



Koninklijk Nederlands
Meteorologisch Instituut
Ministerie van Infrastructuur en Waterstaat

EDR API for Observation Data

API Workshop

23 november 2022

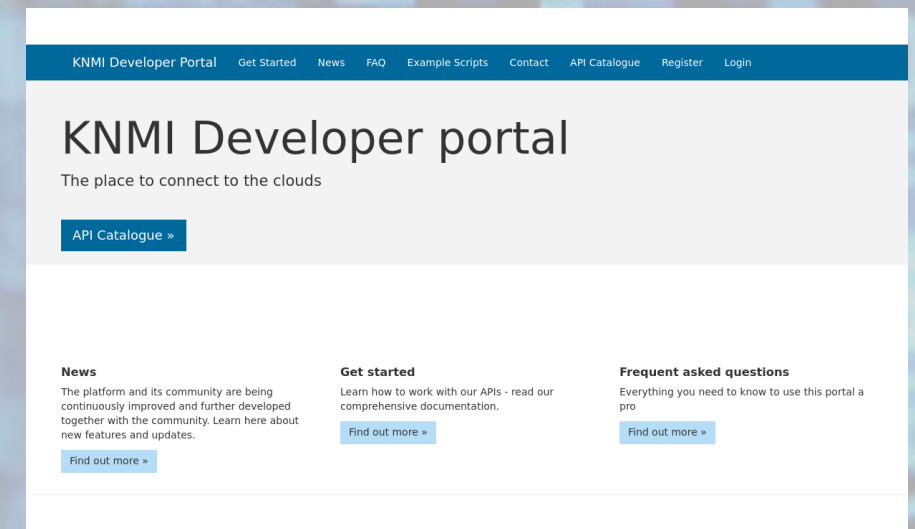
Paul van Schayck



KNMI Developer Portal

- > API keys:
 - Anonymous (public), with restrictions (*fair use*)
 - Personal (bound to a person or organisations)
- > Documentation
- > Example scripts
- > News

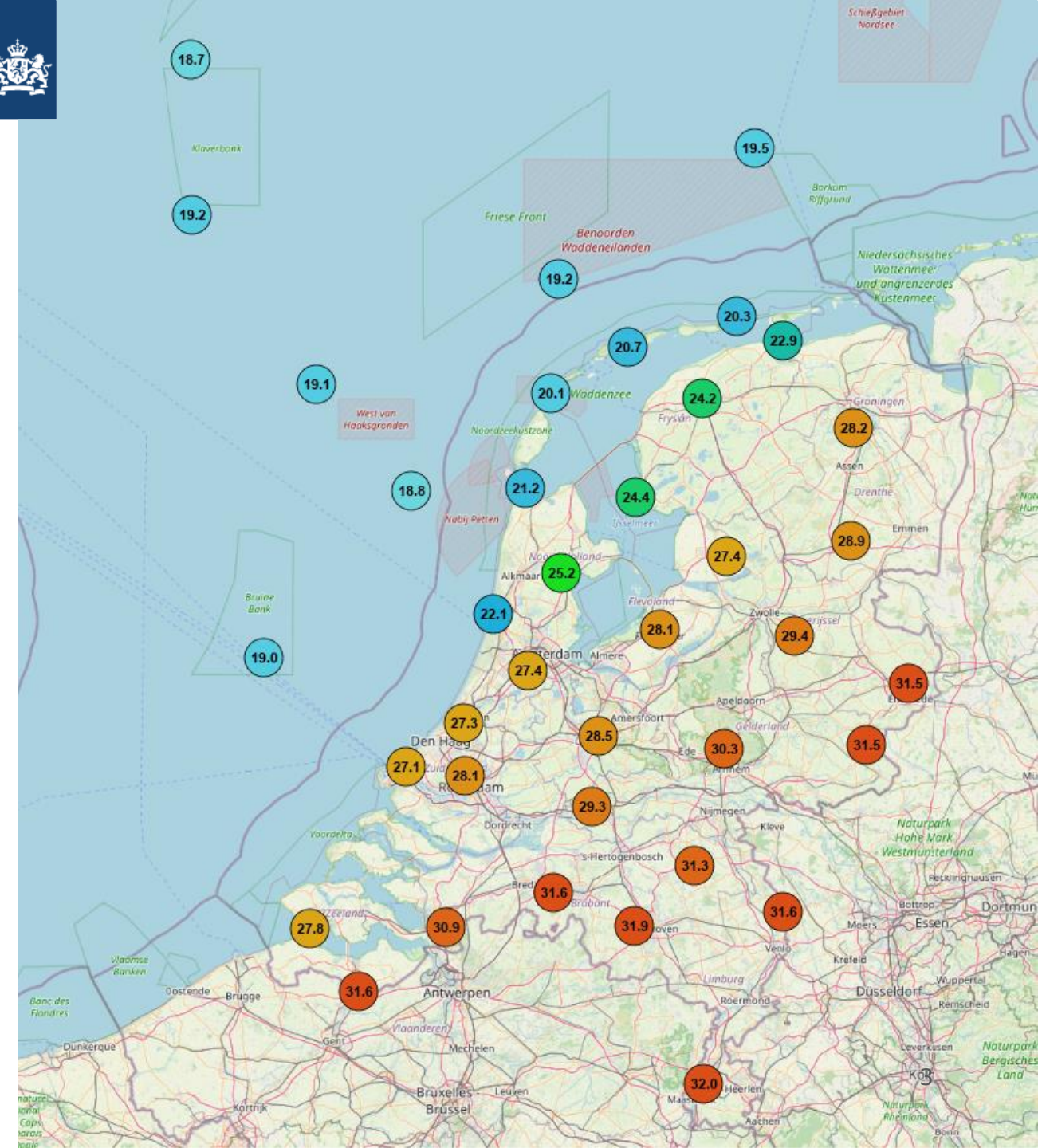
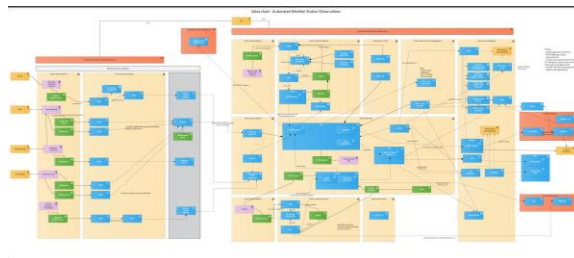
<https://developer.dataplatform.knmi.nl>





