

Earthkit-data introduction

Earthkit 1.0 training

Development Section



earthkit-data

- Retrieves data (GRIB, BUFR, NetCDF, ODB, ...) from sources such as
 - MARS and the CDS, ECMWF Opendata
 - FDB memory stream
 - files and URLs
- Conversion: e.g. GRIB to Xarray
- Writing to targets
- Provides input for the other **earthkit** components

Retrieving data from the ADS

The `ads` data source provides access to the [Copernicus Atmosphere Data Store \(ADS\)](#).

```
[1]: import earthkit.data as ekd
```

Getting GRIB data

```
[2]: ds = ekd.from_source(
    "ads",
    "cams-global-reanalysis-eac4",
    request=dict(
        variable=["particulate_matter_10um", "particulate_matter_1um"],
        area=[50, -50, 20, 50], # N,W,S,E
        date="2012-01-01",
        time="12:00"
    ),
)
```

BUFR: using TEMP data

We load a BUFR file with radiosonde observations. First we ensure the example file is available.

```
[3]: ds.to_fieldlist().
[3]:
```

	parameter.variable	time
0	pm10	
1	pm1	

```
[1]: import earthkit.data as ekd
     ekd.download_example_file("temp_10.bufr")
```

```
[2]: ds = ekd.from_source("file", "temp_10.bufr").to_featurelist()
```

Message list

Reading data from URLs as a stream

```
[1]: import earthkit.data as ekd
```

earthkit-data can read GRIB data from a URL as a stream without writing anything to disk. This can be achieved with the `stream=True` kwarg when calling `from_source()`.

```
[2]: ds = ekd.from_source("url",
    "https://sites.ecmwf.int/repository/earthkit-data/examples/test4.grib",
    stream=True).to_fieldlist()
```

The resulting object only supports one iteration. Having finished the iteration the stream is consumed

<https://earthkit-data.readthedocs.io/en/latest/>

Getting the data

```
import earthkit.data
```

The method to use is **from_source()**:

```
ds = earthkit.data.from_source('file', "test.grib")
```

```
ds = earthkit.data.from_source("url",  
    "https://get.ecmwf.int/repository/test-data/earthkit-data/test-data/t_pl.grib")
```

```
ds = earthkit.data.from_source("mars",  
    param=["2t", "msl"],  
    levtype="sfc",  
    area=[50, -10, 40, 10], #N,W,S,E  
    grid=[2, 2],  
    date="2023-05-10")
```

Conversions

- **from_source()** does not load the data but returns an object with some basic information and conversion methods
- Then we need to choose the best representation for our work and convert the data e.g. with:
- **to_fieldlist()**
- **to_xarray()**
- **to_pandas()**

```
import earthkit.data as ekd

d = ekd.from_source("file", "test4.grib")
d
```

GRIB file

path	test4.grib
size	509.5 KiB
types	fieldlist, pandas, xarray, numpy, array

```
d.available_types
```

```
['fieldlist', 'pandas', 'xarray', 'numpy', 'array']
```

```
d.to_fieldlist().ls()
```

	parameter.variable	time.valid_datetime	time.base_datetime	time.step	vertical.level	vertical.level_ty
0	t	2007-01-01 12:00:00	2007-01-01 12:00:00	0 days	500	pressu
1	z	2007-01-01 12:00:00	2007-01-01 12:00:00	0 days	500	pressu
2	t	2007-01-01 12:00:00	2007-01-01 12:00:00	0 days	850	pressu
3	z	2007-01-01 12:00:00	2007-01-01 12:00:00	0 days	850	pressu

```
d.to_xarray()
```

xarray.Dataset

► Dimensions: (level: 2, latitude: 181, longitude: 360)

Targets and encoders

- `to_target()` can write the data into target with a suitable encoder

```
fl = ekd.from_source("file", "test4.grib").to_fieldlist

# write slice of a fieldlist into a file with a GRIB encoder
fl[0:2].to_target("file", "_my.grib")

# write slice of a fieldlist into an FDB with a GRIB encoder
fl[0:2].to_target("fdb", config=config)
```