retostauffer (https://retostauffer.org)

NAVIGATION ☰

Short guide for downloading ERA5

Some time ago I have written a short introduction on **how to download ERA INTERIM data (https://retostauffer.org/code/Download-ERA-INTERIM/)** from **ecmwf.int (https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/era-interim)**. ERA INTERIM was the fourth version of ECMWFs global reanalysis data set. Just recently (mid 2018) the ECMWF started publishing the new ERA-5 data set with a spatial resolution of about 32 km times 32 kilometers.

Please note that this data set is still in a pre-release state. Every now and then some fields are getting updated as some problems have been identified in the data. The data set is currently going back to January 2000 but will be further extended into the past.

Access to ERA-5 is very similar to accessing ERA INTERIM, however, the new data set is redistributed over the **Copernicus Climate Data Store (https://cds.climate.copernicus.eu/)**. Even if access via the ECMWF public data access portal is still possible it is not recommended to do so as it will be shut down in the near future.

All we need to download data:

- A valid CDS user account and create a local file containing the user token.
- The `cdsapi` python package plus a few lines of python code.

Even if you are not yet familiar to python you'll see that it is pretty simple!

Create a user account, get your API key

If you don't own a user account yet **create a new account (https://cds.climate.copernicus.eu/user/register?destination=/drupal_auth_check)** on the CDS website. After activating your account use your new account to log in. Your user name will be show in the top right corner. You can now enter your user profile by clicking on your user name. On the profile you'll find your user id (**UID**) and your personal **API Key**.

Create local key file

For batch script data downloads you'll have to create a local ASCII file with your user information (UID, API key) which is used by the python package (`cdsapi`). To do so (linux) simply create a file called `.cdsapirc` in your home directory and add the following two lines:

```
url: https://cds.climate.copernicus.eu/api/v2
key: 1234:abcdefghij-134-abcdefgadf-82391b9d3f
```

where `1234` is your personal user ID (UID), the part behind the colon your personal API key. Line one simply contains the URL to the web API. More details **can be found here (https://cds.climate.copernicus.eu/api-how-to)**.

Install cdsapi package

What you'll need next is the python package `cdsapi`. As it is available on **PyPI (https://pypi.org/project/cdsapi/)** you can simply install it using your preferred python package installer, e.g., pip:

```
pip install cdsapi
```

For the sake of completeness: I am using `cdsapi 2.18.4` at the moment.

Write your first data request

To get data you'll have to write a small python script which sends requests to the CDS servers. Here is one simple example how the python scripts will look like:

```
# Import cdsapi
import cdsapi
# Open a new Client instance
c = cdsapi.Client()
# Send your request (download data)
c.retrieve('reanalysis-era5-single-levels', {
        'product_type': 'reanalysis',
        'variable':     'orography',
        'year':         '2016',
        'month':        '01',
        'day':          '01',
        'time':         '00:00'
    }, 'orography.grib')
```

Copy the lines above into a python script (e.g., `orography.py`) and execute it (`python orography.py`). In this simple case I am requesting orography (`variable`) reanalysis (`product_type`) data for January 1, 2016 00:00 UTC (`year`, `month`, `day`, `time`). Orography is a single-level variable and thus identified by the `reanalysis-era5-single-levels` product keyword. Once you executed this script you should now have a file called `orography.grib` just next to your `orography.py` script file.

Note: the **CDS (https://cds.climate.copernicus.eu)** provides a nice web interface where you can create these requests by selecting the parameters and time periods you need. Under **Show API Request** you'll see the request (as shown above) for your custom selection, but not all options are shown. Two different data sets are available:

- **ERA-5 single-level data (https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=overview)**
- **ERA-5 pressure-level data (https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=overview)**

Make a more useful request

Let's assume we are interested in the 00 UTC temperature and geopotential height of the 900 and 700 hectopascal level for 2017 over Europe for some reason. The corresponding request looks like this:

```
# Import cdsapi and create a Client instance
import cdsapi
c = cdsapi.Client()
# More complex request
c.retrieve("reanalysis-era5-pressure-levels", {
        "product_type":   "reanalysis",
        "format":         "netcdf",
        "area":           "52.00/2.00/40.00/20.00",
        "variable":       ["geopotential","temperature"],
        "pressure_level": ["700","900"],
        "year":           "2017",
        "month":          ["01","02","03","04","05","06","07","08","09","10","11","12"],
        "day":            ["01","02","03","04","05","06","07","08","09","10","11",
                           "12","13","14","15","16","17","18","19","20","21","22",
                           "23","24","25","26","27","28","29","30","31"],
        "time":           "00"
    }, "output.nc")
```

- Product identifier: `reanalysis-era5-pressure-levels` as we are interested in pressure level data
- Product type: `reanalysis` (as before)
- Format: instead of downloading grib1 data we would like to have **NetCDF** data. This is nice as the grib1 file format is a bit unhandy sometimes and can be specified using the `format` key.