From Observations to API

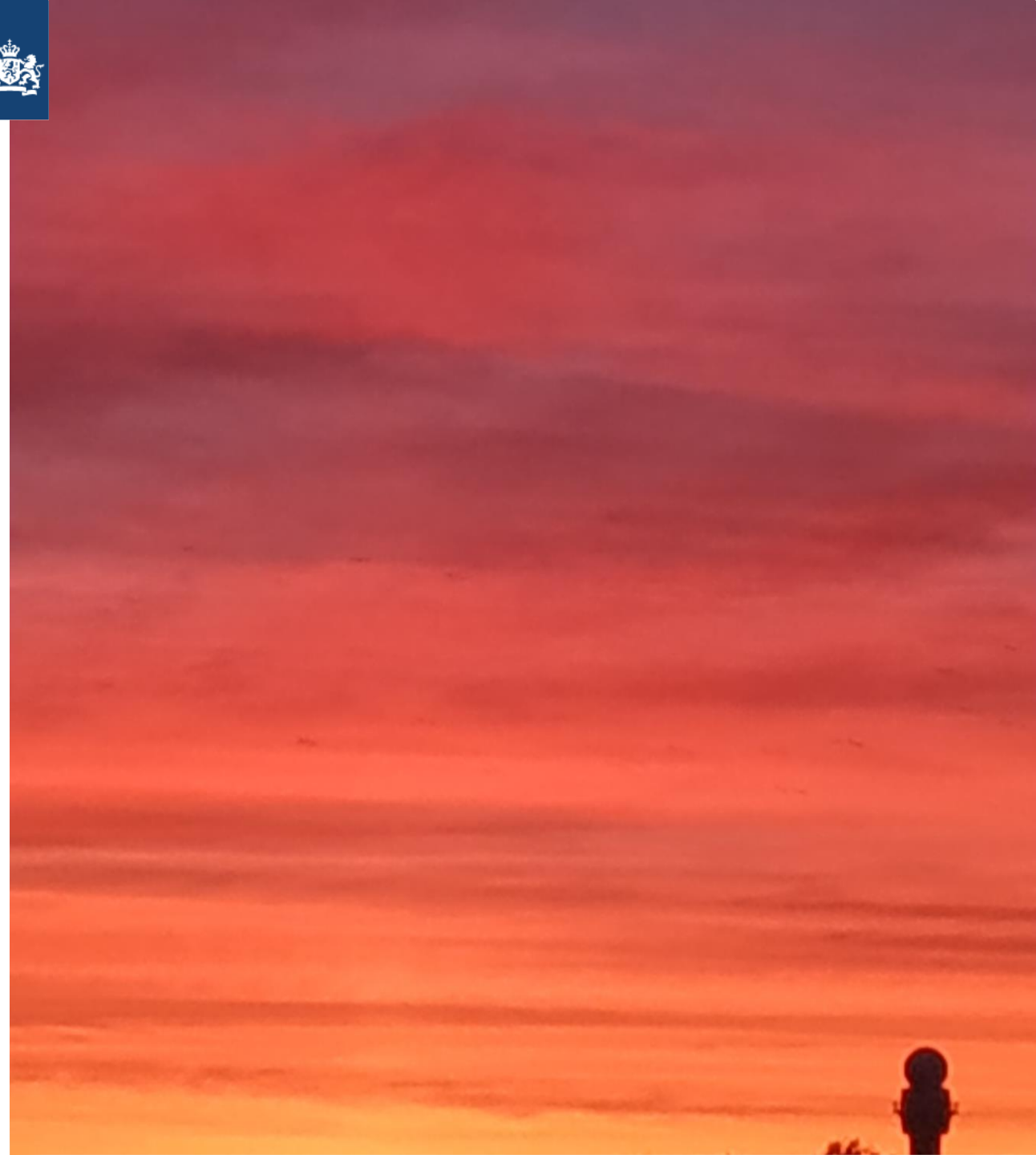
- 40+ automatic weather stations (AWS)
- Many steps:
 - Sensor
 - Intern MeetNet
 - Validation
 - Climate database





