",
    variable       = "geopotential",
    product_type   = "reanalysis",
    area           = c(n = 55, w = -1, s = 50, e = 10),
    year           = "2024",
    month          = "03",
    day            = "01",
    pressure_level = "1000",
    data_format    = "netcdf"
  )

  cds_estimate_costs(dataset = "reanalysis-era5-pressure-levels")
}
```

---

cds_get_account *Get account details*

Description

Retrieve account details associated with the provided `token`

Usage

```
cds_get_account(token = cds_get_token(), ...)
```

Arguments

`token`	An API key to be used for authentication. Will use `cds_get_token()` by default.
`...`	Ignored

Value

Returns a named `list` with account details.

See Also

Other authentication-functions: `cds_account_metrics()`, `cds_check_authentication()`, `cds_get_token()`

Examples

```
if (interactive() && cds_token_works()) {
  cds_get_account()
}
```

cds_get_token *Get or set a Climate Data Service API key*

Description

Many of the Climate Data Services features require a personal Application Programming Interface (API) key. This function will get a token that was previously stored (`cds_set_token()`), and can be used throughout the R session.

Usage

```
cds_get_token(...)

cds_set_token(token, method = c("option", "sysenv"), ...)
```

Arguments

`...`	Ignored
`token`	The API key you wish to set as an R option or to an environment variable.
`method`	Method to store the API key. Should be either `"option"` (default) or `"sysenv"`.

Details

To use an API key, you first need to get one from the Climate Data Service. You can do so, by creating an account and then initialise the key at https://cds.climate.copernicus.eu/profile.

There are different locations where the key can be stored and where `cds_get_token()` will look. It will look for the first successful value of (in order): the environment variable named `"CDSAPI_KEY"`, the R `getOption()` named `"CDSAPI_KEY"`, the environment variable named `"ECMWF_DATASTORES_KEY"`, and the R `getOption()` named `"ECMWF_DATASTORES_KEY"`.

You can set the key at the start of each R session with `cds_set_token()`. If you want a persistent solution, you can add the environment variable (with names shown above) to your system. Or you can add the option (with the names shown above) to your `".profile"` file. This will help you obscure your sensitive account information in your R script.

Value

`cds_get_token()` will return an API key token if it has been set. If it is not set it will return an empty string: `""`. `cds_set_token()` will return `NULL` invisibly.

See Also

Other authentication-functions: `cds_account_metrics()`, `cds_check_authentication()`, `cds_get_account()`

Examples

```
if (interactive() && !cds_token_works()) {
  cds_set_token("this-is-a-dummy-token", "option")
}

if (interactive()) {
  cds_get_token()
}
```

`cds_job_results` *Get the results for a submitted job*

Description

When a job is completed, you can get its results (i.e. download link) by calling this function.

Usage

```
cds_job_results(job_id, ..., token = cds_get_token())
```

Arguments

`job_id`	Hexadecimal code used as identifier of a job. Identifies the job for which to obtain the results.
`...`	Ignored
`token`	An API key to be used for authentication. Will use `cds_get_token()` by default.

Value

Returns a `data.frame` with information about the requested job.

See Also

Other job-functions: `cds_delete_job()`, `cds_download_jobs()`, `cds_list_jobs()`, `cds_submit_job()`

Examples

```
if (interactive() && cds_token_works()) {
  job <- cds_submit_job(
      dataset        = "reanalysis-era5-pressure-levels",
      variable       = "geopotential",
      product_type   = "reanalysis",
      area           = c(n = 55, w = -1, s = 50, e = 10),
      year           = "2024",
      month          = "03",
      day            = "01",
      pressure_level = "1000",
```

```
    data_format    = "netcdf"
  )
cds_job_results(job$jobID)
}
```

`cds_list_datasets` *List or search Climate Data Service datasets*

Description

This will help you decide which datasets you wish to obtain.

Usage

```
cds_list_datasets(dataset, ...)

cds_search_datasets(search, keywords, page = 0, limit = 50, ...)
```

Arguments

`dataset`	A specific dataset for which to list details. If missing all datasets are listed.
`...`	Ignored
`search`	A string containing free text search terms to look for in the available datasets.
`keywords`	A (vector of) string containing specific keywords. Should be listed in `cds_catalogue_vocabulary()`
`page`	When there are more search results than `limit`, results are paginated. Use `page` to specify which page to return starting at `0` (default).
`limit`	Use to limit the number of search results. Defaults to `50`.

Value

A `data.frame` listing the datasets from the Climate Data Service. In case of `cds_search_datasets()`, attributes named `SearchMeta` are added containing the number of matching and number of returned datasets.

See Also

Other exploration-functions: `cds_catalogue_vocabulary()`, `cds_starred()`

Examples

