

Package ‘camsRad’

May 8, 2026

Type Package

Title Client for CAMS Radiation Service

Version 0.3.0

Maintainer Lukas Lundstrom <lukas.rokka@gmail.com>

Description Copernicus Atmosphere Monitoring Service (CAMS) radiations service provides time series of global, direct, and diffuse irradiances on horizontal surface, and direct irradiation on normal plane for the actual weather conditions as well as for clear-sky conditions.

The geographical coverage is the field-of-view of the Meteosat satellite, roughly speaking Europe, Africa, Atlantic Ocean, Middle East. The time coverage of data is from 2004-02-01 up to 2 days ago. Data are available with a time step ranging from 15 min to 1 month. For license terms and to create an account, please see <<http://www.soda-pro.com/web-services/radiation/cams-radiation-service>>.

License MIT + file LICENSE

Depends R (>= 3.1.0)

Imports httr (>= 1.2.1), xml2 (>= 1.0.0)

Suggests ncd4, roxygen2, knitr, testthat, rmarkdown

URL <https://github.com/ropenscilabs/camsRad>

BugReports <https://github.com/ropenscilabs/camsRad/issues>

LazyData TRUE

RoxygenNote 5.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Lukas Lundstrom [aut, cre]

Repository CRAN

Date/Publication 2016-11-30 16:28:49

Contents

camsRad	2
cams_api	2
cams_get_mcclear	4
cams_get_radiation	5
cams_set_user	6

Index	7
--------------	----------

camsRad	<i>R client for CAMS radiation service</i>
---------	--

Description

CAMS radiation service provides time series of global, direct, and diffuse irradiances on horizontal surface, and direct irradiation on normal plane for the actual weather conditions as well as for clear-sky conditions. The geographical coverage is the field-of-view of the Meteosat satellite, roughly speaking Europe, Africa, Atlantic Ocean, Middle East (-66 to 66 degrees in both latitudes and longitudes). The time coverage of data is from 2004-02-01 up to 2 days ago. Data are available with a time step ranging from 15 min to 1 month.

cams_api	<i>API client for Rhrefhttp://www.soda-pro.com/web-services/radiation/cams-radiation-service CAMS radiation service</i>
----------	---

Description

API client for **CAMS radiation service**

Usage

```
cams_api(lat, lng, date_begin, date_end, alt = -999, time_step = "PT01H",
         time_ref = "UT", verbose = FALSE, service = "get_cams_radiation",
         format = "application/csv", filename = "")
```

Arguments

lat	Latitude, in decimal degrees. Required
lng	Longitude, in decimal degrees. Required
date_begin	Start date as 'yyyy-mm-dd' string. Required
date_end	End date as 'yyyy-mm-dd' string. Required
alt	Altitude in meters, use -999 to let CAMS decide. Default -99
time_step	Aggregation: 'PT01M' for minutes, 'PT15M' for 15 minutes, 'PT01H' for hourly, 'P01D' for daily, 'P01M' for monthly. Default 'PT01H'

time_ref	Time reference: 'UT' for universal time, 'TST' for true solar time. Default 'UT'
verbose	TRUE for verbose output. Default "FALSE"
service	'get_mcclear' for CAMS McClear data, 'get_cams_radiation' for CAMS radiation data. Default 'get_cams_radiation'
format	'application/csv', 'application/json', 'application/x-netcdf' or 'text/csv'. Default 'application/csv'
filename	path to file on disk to write to. If empty, data is kept in memory. Default empty

Value

list(ok=TRUE/FALSE, response=response). If ok=TRUE, response is the response from httr::GET. If ok=FALSE, response holds exception text

Examples

```
## Not run:
library(ncdf4)

filename <- paste0(tempfile(), ".nc")

# API call to CAMS
r <- cams_api(
  60, 15, # latitude=60, longitude=15
  "2016-06-01", "2016-06-10", # for 2016-06-01 to 2016-06-10
  time_step="PT01H", # hourly data
  service="get_cams_radiation", # CAMS radiation
  format="application/x-netcdf", # netCDF format
  filename=filename) # file to save to

# Access the on disk stored ncdf4 file
nc <- nc_open(r$response$content)
# list names of available variables
names(nc$var)

# create data.frame with timestamp and global horizontal irradiation
df <- data.frame(datetime=as.POSIXct(nc$dim$time$vals, "UTC",
                                     origin="1970-01-01"),
                 GHI = ncv_get(nc, "GHI"))

plot(df, type="l")

nc_close(nc)

## End(Not run)
```

cams_get_mcclear	<i>Retrieve McClear clear sky solar radiation data</i>
------------------	--

Description

Retrieve McClear clear sky solar radiation data

Usage

```
cams_get_mcclear(lat, lng, date_begin, date_end, time_step = "PT01H",  
  alt = -999, verbose = FALSE)
```

Arguments

lat	Latitude, in decimal degrees. Required
lng	Longitude, in decimal degrees. Required
date_begin	Start date as 'yyyy-mm-dd' string. Required
date_end	End date as 'yyyy-mm-dd' string. Required
time_step	Aggregation: 'PT01M' for minutes, 'PT15M' for 15 minutes, 'PT01H' for hourly, 'P01D' for daily, 'P01M' for monthly. Default 'PT01H'
alt	Altitude in meters, use -999 to let CAMS decide. Default -99
verbose	TRUE for verbose output. Default "FALSE"

Value

A data frame with requested solar data

Examples

```
## Not run:  
df <- cams_get_mcclear(  
  lat=60, lng=15, date_begin="2016-01-01", date_end="2016-01-15")  
print(head(df))  
  
## End(Not run)
```

cams_get_radiation *Retrieve CAMS solar radiation data*

Description

Retrieve CAMS solar radiation data

Usage

```
cams_get_radiation(lat, lng, date_begin, date_end, time_step = "PT01H",  
alt = -999, verbose = FALSE)
```

Arguments

lat	Latitude, in decimal degrees. Required
lng	Longitude, in decimal degrees. Required
date_begin	Start date as 'yyyy-mm-dd' string. Required
date_end	End date as 'yyyy-mm-dd' string. Required
time_step	Aggregation: 'PT01M' for minutes, 'PT15M' for 15 minutes, 'PT01H' for hourly, 'P01D' for daily, 'P01M' for monthly. Default 'PT01H'
alt	Altitude in meters, use -999 to let CAMS decide. Default -99
verbose	TRUE for verbose output. Default "FALSE"

Value

A data frame with requested solar data

Examples

```
## Not run:  
# Get hourly solar radiation data  
df <- cams_get_radiation(  
  lat=60, lng=15,  
  date_begin="2016-06-01", date_end="2016-06-15")  
head(df)  
  
# Get daily solar radiation data  
df <- cams_get_radiation(  
  lat=60, lng=15,  
  date_begin="2016-06-01", date_end="2016-06-15",  
  time_step="P01D")  
head(df)  
  
## End(Not run)
```

cams_set_user	<i>Set username used for authentication by CAMS radiation service</i>
---------------	---

Description

Set username used for authentication by CAMS radiation service

Usage

```
cams_set_user(username)
```

Arguments

username	Email registered at soda-pro.com. Required
----------	--

Examples

```
## Not run:  
# cams_set_user("your@email.com") # An email registered at soda-pro.com  
  
## End(Not run)
```

Index

`cams_api`, [2](#)
`cams_get_mcclear`, [4](#)
`cams_get_radiation`, [5](#)
`cams_set_user`, [6](#)
`camsRad`, [2](#)
`camsRad-package (camsRad)`, [2](#)