

CE HUB API SERVICE DOCUMENTATION

Contents

1	INTRODUCTION	3
2	ACCESS PERMISSION	4
2.1	API KEY	4
3	FORECAST API.....	4
3.1	AUTHORIZATION.....	4
3.2	TYPES OF FORECAST DATA SOURCES	6
3.3	METHODS	6
3.3.1	<i>ShortRangeForecastHourly</i>	6
3.3.2	<i>ShortRangeForecastDaily</i>	9
3.3.3	<i>Nowcast</i>	11
3.4	LIST OF MEASURE LABEL	13
4	HISTORICAL API.....	14
4.1	JSON QUERY STRUCTURE.....	15
4.1.1	<i>Units</i>	15
4.1.2	<i>GeoJSON Geometry</i>	15
4.1.3	<i>Output Format</i>	16
4.1.4	<i>Time Intervals</i>	17
4.1.5	<i>Datasets and Variables</i>	17
4.1.6	<i>List of Variable JSON</i>	18
4.1.7	<i>Useful Links</i>	19

1 Introduction

The objective of **CEHub**, the **Crop Environmental data Hub** is to provide a central access point

- to Syngenta's plant environmental data associated with time and geographical coordinates.
- to global, quality controlled, homogeneous and therefore reusable data sets.

CEHub aims to support any Syngenta department with environmental data supply

- to perform large analysis over environmental data
- to develop digital internal and customer-facing decisions support solutions

There are two different API's:

- **Weather forecast**
API tailored to get 7 or 14-day forecast automatically with the best spatial resolution and most frequent updates.
- **Historical environmental data**
API tailored to get
 - historical weather
 - vegetation health
 - soil characteristic
 - land use
 - topography

2 Access permission

2.1 API key

A key is required for each group to execute API calls.

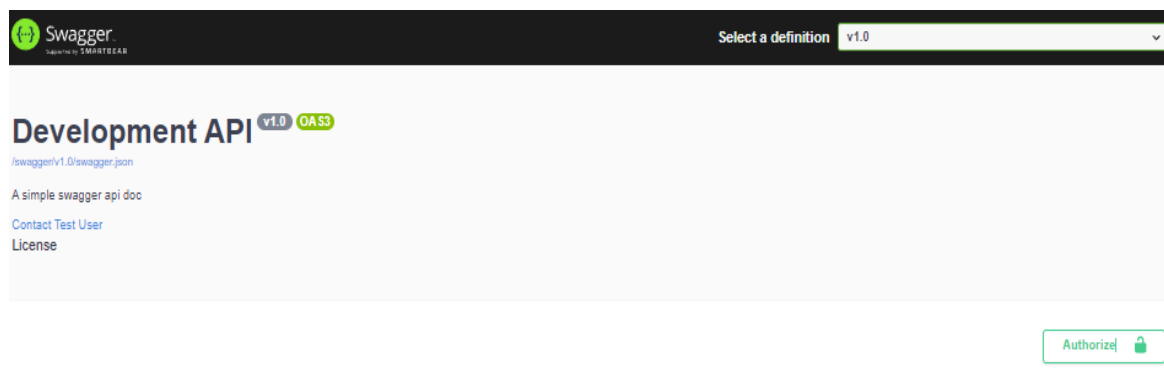
The API keys for forecast and historical will be shared via the onboarding tool.