```
if (interactive()) {
  cds_list_datasets()
  cds_search_datasets("rain", "Temporal coverage: Future")
}
```

cds_list_jobs *List jobs submitted to the Climate Data Service*

Description

Once submitted with `cds_submit_job()` you can check the status of the job with this function. You can list all available jobs, or specific jobs.

Usage

```
cds_list_jobs(
  job_id = NULL,
  status = NULL,
  limit = 50,
  ...,
  token = cds_get_token()
)
```

Arguments

`job_id`	The id of a specific job, if you want the results for that job. If `NULL` (default) it is ignored.
`status`	Only return jobs with the status stated by this argument. Default is `NULL` meaning that jobs with any status are returned. Should be any of `"accepted"`, `"running"`, `"successful"`, `"failed"`, or `"rejected"`.
`limit`	Use to limit the number of listed results. Defaults to `50`.
`...`	Ignored
`token`	An API key to be used for authentication. Will use `cds_get_token()` by default.

Value

Returns a `data.frame` of submitted jobs.

See Also

Other job-functions: `cds_delete_job()`, `cds_download_jobs()`, `cds_job_results()`, `cds_submit_job()`

Examples

```
if (interactive() && cds_token_works()) {
  cds_list_jobs()
}
```

cds_list_licences *Obtain a list of licences that can be accepted*

Description

Some datasets require you to accept specific licences. This function will provide an overview of all licences associated with datasets and can be accepted.

Usage

```
cds_list_licences(...)
```

Arguments

...	Ignored

Value

Returns a `data.frame` of available licences that can be accepted.

See Also

Other licences: `cds_accept_licence()`, `cds_accepted_licences()`

Examples

```
if (interactive()) {
  cds_list_licences()
}
```

cds_python_to_r *Translate Python Climate Data Store request to R request*

Description

When looking for a dataset on https://cds.climate.copernicus.eu/datasets, you have the option to copy the API request code to the clipboard. However, this is Python code and cannot be used directly in this package. Use this function to translate the code to a request that can be handled by the package. For more details see `vignette("translate")`

Usage

```
cds_python_to_r(text, ...)
```

Arguments

text	A character string containing the Python code copied from a dataset download website https://cds.climate.copernicus.eu/datasets. When missing, the function will use any text on the system clipboard. This means that you can copy the API request code from the website to the clipboard, then call this function without arguments.
...	Ignored

Value

A named list that can be used as input for the functions cds_submit_job() and cds_estimate_costs()

See Also

Other helper-functions: cds_build_request(), cds_cite_dataset(), cds_dataset_form(), cds_estimate_costs()

Examples

```
python_code <-
"import cdsapi
dataset = \"reanalysis-era5-land\"
request = {
  \"variable\": [\"2m_dewpoint_temperature\"],
  \"year\": \"2025\",
  \"month\": \"01\",
  \"day\": [\"01\"],
  \"time\": [
    \"00:00\", \"01:00\", \"02:00\",
    \"03:00\", \"04:00\", \"05:00\",
    \"06:00\", \"07:00\", \"08:00\",
    \"09:00\", \"10:00\", \"11:00\",
    \"12:00\", \"13:00\", \"14:00\",
    \"15:00\", \"16:00\", \"17:00\",
    \"18:00\", \"19:00\", \"20:00\",
    \"21:00\", \"22:00\", \"23:00\"
  ],
  \"data_format\": \"netcdf\",
  \"download_format\": \"unarchived\"
}

client = cdsapi.Client()
client.retrieve(dataset, request).download()
"

cds_python_to_r(python_code)
```

cds_starred *Add or remove a star to a dataset, or list starred datasets*

Description

Use stars to keep track of your favourite datasets. Use these functions to add or remove a star.

Usage

```
cds_starred(..., token = cds_get_token())

cds_assign_star(dataset, ..., token = cds_get_token())

cds_remove_star(dataset, ..., token = cds_get_token())
```

Arguments

...	Ignored
token	An API key to be used for authentication. Will use cds_get_token() by default.
dataset	Name of the dataset to assign a star to, or remove it from.

Value

In case of cds_assign_star() returns the name of the starred dataset. In case of cds_starred() a vector of names of starred datasets. In case of cds_remove_star() returns NULL invisibly.

See Also

Other exploration-functions: cds_catalogue_vocabulary(), cds_list_datasets()

Examples

```
if (interactive() && cds_token_works()) {
  cds_assign_star("reanalysis-carra-single-levels")
  cds_starred()
  cds_remove_star("reanalysis-carra-single-levels")
}
```

cds_submit_job *Submit a download job for a dataset*

Description

Submit a request to the Copernicus Climate Data Service to download (part of) a dataset. If the request is successful, a job identifier is returned which can be used to actually download the data (`cds_download_jobs()`).

Usage

