

# Package ‘ecmwfr’

February 7, 2019

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

**Version** 1.1.0

**Description** Programmatic interface to the European Centre for Medium-Range Weather Forecasts public and restricted dataset web services (ECMWF; <<https://www.ecmwf.int/>>) as well as Copernicus's Climate Data Store (CDS; <<https://cds.climate.copernicus.eu/>>). Allows for easy downloads of weather forecasts and climate data directly to your R work space or your computer.

**URL** <https://github.com/khufkens/ecmwfr>

**BugReports** <https://github.com/khufkens/ecmwfr/issues>

**Depends** R (>= 3.4)

**Imports** httr, keyring, memoise

**License** AGPL-3

**LazyData** true

**ByteCompile** true

**RoxygenNote** 6.1.1

**Suggests** rmarkdown, covr, testthat, raster, maps, ncdf4, stars, knitr,

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Koen Hufkens [aut, cre],  
Reto Stauffer [ctb] (<<https://orcid.org/0000-0002-3798-5507/>>)

**Maintainer** Koen Hufkens <[koen.hufkens@gmail.com](mailto:koen.hufkens@gmail.com)>

**Repository** CRAN

**Date/Publication** 2019-02-06 23:11:15 UTC

## R topics documented:

<code>wf_check_request</code> . . . . .	2
<code>wf_datasets</code> . . . . .	3
<code>wf_delete</code> . . . . .	4

wf_get_key . . . . .	5
wf_product_info . . . . .	6
wf_request . . . . .	7
wf_services . . . . .	8
wf_set_key . . . . .	9
wf_transfer . . . . .	10
wf_user_info . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

wf_check_request	<i>check ECMWF / CDS data requests</i>
------------------	--

---

### Description

Check the validity of a data request, and login credentials.

### Usage

```
wf_check_request(user, request)
```

### Arguments

user	user (email address) used to sign up for the ECMWF data service, used to retrieve the token set by <a href="#">wf_set_key</a>
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 Kufkens

### See Also

[wf\\_set\\_key](#) [wf\\_transfer](#), [wf\\_request](#)

---

wf_datasets	<i>ECMWF dataset list</i>
-------------	---------------------------

---

**Description**

Returns a list of datasets

**Usage**

```
wf_datasets(user, service = "webapi", simplify = TRUE,  
            verbose = FALSE)
```

**Arguments**

user	user (email address) used to sign up for the ECMWF data service, used to retrieve the token set by <a href="#">wf_set_key</a>
service	service to use ecmwf webapi or cds (default = "webapi")
simplify	simplify the output, logical (default = TRUE)
verbose	boolean, default FALSE

**Value**

returns a nested list or data frame with the ECMWF datasets

**Author(s)**

Koen Kufkens

**See Also**

[wf\\_set\\_key](#) [wf\\_transfer](#) [wf\\_request](#)

**Examples**

```
## Not run:  
# set key  
wf_set_key(email = "test@mail.com", key = "123")  
  
# get a list of services  
wf_services("test@mail.com")  
  
# get a list of datasets  
wf_datasets("test@mail.com")  
  
## End(Not run)
```

---

wf_delete	<i>ECMWF delete request</i>
-----------	-----------------------------

---

**Description**

Deletes a staged download from the queue

**Usage**

```
wf_delete(user, url, service = "webapi", verbose = TRUE)
```

**Arguments**

user	user (email address) used to sign up for the ECMWF data service, used to retrieve the token set by <a href="#">wf_set_key</a>
url	url to query
service	character, one of ecmwf or cds depending on the data set to be deleted.
verbose	show feedback on processing

**Author(s)**

Koen Kufkens

**See Also**

[wf\\_set\\_key](#) [wf\\_transfer](#) [wf\\_request](#)

**Examples**

```
## Not run:  
# set key  
wf_set_key(email = "test@mail.com", key = "123")  
  
# get key  
wf_get_key(email = "test@mail.com")  
  
## End(Not run)
```

---

wf_get_key	<i>Get secret ECMWF / CDS token</i>
------------	-------------------------------------

---

**Description**

Returns you token set by [wf\\_set\\_key](#)

**Usage**

```
wf_get_key(user)
```

**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 Kufkens

**See Also**

[wf\\_set\\_key](#)

**Examples**

```
## Not run:  
# set key  
wf_set_key(user = "test@mail.com", key = "123")  
  
# get key  
wf_get_key(user = "test@mail.com")  
  
## End(Not run)
```

---

wf_product_info	<i>CDS product list for a given dataset</i>
-----------------	---

---

### Description

Shows and returns detailed product information about a specific data set (see [wf\\_datasets](#)).

### Usage

```
wf_product_info(user, dataset, service = "webapi", simplify = TRUE,  
               verbose = FALSE)
```

### Arguments

user	string, user ID used to sign up for the CDS data service, used to retrieve the token set by <a href="#">wf_set_key</a> .
dataset	character, name of the data set for which the product information should be loaded.
service	which service to use, one of webapi or cds
simplify	boolean, default TRUE. If TRUE the description will be returned as tidy data instead of a nested list.
verbose	boolean, default FALSE.