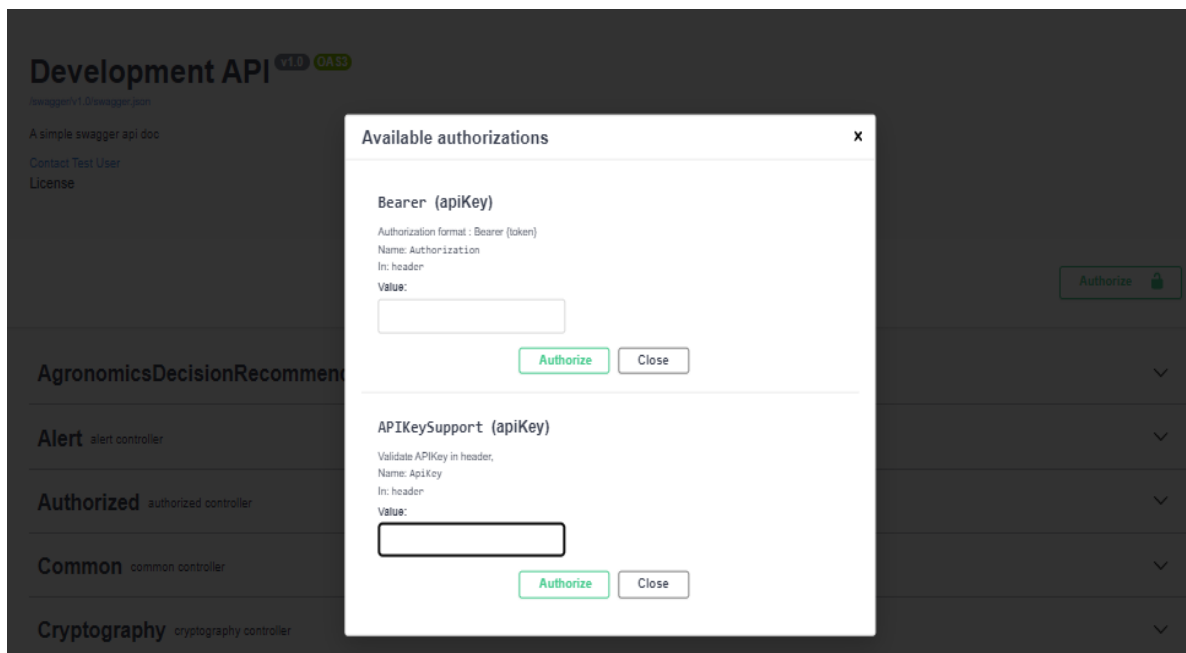
3 Forecast API

Swagger link : <https://services.cehub.syngenta-ais.com/swagger/index.html>

3.1 Authorization

You can click “Authorize” button on swagger home page.



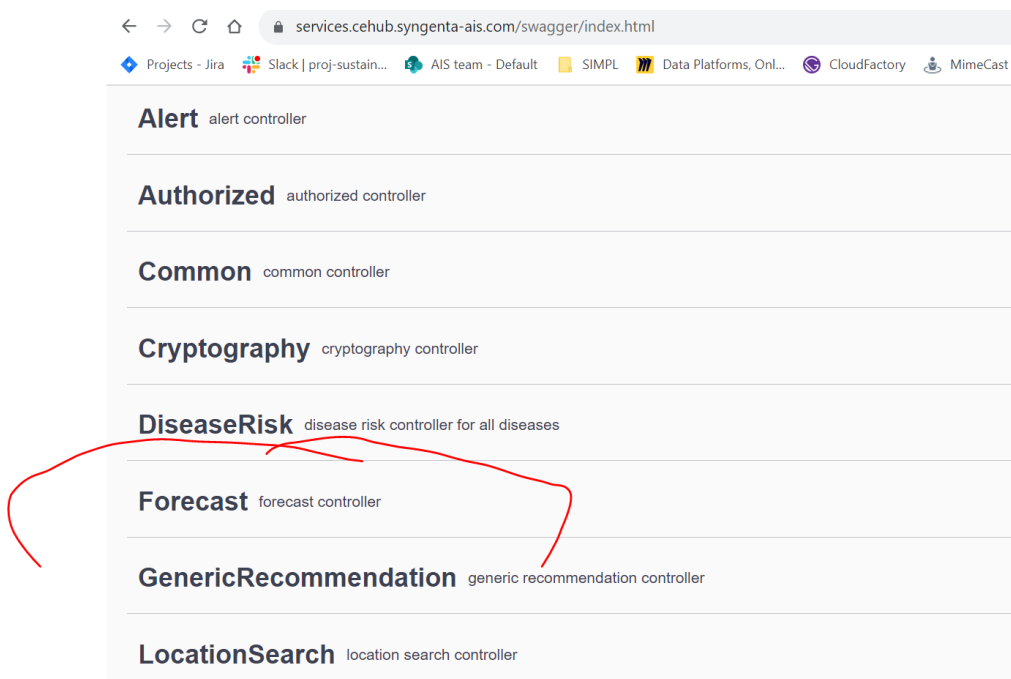


There are 2 ways to Authorized it,

1. Bearer token,
2. Apikey.

kindly use the apikey option which you have obtained from the onboarding tool.