Why a new API?

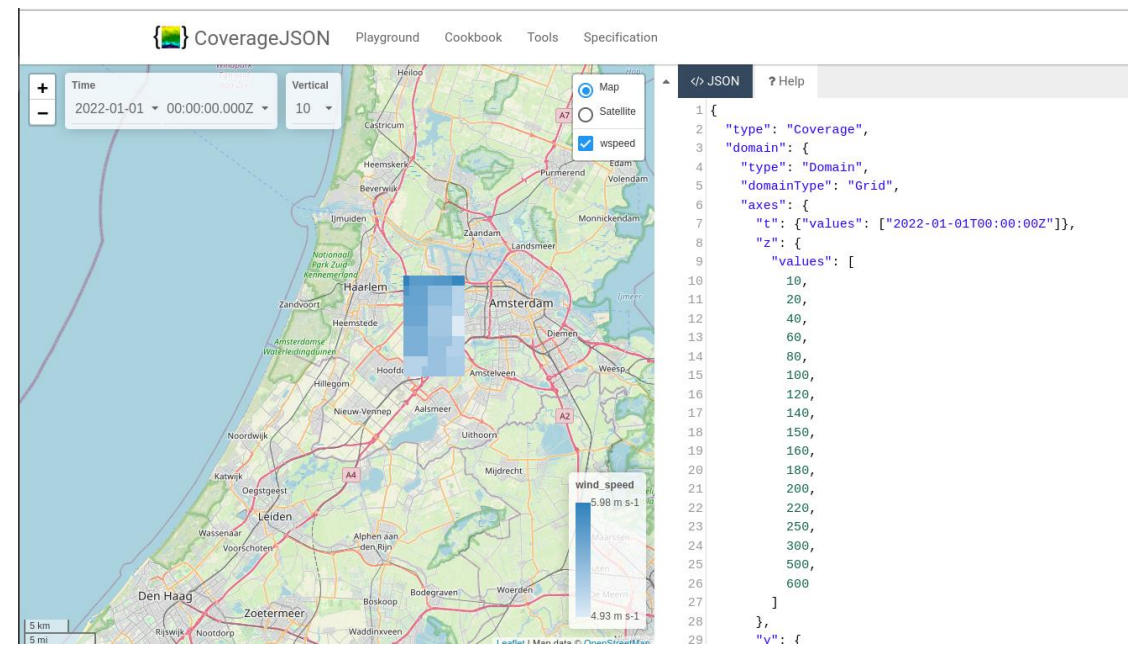
- › Open Data API
 - File based in 10 min intervals
 - Only NetCDF
- › Not very novice friendly
- › Difficult to integrate in applications
- › No official specification





