

# **Weather API 1.3.0**

## **Release note**

May 15<sup>th</sup>, 2019

# Index

## Feature extensions

New main endpoint Point Forecast Service 3

Additional gap filling parameters 5

## Feature extensions (internal)

API monitoring dashboards and alarms 6

## Hardening

Load test environment set up 7

## Feature extensions

### New main endpoint Point Forecast Service

A new version of the point Forecast Service has been developed which will be the default used version from now on. Its purpose is to provide the forecast data of MOS via the API as consistent as possible when compared to the original MOS data and how it's being used in feeds.

Key differences with previous default version:

New default version	Former version
<a href="https://point-forecast.weather.mg/forecast/hourly">https://point-forecast.weather.mg/forecast/hourly</a>	<a href="https://point-forecast.weather.mg/search">https://point-forecast.weather.mg/search</a>
Uses exact MOS data as input	Has derived method for some elements like minimum and maximum temperature
Uses default time periods as in MOS and according to WMO standards	Uses more and alternative time periods than MOS and WMO standards
Uses UTC time zone as default	Applies local time off-set to UTC and therefore provides non-standard time periods roughly corrected to local day/night periods

The new version will be documented as default as it is consistent with both MOS and WMO regulations. This also ensures better consistency with feeds enabling smoother migration towards API.

The deprecated endpoint can still be used by existing customers but should not be actively promoted to new users. This functionality might only be used after consultation when customers cannot cope with standard reporting times and want MeteoGroup to solve the complexity of translating a measured value to the best-fitting local time (night/day relevant periods).

Below a specific example is described using the new (/hourly) and old (/search) endpoints.

<p>Example query new endpoint:</p> <p><a href="https://point-forecast.weather.mg/forecast/hourly?meteoGroupStationIds=12566&amp;validUntil=2019-05-05T12:00:00.000Z&amp;timeZone=UTC&amp;validFrom=2019-05-04T00:00:00.000Z&amp;fields=maxWindGustInKilometerPerHour&amp;validPeriod=PT12H">https://point-forecast.weather.mg/forecast/hourly?meteoGroupStationIds=12566&amp;validUntil=2019-05-05T12:00:00.000Z&amp;timeZone=UTC&amp;validFrom=2019-05-04T00:00:00.000Z&amp;fields=maxWindGustInKilometerPerHour&amp;validPeriod=PT12H</a></p>	<p>Example query old endpoint:</p> <p><a href="https://point-forecast.weather.mg/search?meteoGroupStationIds=12566&amp;validUntil=2019-05-05T12:00:00.000Z&amp;timeZone=UTC&amp;validFrom=2019-05-04T00:00:00.000Z&amp;fields=maxWindGustInKilometerPerHour&amp;validPeriod=PT12H">https://point-forecast.weather.mg/search?meteoGroupStationIds=12566&amp;validUntil=2019-05-05T12:00:00.000Z&amp;timeZone=UTC&amp;validFrom=2019-05-04T00:00:00.000Z&amp;fields=maxWindGustInKilometerPerHour&amp;validPeriod=PT12H</a></p>
<p>Response includes:</p> <pre>"meteoGroupStationId": "12566",   "validFrom": "2019-05-04T06:00:00Z",   "validUntil": "2019-05-04T18:00:00Z",   "validPeriod": "PT12H",   "maxWindGustInKilometerPerHour": 24.4</pre>	<p>Response includes:</p> <pre>"meteoGroupStationId": "12566",   "validFrom": "2019-05-04T04:00:00Z",   "validUntil": "2019-05-04T16:00:00Z",   "validPeriod": "PT12H",   "maxWindGustInKilometerPerHour": 24.4</pre>

In the query request the only difference is the endpoint being requested. As all documentation will be pointing towards the /hourly endpoint the new customers will automatically be using this.

In the response the values are the same but the difference is the time for which it is given ("valid from" and "valid until"). The new endpoint gives the standard UTC times (as in MOS) while the old endpoint provides the time zone in UTC as well but corrected for the local time offset from UTC. For this station (Krakow, Poland) the difference between UTC and local time is 2 hours which is reflected in the response.

In the new endpoint parameters which normally are presented for day/night periods like minimum/maximum temperature, will be presented for 06/18 UTC only in Europe and every 6 hours elsewhere globally. This is standard in MOS as well.

## Additional gap filling parameters

MeteoGroup's Point Observation API delivers high-quality weather observation data from more than 20,000 stations world-wide. Sometimes recent observations are missing, especially for specific parameters while instruments can be broken or out for maintenance. As the input of observations can be critical for business processes we provide the functionality to receive a pseudo-observation when real observations are missing.