Kindly select the Forecast API's using the forecast section of the swagger as shown below



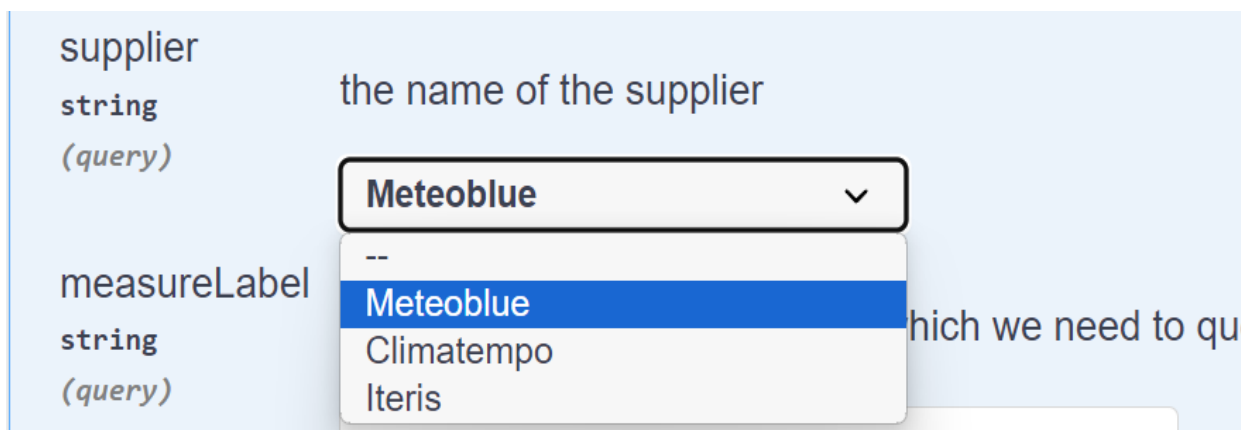
3.2 Types of forecast data sources

We work with various third party providers for getting the weather data and the list of providers are mentioned below.

These are the list of providers that we are currently offering.

providers	Data source team
Meteoblue	Meteoblue
Iteris	Iteris
Climatempo	Climatempo
ClimatempoAgro	Climatempo, but different part

Kindly use **Meteoblue** as the datasource as shown below



3.3 Methods

3.3.1 ShortRangeForecastHourly

Description

- This method is used to generate the forecast hourly measure data i.e. current conditions of the given location.
- Based on the requested time range, the API provides short range forecast with a time interval of 1 hour

- API will work only with GPS co-ordinates i.e. latitude and longitude or WKT (explain in section below) and not with zip code or name of the locations.

Input Parameters

Variable Name	Data Type	Required	Description	Default Value	Optional Values
wkt	String	No	The well-known text, such as: geometrycollection(point(7 42), point(8 20)) or point(7 42)		
latitude	Double	No	The latitude value of the location which want to search		
longitude	Double	No	The longitude of the location which want to search		
startDate	DateTime	No	The start date of the location which want to search		
endDate	DateTime	No	The end date of the location which want to search		
supplier	String	No	Only can be one value of optional values	Meteoblue	Meteoblue Iteris Climatepro
measureLabel	String	No	Contains the measure label for which we need to query, separated by delimiter(,)		List of measure Label can be found in the section 3.4
top	Int	No	It will take all data if top is null, otherwise, take top size data	20	
format	String	No	Only can be one value of optional values	json	json csv

GET /api/Forecast/ShortRangeForecastHourly get forecast hourly data

Parameters

Name	Description
wkt	the well-known text, such as: geometrycollection(point(7 42), point(8 20)) or point(7 42); format: point(lng lat)
latitude	the number of the latitude
longitude	the number of the longitude
startDate	the value of the start date
endDate	the value of the end date
supplier	the name of the supplier
measureLabel	contains the measure label for which we need to query, separated by delimiter(,)
top	
format	

Execute Clear

Sample Url

<http://services.cehub.syngenta-ais.com/api/Forecast/ShortRangeForecastHourly?wkt=point%287%2042%29&startDate=2023-11-23&endDate=2023-11-27&supplier=Meteoblue&top=20&format=json&ApiKey=xxxx>