EDR API for Observations

- Following EDR specification
- OpenGeoSpatial Consortium
 - Enviromental Data Retrieval (EDR)
 - CoverageJSON
 - Also: GeoJSON, NetCDF, CSV
- Functionality:
 - (geo) filtering
 - Specific variables
 - Time series (live and historic)



<https://covjson.org/playground/>



CoverageJSON

A format for publishing geotemporal data to the web.

```
{
  "type" : "Coverage",
  "domain" : {
    "type": "Domain",
    "domainType": "Point",
    "axes": {
      "x": { "values": [1] },
      "y": { "values": [20] },
      "z": { "values": [1] },
      "t": { "values": ["2008-01-01T04:00:00Z"] }
    },
    "referencing": [...]
  },
  "parameters" : {
    "temperature": {...}
  },
  "ranges" : {
    "temperature" : {
      "type" : "NdArray",
      "dataType": "float",
      "values" : [...]
    }
  }
}
```


EDR API Endpoints

An Environmental Data Retrieval (EDR) API provides a family of lightweight interfaces to access Environmental Data resources.



Capabilities

Essential characteristics of the information available from the API.

GET / landing page of this API

GET /**conformance** Information about standards that this API conforms to

GET /**collections** List the available collections from the service

Collection metadata

Description of the information available from the collections

GET /**collections/{collectionId}** List query types supported by the collection

Instance metadata

Description of the information available from the instances of a collection

GET /**collections/{collectionId}/instances** List data instances of {collectionId}

Collection data queries

Data queries available.

GET /**collections/{collectionId}/position** Query end point for position queries of collection {collectionId}

GET /**collections/{collectionId}/radius** Query end point for radius queries of collection {collectionId}

GET /**collections/{collectionId}/area** Query end point for area queries of collection {collectionId} defined by a polygon

GET /**collections/{collectionId}/cube** Query end point for Cube queries of collection {collectionId} defined by a cube

GET /**collections/{collectionId}/trajectory** Query end point for trajectory queries of collection {collectionId} defined by a wkt linestring

GET /**collections/{collectionId}/corridor** Query end point for Corridor queries of collection {collectionId} defined by a polygon

GET /**collections/{collectionId}/items** List available items

GET /**collections/{collectionId}/items/{itemId}** Return item {itemId} from collection {collectionId}

GET /**collections/{collectionId}/locations** List available location identifiers for the collection



Using the EDR API for Observation data

Setting up

```
import requests
import pandas
import covjson2pandas
```

```
collection = "observations"
base_url = f"https://api.dataplatform.knmi.nl/edr/collections/{collection}"
token = "YOUR TOKEN"
```

```
headers = {
    'Authorization': token
}
```

Metadata

```
r = requests.get(base_url, headers=headers)
r.raise_for_status()
data = r.json()
```

```
,
"id": "observations",
"extent": {
  "spatial": {
    "bbox": [
      [
        12.13,
        -68.27583333333333,
        55.39916666666667,
        7.14916666666667
      ]
    ],
    "crs": "WGS84"
  },
  "temporal": {
    "interval": [
```




Using the EDR API for Observation data

Requesting data

```
params = {  
    'coord': 'POINT(5.14527777777778 51.8577777777778)',  
    'datetime': '2022-07-19T18:00:00Z/2022-07-19T18:10:00Z',  
    'parameter-name': 't_dryb_10'  
}  
r = requests.get(base_url + "/position", params, headers=headers)  
r.raise_for_status()  
data = r.json()
```

```
{  
  "type": "Coverage",  
  "domain": {  
    "type": "Domain",  
    "domainType": "PointSeries",  
    "axes": {  
      "x": {  
        "values": [  
          5.14527777777778  
        ]  
      },  
      "y": {  
        "values": [  
          51.8577777777778  
        ]  
      },  
      "t": {  
        "values": [  
          "2022-07-19T18:00:00Z",  
          "2022-07-19T18:10:00Z"  
        ]  
      }  
    },  
    "ranges": {  
      "t_dryb_10": {  
        "type": "NdArray",  
        "dataType": "float",  
        "axisNames": [  
          "x",  
          "y",  
          "t"  
        ],  
        "shape": [  
          1,  
          1,  
          2  
        ],  
        "values": [  
          33.3,  
          32.6  
        ]  
      }  
    }  
  }  
}
```