The following parameters have been added to the list of gap-filled parameters for period PT1H:

- globalRadiationInJoulePerSquareCentimeter
- averageGlobalRadiationInWattPerSquareMeter
- precipitationAmount (in Inch and millimeter)

Gap-filled data is labelled in the API response. This functionality can be used when an enterprise package is bought.

The gap-filling method is always applied consistently using the +0 forecast period of our advanced MOS forecasting system. This is the most likely observed weather using the most recent observations and forecasts from the station itself and surrounding stations.

For the gap-filling feature to work, you must supply the following items in your API call:

- gapFilling. This is a Boolean parameter that, when set to true, activates the gap-filling feature for the API request.
- gapFillingInfo. This is an optional field to retrieve information about what parameters in the response were actually gap-filled. To get gap-filled field information, add this item for the fields parameter in the API call.

A URL example with the added parameters would look like this:

```
https://point-observation.weather.mg/observation/forecast/hourly?  
meteoGroupStationIds=3010390&observedPeriod=PT1H&fields=meteoGroupStationId,globalRadiationInJoulePerSquareCentimeter,averageGlobalRadiationInWattPerSquareMeter,gapFillingInfo&observedFrom=2019-04-22T00:00:00Z&observedUntil=2019-04-22T01:00:00Z&gapFilling=true
```

In the response, gap filling returns the gapFillingInfo object that contains the following items (highlighted in red in the sample response below):

- The dataSource string. It contains the data source—forecast.
- The issuedAt string. It contains the timestamp of the forecast in the [standard ISO 08601 notation](#).
- The gapFilledFields array. It shows which weather parameters have been processed using the gap-filling feature. To see the specific values for these weather parameters, look at the list of weather parameters returned in the response (highlighted in red).

The response for the URL example given is:

Response example:

```
{
```

```
"observations": [{
  "meteoGroupStationId": "3010390"
}, {
  "meteoGroupStationId": "3010390",
  "observedFrom": "2019-04-22T01:00:00+02:00",
  "observedUntil": "2019-04-22T02:00:00+02:00",
  "observedPeriod": "PT1H",
  "gapFillingInfo": {
    "dataSource": "Forecast",
    "dataSourceUrl": "https://point-forecast.weather.mg/search?validPeriod=PT1H&fields=averageGlobalRadiationInWattPerSquareMeter,globalRadiationInJoulePerSquareCentimeter,issuedAt&meteoGroupStationIds=3010390&validFrom=2019-04-22T00:00:00Z&validUntil=2019-04-22T02:00:00Z",
    "issuedAt": "2019-04-22T02:25:00.000Z",
    "gapFilledFields": ["averageGlobalRadiationInWattPerSquareMeter", "globalRadiationInJoulePerSquareCentimeter"]
  },
  "globalRadiationInJoulePerSquareCentimeter": 0.0,
  "averageGlobalRadiationInWattPerSquareMeter": 0.0
}, {
  "meteoGroupStationId": "3010390",
  "observedFrom": "2019-04-22T02:00:00+02:00",
  "observedUntil": "2019-04-22T03:00:00+02:00",
  "observedPeriod": "PT1H",
  "gapFillingInfo": {
    "dataSource": "Forecast",
    "dataSourceUrl": "https://point-forecast.weather.mg/search?validPeriod=PT1H&fields=averageGlobalRadiationInWattPerSquareMeter,globalRadiationInJoulePerSquareCentimeter,issuedAt&meteoGroupStationIds=3010390&validFrom=2019-04-22T00:00:00Z&validUntil=2019-04-22T02:00:00Z",
    "issuedAt": "2019-04-22T02:25:00.000Z",
    "gapFilledFields": ["averageGlobalRadiationInWattPerSquareMeter", "globalRadiationInJoulePerSquareCentimeter"]
  },
  "globalRadiationInJoulePerSquareCentimeter": 0.0,
  "averageGlobalRadiationInWattPerSquareMeter": 0.0
}
}]
}
```

# FEATURE EXTENSIONS (INTERNAL)

## API MONITORING DASHBOARDS AND ALARMS

### Dashboard set up

An improved dashboard for continuous monitoring of the main product API's has been set up, supporting optimal 24x7 support and early warning notification. Key processes were listed, thresholds were defined and visual boards were defined showing these processes and thresholds in 1 overview.



### Alarm settings

Based on the chosen key processes and thresholds alarms were defined with corresponding actions required to ensure early notice of potential issues and swift analysis and follow up once issues occur. Alarms and procedures are renewed, optimizing the operational 24x7 support environment.

# Hardening

## Load tests environment set up

A new load test environment was set up to enable better analysis on performance and scaling of Point Observation and Point Forecast Services. By mirroring the production environment for these services and supporting Station Metadata and Elevation Services we are able to better prepare for future scaling of the API's.