***kindly change the start date and end date accordingly**

Sample Response

```
[
  {
    "latitude": 42,
    "longitude": 7,
    "offset": 0,
    "date": "2023/11/23 00:00:00",
    "measureLabel": "TempAir_Hourly (C)",
    "value": "16"
  },
  {
    "latitude": 42,
    "longitude": 7,
    "offset": 0,
    "date": "2023/11/23 00:00:00",
    "measureLabel": "TempAir_Surface_Hourly(C)",
    "value": "17.6"
  },
  {
    "latitude": 42,
    "longitude": 7,
    "offset": 0,
    "date": "2023/11/23 00:00:00"
```


3.3.2 ShortRangeForecastDaily

- This method is used to generate the forecast daily measure data i.e. current conditions of the given location.
- Based on the requested time range, the API provides short range forecast with a time interval of 1 day
- API will work only with GPS co-ordinates i.e. latitude and longitude or WKT (explain in section below) and not with zip code or name of the locations.

Input Parameters

Variable Name	Data Type	Required	Description	Default Value	Optional Values
wkt	String	No	The well-known text, such as: geometrycollection(point(7 42), point(8 20)) or point(7 42)		
latitude	Double	No	The latitude value of the location which want to search		
longitude	String	No	The longitude of the location which want to search		
startDate	DateTime	No	The start date of the location which want to search		
endDate	DateTime	No	The end date of the location which want to search		
supplier	String	No	Only can be one value of optional values	Meteoblue	Meteoblue Iteris Climatepro ClimateproAgro
measureLabel	String	No	Contains the measure label for which we need to query, separated by delimiter(,)		List of measure Label can be found in the section 3.4
top	Int	No	It will take all data if top is null, otherwise, take top size data	20	
format	String	No	Only can be one value of optional values	json	json csv

GET /api/Forecast/ShortRangeForecastDaily get forecast daily data

Parameters

Name	Description
wkt string (query)	*Either wkt or latitude and longitude the well-known text, such as: geometrycollection(point(7 42), point(8 20)) or point(7 42); format: point(lng lat)
latitude number(\$double) (query)	*Either wkt or latitude and longitude the number of the latitude
longitude number(\$double) (query)	*Either wkt or latitude and longitude the number of the longitude
startDate string(\$date-time) (query)	the value of the start date
endDate string(\$date-time) (query)	the value of the end date
supplier string (query)	the name of the supplier
measureLabel string (query)	contains the measure label for which we need to query, separated by delimiter(,)
top integer(\$int32) (query)	
format string (query)	

Execute Clear

Sample Url

<http://services.cehub.syngenta-ais.com/api/Forecast/ShortRangeForecastDaily?wkt=point%287%2042%29&startDate=2023-11-23&endDate=2023-11-27&supplier=Meteoblue&top=20&format=json&ApiKey=xxxx>

kindly change the start date and end date accordingly

Sample Response

```
[
  {
    "latitude": 42,
    "longitude": 7,
    "date": "2023/11/23 00:00:00",
    "measureLabel": "TempAir_DailyMin (C)",
    "dailyValue": "15.5"
  },
  {
    "latitude": 42,
    "longitude": 7,
    "date": "2023/11/23 00:00:00",
    "measureLabel": "TempAir_DailyMax (C)",
    "dailyValue": "16.2"
  },
  {
    "latitude": 42,
    "longitude": 7,
    "date": "2023/11/23 00:00:00",
    "measureLabel": "TempAir_DailyAvg (C)",
    "dailyValue": "15.85"
  }
]
```

3.3.3 Nowcast

- This method is used to generate the nowcast i.e. current conditions of the given location.
- Based on the requested time range, the API provides short range forecast with a time interval of 15 mins.
- Currently we support the nowcast data from meteoblue and Iteris.
- API will work only with GPS co-ordinates i.e. latitude and longitude or WKT (explain in section below) and not with zip code or name of the locations.