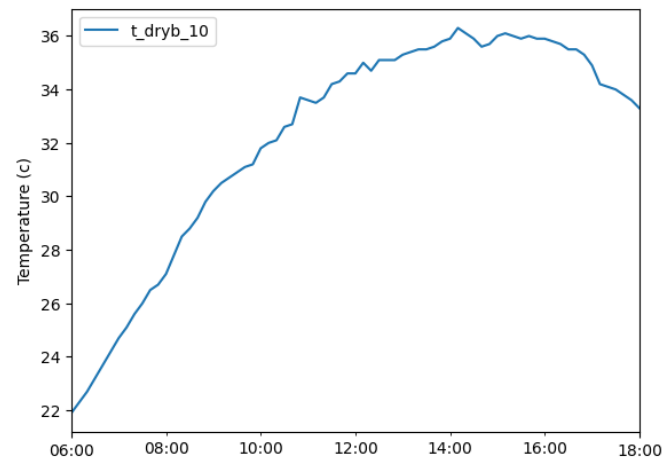


Demo's

Plotting temperature

```
ax = df.plot(y='t_dryb_10')
ax.set_ylabel("Temperature (c)")
ax
```

```
<AxesSubplot: ylabel='Temperature (c)'>
```



Postman

edr-api-examples / collections-data-position

GET [https://api.dataplatform.knmi.nl/edr/collections/observations/position?datetime=2022-07-22T04:10:00Z/2022-07-22T06:10:00Z&coord=POINT\(3.275 51.997777777777778\)¶meter-name=L_dryb_10,u_10&crs=WGS84](https://api.dataplatform.knmi.nl/edr/collections/observations/position?datetime=2022-07-22T04:10:00Z/2022-07-22T06:10:00Z&coord=POINT(3.275 51.997777777777778)¶meter-name=L_dryb_10,u_10&crs=WGS84)

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> datetime	2022-07-22T04:10:00Z/2022-07-22T06:10:00Z	
<input checked="" type="checkbox"/> coord	POINT(3.275 51.997777777777778)	
<input checked="" type="checkbox"/> parameter-name	L_dryb_10,u_10	
<input checked="" type="checkbox"/> crs	WGS84	
<input type="checkbox"/> t	application/prs.coverage+json	
Key	Value	Descriptor

Body Cookies Headers (13) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "type": "Coverage",
3   "domain": {
4     "type": "Domain",
5     "domainType": "PointSeries",
6     "axes": {
7       "x": {
8         "values": [
9           3.275
10        ]
11      },
12      "y": {
13        "values": [
14          51.997777777777778
15        ]
16      },
17      "t": {
18        "values": [
19          "2022-07-22T04:10:00Z",
20          "2022-07-22T04:20:00Z",
21          "2022-07-22T04:30:00Z",
22          "2022-07-22T04:40:00Z",
23          "2022-07-22T04:50:00Z",
24          "2022-07-22T05:00:00Z",
25          "2022-07-22T05:10:00Z",
26          "2022-07-22T05:20:00Z",
27          "2022-07-22T05:30:00Z",
28          "2022-07-22T05:40:00Z",
29          "2022-07-22T05:50:00Z",
30          "2022-07-22T06:00:00Z",
31          "2022-07-22T06:10:00Z"
32        ]
33      }
34    },
35    "referencing": [
36      {
37        "coordinates": [
38          "y",
39          "x"
```



Getting started

> Materials:

- <https://tinyurl.com/kdp-api-workshop>
- Click on EDR-API

> Request API key

- <https://developer.dataplatform.knmi.nl/register/>
- <https://developer.dataplatform.knmi.nl/apis/>

> Software:

- Python3.8 (or later)
- Postman (<https://www.postman.com/>)

> CoverageJSON playground:

- <https://covjson.org/playground/>

> Documentation:

- <https://developer.dataplatform.knmi.nl/apis/>

> Specification:

- <https://covjson.org/spec/>
- <https://ogcapi.ogc.org/edr/>



Exercises

- 1. Plot temperature at Schiphol for the past 7 days
 - Either in Python
 - Or with Postman in CovJSON Playground
- 2. Calculate and plot monthly rain totals for a location
 - In Python (suggestion: use pandas)



Future developments

- > Stable release of EDR API
 - Bug fixes
 - Adhering to specification
- > Data
 - All observation data since 2003
 - Gridded data (initially 6 datasets)
Temperature and Wind
- > Output
 - More formats (NetCDF, CSV?)





Questions?

opendata@knmi.nl

Annegies van 't Zand
Software Engineer
KNMI Data Platform

annegies.van.t.zand@knmi.nl

Paul van Schayck
Software Engineer
KNMI Data Platform

paul.van.schayck@knmi.nl