```
cds_submit_job(
  dataset,
  ...,
  wait = TRUE,
  check_quota = TRUE,
  check_licence = TRUE,
  token = cds_get_token()
)
```

Arguments

`dataset`	The dataset name to be downloaded, or a `list` returned by `cds_build_request()`. When this argument is missing the function will attempt to build a request with text on the clipboard using `cds_python_to_r()`.
`...`	Subsetting parameters passed onto `cds_build_request()`. Should be empty when `dataset` is missing or a `list`.
`wait`	A `logical` value indicating if the function should wait for the submitted job to finish. Set it to `FALSE` to continue without waiting
`check_quota`	Each account has a quota of data that can be downloaded. If this argument is set to `TRUE` (default) it is checked if the request doesn't exceed your quota. Set it to `FALSE` to skip this step and speed up the submission.
`check_licence`	Datasets generally require you to accept certain terms of use. If this argument is set to `TRUE` (default), it will be checked if you have accepted all required licences for the submitted request. Set it to `FALSE` to skip this step and speed up the submission.
`token`	An API key to be used for authentication. Will use `cds_get_token()` by default.

Value

Returns a `data.frame` containing information about the submitted job.

See Also

Other job-functions: `cds_delete_job()`, `cds_download_jobs()`, `cds_job_results()`, `cds_list_jobs()`

Examples

```
if (interactive() && cds_token_works()) {
  job <- cds_submit_job(
      dataset        = "reanalysis-era5-pressure-levels",
      variable       = "geopotential",
      product_type   = "reanalysis",
      area           = c(n = 55, w = -1, s = 50, e = 10),
      year           = "2024",
      month          = "03",
      day            = "01",
      pressure_level = "1000",
      data_format    = "netcdf"
    )

  ## Or split into two separate steps:

  req <- cds_build_request(
      dataset        = "reanalysis-era5-pressure-levels",
      variable       = "geopotential",
      product_type   = "reanalysis",
      area           = c(n = 55, w = -1, s = 50, e = 10),
      year           = "2024",
      month          = "03",
      day            = "01",
      pressure_level = "1000",
      data_format    = "netcdf"
    )
  job <- cds_submit_job(req)
}
```

Index

∗ **authentication-functions**
  cds_account_metrics, 4
  cds_check_authentication, 6
  cds_get_account, 11
  cds_get_token, 12
∗ **exploration-functions**
  cds_catalogue_vocabulary, 5
  cds_list_datasets, 14
  cds_starred, 18
∗ **helper-functions**
  cds_build_request, 4
  cds_cite_dataset, 7
  cds_dataset_form, 7
  cds_estimate_costs, 10
  cds_python_to_r, 16
∗ **job-functions**
  cds_delete_job, 8
  cds_download_jobs, 9
  cds_job_results, 13
  cds_list_jobs, 15
  cds_submit_job, 19
∗ **licences**
  cds_accept_licence, 2
  cds_accepted_licences, 3
  cds_list_licences, 16

cds_accept_licence, 2
cds_accept_licence(), *3*, *16*
cds_accepted_licences, 3
cds_accepted_licences(), *3*, *16*
cds_account_metrics, 4
cds_account_metrics(), *6*, *11, 12*
cds_assign_star *(*cds_starred*)*, 18
cds_build_request, 4
cds_build_request(), *7, 8*, *10*, *17*
cds_catalogue_vocabulary, 5
cds_catalogue_vocabulary(), *14*, *18*
cds_check_authentication, 6
cds_check_authentication(), *4*, *11, 12*
cds_cite_dataset, 7
cds_cite_dataset(), *5*, *8*, *10*, *17*
cds_dataset_form, 7
cds_dataset_form(), *5*, *7*, *10*, *17*
cds_delete_job, 8
cds_delete_job(), *9*, *13*, *15*, *19*
cds_download_jobs, 9
cds_download_jobs(), *8*, *13*, *15*, *19*
cds_estimate_costs, 10
cds_estimate_costs(), *5*, *7, 8*, *17*
cds_get_account, 11
cds_get_account(), *4*, *6*, *12*
cds_get_token, 12
cds_get_token(), *4*, *6*, *11*
cds_job_results, 13
cds_job_results(), *8, 9*, *15*, *19*
cds_list_datasets, 14
cds_list_datasets(), *6*, *18*
cds_list_jobs, 15
cds_list_jobs(), *8, 9*, *13*, *19*
cds_list_licences, 16
cds_list_licences(), *3*
cds_python_to_r, 16
cds_python_to_r(), *5*, *7, 8*, *10*
cds_remove_star *(*cds_starred*)*, 18
cds_search_datasets
  *(*cds_list_datasets*)*, 14
cds_set_token *(*cds_get_token*)*, 12
cds_starred, 18
cds_starred(), *6*, *14*
cds_submit_job, 19
cds_submit_job(), *8, 9*, *13*, *15*
cds_token_works
  *(*cds_check_authentication*)*, 6