Input Parameters

Variable Name	Data Type	Required	Description	Default Value	Optional Values
wkt	String	No	the well-known text, such as: geometrycollection(point(7 42), point(8 20)) or point(7 42); format: point(lng lat) <i>*Either wkt or latitude and longitude</i>		
latitude	Double	No	the number of the latitude		
longitude	String	No	the number of the longitude		
startDate	DateTime	No	The start date of the location which want to search		
endDate	DateTime	No	The end date of the location which want to search		
supplier	String	No	Only can be one value of optional values	Meteoblue	Meteoblue Iteris
measureLabel	String	No	Contains the measure label for which we need to query, separated by delimiter(,)		List of measure Label can be found in the section 3.4
top	Int	No	It will take all data if top is null, otherwise, take top size data	20	
format	String	No	Only can be one value of optional values	json	json csv

GET /api/Forecast/Nowcast get nowcast data

Parameters

Name	Description
wkt string (query)	*Either wkt or latitude and longitude the well-known text, such as: geometrycollection(point(7 42), point(8 20)) or point(7 42); format: point(lng lat)
latitude number(\$double) (query)	*Either wkt or latitude and longitude the number of the latitude
longitude number(\$double) (query)	*Either wkt or latitude and longitude the number of the longitude
startDate string(\$date-time) (query)	the value of the start date
endDate string(\$date-time) (query)	the value of the end date
supplier string (query)	the name of the supplier
measureLabel string (query)	contains the measure label for which we need to query, separated by delimiter(,)
top integer(\$int32) (query)	
format string (query)	

Execute Clear

Sample Url

<http://services.cehub.syngenta-ais.com/api/Forecast/Nowcast?wkt=point%287%2042%29&startDate=2023-11-23&endDate=2023-11-27&supplier=Meteoblue&top=20&format=json&ApiKey=xxxx>

kindly change the start date and end date accordingly

Sample Response

```
[
  {
    "latitude": 42,
    "longitude": 7,
    "offset": 0,
    "date": "2023/11/23 00:00:00",
    "measureLabel": "Temperature_15Min (C)",
    "value": "16.01"
  },
  {
    "latitude": 42,
    "longitude": 7,
    "offset": 0,
    "date": "2023/11/23 00:00:00",
    "measureLabel": "TempAirFelt_15Min (C)",
    "value": "4.35"
  },
  {
    "latitude": 42,
    "longitude": 7,
    "offset": 0,
    "date": "2023/11/23 00:00:00",
    "measureLabel": "TempAirFelt_15Min (C)",
    "value": "4.35"
  }
]
```

3.4 List of Measure Label

MeasureType	Label
Nowcast	Temperature_15Min (C)
Nowcast	WindSpeed_15Min (m/s)
Nowcast	WindDirection_15Min
Nowcast	HumidityRel_15Min (pct)
Hourly	Cloudcover_Hourly (pct)
Hourly	GlobalRadiation_HourlySum (Wh/m2)
Hourly	HumidityRel_Hourly (pct)
Hourly	Precip_HourlySum (mm)
Hourly	PrecipProbability_Hourly (pct)
Hourly	ShowerProbability_Hourly (pct)
Hourly	SnowFraction_Hourly
Hourly	SunshineDuration_Hourly (min)
Hourly	TempAir_Hourly (C)
Hourly	Visibility_Hourly (m)
Hourly	WindDirection_Hourly (Deg)
Hourly	WindGust_Hourly (m/s)
Hourly	WindSpeed_Hourly (m/s)
Hourly	Soilmoisture_0to10cm_Hourly (vol%)
Hourly	Soiltemperature_0to10cm_Hourly (C)
Hourly	Referenceevapotranspiration_HourlySum (mm)
Hourly	LeafWetnessProbability_Hourly (pct)
Daily	Cloudcover_DailyAvg (pct)
Daily	Evapotranspiration_DailySum (mm)
Daily	GlobalRadiation_DailySum (Wh/m2)
Daily	HumidityRel_DailyAvg (pct)
Daily	HumidityRel_DailyMax (pct)
Daily	HumidityRel_DailyMin (pct)
Daily	Precip_DailySum (mm)
Daily	PrecipProbability_Daily (pct)
Daily	ShowerProbability_DailyMax (pct)
Daily	SnowFraction_Daily (pct)
Daily	SunshineDuration_DailySum (min)
Daily	TempAir_DailyAvg (C)
Daily	TempAir_DailyMax (C)
Daily	TempAir_DailyMin (C)
Daily	ThunderstormProbability_DailyMax (pct)
Daily	WindDirection_DailyAvg (Deg)
Daily	WindGust_DailyMax (m/s)