### 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:  
# Opend description in browser  
wf_product_info(NULL, "reanalysis-era5-single-levels")  
  
# Return information  
info <- wf_product_info(NULL,  
  "reanalysis-era5-single-levels", show = FALSE)  
names(info)  
  
## End(Not run)
```

---

wf_request	<i>ECMWF data request and download</i>
------------	--

---

### 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](#). The function only allows NetCDF downloads, and will override calls for grib data. Note that the function will do some basic checks on the request input to identify possible problems.

### Usage

```
wf_request(user, request, transfer = FALSE, path = tempdir(),
           time_out = 3 * 3600, verbose = TRUE)
```

### Arguments

user	user (email address) used to sign up for the ECMWF data service, used to retrieve the token set by <a href="#">wf_set_key</a>
request	nested list with query parameters following the layout as specified on the ECMWF API page
transfer	logical, download data TRUE or FALSE (default = FALSE)
path	path were to store the downloaded data
time_out	how long to wait on a download to start (default = 3*3600 seconds).
verbose	show feedback on processing

### Value

a download query staging url or (invisible) filename of the NetCDF file on your local disc

### Author(s)

Koen Kufkens

### See Also

[wf\\_set\\_key](#) [wf\\_transfer](#)

### Examples

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

request <- = list(stream = "oper",
                 levtype = "sfc",
```

```

param = "167.128",
dataset = "interim",
step = "0",
grid = "0.75/0.75",
time = "00",
date = "2014-07-01/to/2014-07-02",
type = "an",
class = "ei",
area = "50/10/51/11",
format = "netcdf",
target = "tmp.nc")

# get the default test data
wf_request(user = "test@mail.com", request = request)

## End(Not run)

```

---

wf\_services

*ECMWF services list*


---

### Description

Returns a list of services

### Usage

```
wf_services(user, simplify = TRUE)
```

### Arguments

user	user (email address) used to sign up for the ECMWF data service, used to retrieve the token set by <a href="#">wf_set_key</a>
simplify	simplify the output, logical (default = TRUE)

### Value

returns a nested list or data frame with the ECMWF services

### See Also

[wf\\_set\\_key](#) [wf\\_transfer](#) [wf\\_request](#)

### Examples

```

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

```



```
# get a list of services
wf_services("test@mail.com")

# get a list of datasets
wf_services("test@mail.com")

## 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(user, key)
```

## Arguments

user	user (email address) used to sign up for the ECMWF data service
key	token provided by ECMWF

## Author(s)

Koen Kufkens

## See Also

[wf\\_get\\_key](#)

## Examples

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

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

## 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(user, url, service = "webapi", path = tempdir(),
            filename = tempfile("ecmwfr_", fileext = ".nc", tmpdir = ""),
            verbose = TRUE)
```

### Arguments

user	user (email address) used to sign up for the ECMWF data service, used to retrieve the token set by <a href="#">wf_set_key</a> .
url	url to query
service	which service to use, one of webapi or cds
path	path were to store the downloaded data
filename	filename to use for the downloaded data
verbose	show feedback on data transfers
...	forwarded to <a href="#">wf_transfer</a>

### Value

a netCDF of data on disk as specified by a [wf\\_request](#)

### Author(s)

Koen Kufkens

### See Also

[wf\\_set\\_key](#) [wf\\_request](#)

### Examples

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

# request data and grab url and try a transfer
r <- wf_request("test@mail.com", transfer = FALSE)
```

```
# check transfer, will download if available
wf_transfer("test@email.com", url = r$href)

## End(Not run)
```

---

wf_user_info	<i>ECMWF WebAPI user info query</i>
--------------	-------------------------------------

---

### Description

Returns user info for the ECMWF WebAPI

### Usage

```
wf_user_info(user)
```

### Arguments

user	user (email address) used to sign up for the ECMWF data service, used to retrieve the token set by <a href="#">wf_set_key</a>
------	---

### Value

returns a data frame with user info

### See Also

[wf\\_set\\_key](#) [wf\\_services](#) [wf\\_datasets](#)

### Examples

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

# get user info
wf_user_info("test@mail.com")

## End(Not run)
```

# Index

\*Topic **climate,**

wf\_check\_request, 2  
wf\_datasets, 3  
wf\_delete, 4  
wf\_product\_info, 6  
wf\_request, 7  
wf\_services, 8  
wf\_transfer, 10  
wf\_user\_info, 11

\*Topic **data**

wf\_check\_request, 2  
wf\_datasets, 3  
wf\_delete, 4  
wf\_product\_info, 6  
wf\_request, 7  
wf\_services, 8  
wf\_transfer, 10  
wf\_user\_info, 11

\*Topic **download,**

wf\_check\_request, 2  
wf\_datasets, 3  
wf\_delete, 4  
wf\_product\_info, 6  
wf\_request, 7  
wf\_services, 8  
wf\_transfer, 10  
wf\_user\_info, 11

\*Topic **key**

wf\_get\_key, 5  
wf\_set\_key, 9

\*Topic **management**

wf\_get\_key, 5  
wf\_set\_key, 9

\*Topic **re-analysis**

wf\_check\_request, 2  
wf\_datasets, 3  
wf\_delete, 4  
wf\_product\_info, 6  
wf\_request, 7

wf\_services, 8  
wf\_transfer, 10  
wf\_user\_info, 11

wf\_check\_request, 2  
wf\_datasets, 3, 6, 11  
wf\_delete, 4  
wf\_get\_key, 5, 9  
wf\_product\_info, 6  
wf\_request, 2-4, 7, 8, 10  
wf\_services, 8, 11  
wf\_set\_key, 2-8, 9, 10, 11  
wf\_transfer, 2-4, 7, 8, 10, 10  
wf\_user\_info, 11