- Spatial extent: the keyword `area` allows to download a very specific subset. The definition is `N/E/S/W` in degrees longitude and latitude. Negative values correspond to `S` and `E`. In this case a domain over Europe.
- Fields: we would like to get geopotential height and temperature (`variable`) on two levels, namely 900 and 700 hectopascal (`pressure_level`).
- As we are interested in 00 UTC data for 2018 we simply specify `"time":"00"`, `"year":"2018"`, all months (`01` to `12`) and all days (`01` to `31`). Don't worry that some combinations do not exist (e.g., February 31), the API will simply ignore these fields.

At the end you should get the output file `output.nc`. To drop some numbers: the output file is of about **10 MB** in this case and the whole request took me **less than five minutes**.

Batch-scripting the requests

Rather than copy-pasting API requests you can, of course, also write some more fancy scripts. The script below is a script I've written few weeks ago to download a set of files for one of our current studies.

- Reto's **ERA5 Downloader (https://retostauffer.org/data/ERA5_Downloader.py)** (**download the script here (https://retostauffer.org/data/ERA5_Downloader.py)**)

Feel free to use, modify and redistribute the script (GPL2 license). Please note that the script is quite likely not free of bugs. If you find some please let me know :).

Some of the features:

- Should be python3 and python2 ready.
- Provides some help (call `python ERA5_Downloader.py --help`).
- Downloads one parameter at a time.
- Is downloading data in junks (month/year) stored in a sub-folder called `era5_data`.
- Checks whether a file has already been downloaded.
- **Is subsetting Europe!** The `area` is hard-coded in the script, needs to be adjusted depending on what you want to have.
- This script has been written for one of my applications in only few quarter of minutes and is far away from beeing perfect or super flexible. However, it might be a good starter if you are looking for a script to start with.

Output of ERA5_Downloader.py –help

If you missed the link above: the **ERA5 Downloader (https://retostauffer.org/data/ERA5_Downloader.py)** can be **downloaded here (click) (https://retostauffer.org/data/ERA5_Downloader.py)**.

```
Usage: Usage:

    ERA5_Downloader.py --years <years> --parameter <parameter> --level <level>
    ERA5_Downloader.py -y <years> -p <parameter> -l <level>

    Downloading ERA5 reanalysis data from Copernicus Climate Data
    Services. NOTE: Not yet made for ensemble ERA5!

    Requires `cdsapi` to be installed (python package to access
    Copernicus Climate Data Services, CDS).

    How to install cdsapi and the required API key:
    * https://cds.climate.copernicus.eu/api-how-to

    Available parameters for "single level" (-t/--type sf):
    * https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=form

    Available parameters for "pressure level" (-t/--type pl):
    * https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=form

    Usage examples:

    10m wind speed for the year 2018
    >>> python ERA5_Downloader.py --years 2018 --parameter 10m_u_component_of_wind

    10m wind speed for 2010 and 2018
    >>> python ERA5_Downloader.py --years 2010,2018 --parameter 10m_u_component_of_wind

    10m wind speed for 2008, 2009, 2010, ..., 2018
    >>> python ERA5_Downloader.py --years 2008-2018 --parameter 10m_u_component_of_wind

    700hPa geopotential height, 2018
    >>> python ERA5_Downloader.py --years 2018 --parameter geopotential --level 700



Options:
  -h, --help            show this help message and exit
  -y YEARS, --years=YEARS
                        years for which the data should be downloaded. Can
                        either be a single year (e.g., 2010), a comma
                        separated list (e.g., 2010,2011), or a sequence of the
                        form <bgn>-<end> (e.g., 2010-2018; 2010 to 2018).
  -p PARAM, --parameter=PARAM
                        the parameter/variable to be downloaded. Please check
                        the cds website to see what's availabe.
  -l LEVEL, --level=LEVEL
                        level, only for pressure level data. For "single level
                        data" (e.g., surface level variables) do not set this
                        parameter!
  -t, --test            development flag. If set, only one day (January 1th of
                        the first year) will be downloaded before the script
                        stops. The output file will be labeled with "_test_".
```

If you improve and/or extend the code it would be great to send me a message and the extensions such that I can include the changes in the original repository.

📅 2018-12-15 🗂 CODE
🏷 code 🏷 python 🏷 data

« Python colorspace package (https://retostauffer.org/news/python-colorspace-v0.1.0/)

Archive (https://retostauffer.org/blog/archive/)

Convenient GFS Reforecast V2 Downloader » (https://retostauffer.org/software/PyGFSV2-beta/)

HEADER IMAGE CREDIT

... goes to belart84 (https://unsplash.com/@belart84) on Unsplash (https://unsplash.com/).

IF YOU NEED TO CONTACT ME

e-Mail Address: reto.stauffer@uibk.ac.at

SERVICES

()
Contact/About (/about/)
Privacy Policy (/privacypolicy/)
RSS (/feed.xml)
Atom (/atom.xml)
sitemap.xml (/sitemap.xml)

THANKS TO

()
Jekyll theme by Phlow (https://github.com/Phlow)
Images by Unsplash (http://unsplash.com/)