Daily	WindSpeed_DailyAvg (m/s)
Daily	WindSpeed_DailyMax (m/s)
Daily	WindSpeed_DailyMin (m/s)
Daily	WindDirection_DailyAvg
Daily	Soilmoisture_0to10cm_DailyMax (vol%)
Daily	Soilmoisture_0to10cm_DailyAvg (vol%)
Daily	Soilmoisture_0to10cm_DailyMin (vol%)
Daily	Soiltemperature_0to10cm_DailyMax (C)
Daily	Soiltemperature_0to10cm_DailyAvg (C)
Daily	Soiltemperature_0to10cm_DailyMin (C)
Daily	Referenceevapotranspiration_DailySum (mm)

4 Historical API

The Historical API grants access to the entire weather data archive, which is comprised of various weather variables gathered from over 50 data sources.

URL : <http://my.meteoblue.com/dataset/query?apikey=APIKEY>

Kindly replace APIKEY value with the key obtained from the onboarding tool.

We cannot use the GET parameters is not sufficient to query datasets dynamically. Instead a HTTP JSON request body is used:

```
{
  "units": {
    "temperature": "C",
    "velocity": "km/h",
    "length": "metric",
    "energy": "watts"
  },
  "geometry": {
    "type": "MultiPoint",
    "coordinates": [[7.57327,47.558399,279]], // lon, lat, asl
    "locationNames": ["Basel"]
  },
  "format": "json",
  "timeIntervals": [
    "2019-01-01T+00:00/2019-12-31T+00:00"
  ],
  "queries": [{
    "domain": "NEMSGLOBAL",
    "gapFillDomain": null,
    "timeResolution": "hourly",
    "codes": [{
      "code": 157,
      "level": "180-0 mb above gnd"
    }]
  }]
}
```

```
}]
```

4.1 JSON Query Structure

The JSON body uses various structures and arrays that are nested to build complex queries with recursive transformations. All JSON attributes are case-sensitive and use camel-case names. As in the example above, the outer JSON structure contains properties like `units`, `geometry`, `timeIntervals` or `queries`.

Property	Type	Description
<code>units</code>	Structure: Units	Option to select units like Fahrenheit
<code>geometry</code>	Structure: GeoJSON	Select polygon or points
<code>format</code>	String enumeration: Format	Which output format to use
<code>timeIntervals</code>	Array of Structure: TimeInterval	Define time intervals to read
<code>queries</code>	Array of Structure: Query	Per dataset queries

4.1.1 Units

If units are not set, the defaults are Celsius, km/h, metric and watts

Property	Type	Description
<code>temperature</code>	String	c(Celsius) or f (Fahrenheit)
<code>velocity</code>	String	km/h, m/s , mph, kn or bft
<code>length</code>	String	metric or imperial
<code>energy</code>	String	watts or joules

4.1.2 GeoJSON Geometry

Please make sure to provide all input coordinates in the correct order: "lon" -> "lat" (-> "asl")

The `geometry` structure is based on GeoJSON, but extended to support features like location names and additional attributes.

Coordinates are defined as tuple of longitude, latitude and elevation above sea level. Elevation is optional and will be automatically resolved from an 80 m resolution digital elevation model (DEM). `locationNames` can be optionally specified and will be replicated in the output.

For Polygon and multiPolygon, The first and last coordinate must be the same. Please make sure to supply a valid polygon without self-intersections.

The sample GeoJSON body for

Point

```
{
  "type": "Point",
  "coordinates": [8.6, 47.5, 351.1] // lon, lat, asl
}
```

MultiPoint

```
{
  "type": "MultiPoint",
  "coordinates": [[8.6, 47.5, 351.1], [8.55, 47.37, 429]], // lon, lat, asl
  "locationNames": ["Basel", "Zürich"]
}
```

Polygon

```
{
  "type": "Polygon",
  "coordinates": [
    [[7.5, 47.5], [7.5, 47.6], [7.7, 47.6], [7.7, 47.5], [7.5, 47.5]] // lon, lat
  ]
}
```

MultiPolygon

```
{
  "type": "MultiPolygon",
  "coordinates": [
    [[8.0, 47.4], [8.0, 47.6], [8.2, 47.6], [8.2, 47.4], [8.0, 47.4]], // lon, lat
    [[7.5, 47.5], [7.5, 47.6], [7.7, 47.6], [7.7, 47.5], [7.5, 47.5]] // lon, lat
  ],
  "excludeSeaPoints": true,
  "fallbackToNearestNeighbour": true
}
```

The optional Boolean parameter `excludeSeaPoints` can be set to `true`, to ignore grid-cells that are located on the sea. If no grid-cells are within the polygon, the result would be empty.

If `fallbackToNearestNeighbour` is set to `true`, the result will select the nearest neighbour grid-cell instead.

4.1.3 Output Format

The attribute `format` accepts the following values:

- `json`: Recommended JSON format (default, if not set)
- `csv`: CSV format for large amount of locations
- `csvTimeOriented`: CSV format for long time-ranges
- `csvIrregular`: CSV format for mixed time-intervals and locations

- geoJson: JSON output to create map with bullet points
- geoJsonHtml: HTML page that embeds a map library and the map json
- netCDF: Recommended binary format for further scientific data analysis

4.1.4 Time Intervals

Time intervals and timezones can be specified using the ISO8601 format. The timeIntervals attribute is an array of ISO8601 strings. Per default the web-interfaces generate time-intervals with a timezone offset, but without specifying the hour and minute.

```
{
  "timeIntervals": [
    "2015-05-01T+00:00/2015-05-02T+00:00",
    "2016-05-01T+00:00/2016-05-02T+00:00"
  ]
}
```

In the intervals above, 2 full days are selected. For hourly data, the API would return 48 hourly values for each time interval. In the API syntax time-intervals could be specified to select exactly 1 hour:

```
{
  "timeIntervals": [
    "2019-01-01T00:00+00:00/2019-01-01T01:00+00:00"
  ]
}
```

4.1.5 Datasets and Variables

The selection of datasets and variables is specified in the attribute queries as an array to select multiple datasets. For each dataset, specified by the domain attribute, multiple weather variable codes can then be chosen.

In this example, three variables are selected from ERA5T and then transformed with two transformations. In the same call, data can be selected from the dataset NEMS12 and transformed individually.

```
{
  "queries": [
    {
      "domain": "ERA5T",
      "gapFillDomain": "NEMSGLOBAL",
      "timeResolution": "hourly",
      "codes": [
        {"code": 11, "level": "2 m above gnd"},
        {"code": 52, "level": "2 m above gnd"},
        {"code": 157, "level": "180-0 mb above gnd"}
      ],
      "transformations": [
        {
          "type": "aggregateDaily",
          "aggregation": "mean"
        }
      ]
    },
    {
      "domain": "NEMS12",
      "gapFillDomain": null,
      "codes": [ ... ]
    }
  ]
}
```

```

    "transformations": [ ... ]
  }
]
}

```

Attributes for the structure query:

Property	Type	Description
domain	String	ERA5T (kindly use this dataset)
gapFillDomain	Optional String	NEMSGLOBAL (kindly use this dataset, dataset to use to fill gaps).
timeResolution	String	hourly or daily
codes	Array of Codes	Individual selection of weather variables.
transformations	Optional array of transformations	

Notes:

- timeResolution specified the resolution to read. It can also be set to **daily** although the dataset only offers hourly data to calculate daily aggregations automatically. Aggregations like **monthly** must use transformations.

Once the dataset has been selected, multiple variables at different levels can be encoded into the call. The API is capable of assigning multiple variables per dataset at once. This could improve API call performance because expensive spatial calculations are only performed once.

Attributes for the structure code:

Property	Type	Description
code	Integer	Numeric variable code. E.g. 11 for temperature
level	String	Level the variable. E.g. 2 m above gnd
aggregation	Optional String	min, max, mean, sum to be used with daily aggregations

4.1.6 List of Variable JSON



CE%20Hub_variables
_JSON_request.xlsx

4.1.7 Useful Links

The detailed API Documentation : <https://docs.meteoblue.com/en/weather-apis/dataset-api/dataset-api>

Python SDK : <https://github.com/meteoblue/python-dataset-sdk>