

meteostat API

Data collected by meteostat is available through our JSON API. This interface gives developers of websites, apps and other projects access to weather statistics and climate data provided by meteostat. The API is generally free of charge. However, access is limited to 200 calls per key and hour. Once this limit is reached, all of your requests will result in HTTP 403, forbidden, until your quota is cleared.

As the meteostat API is provided at no charge, we're limiting the total amount of requests to our historical data archive. If the traffic on our API platform gets too high, we will start rejecting requests with HTTP 503. In that case, please wait a few seconds and try again.

The meteostat API is a free service that powers the meteostat web application and multiple third-party projects. We cannot guarantee for the availability of this service. Users of the meteostat API must follow the [terms of service](#) and copyright guidelines. **The meteostat API is limited to non-commercial usage, only.** By using the meteostat API you agree to these terms.

Follow [meteostat on Twitter](#) for updates and [join our Slack workspace](#) for support & discussion.

Introduction

The meteostat API makes historical weather data and climate normals available for everyone. In order to query data you will need to apply for an [API key](#). The format of the data output is JSON. You can parse JSON data using the programming language of your choice.

The API provides different methods. Parameters are passed to a method via the HTTP request in order to specify the query. Lets take a look at the URL structure.

URL structure

```
https://api.meteostat.net/{VERSION}/{PACKAGE}/{METHOD}?{PARAMETERS}
```

It depends on the method which parameters you have to define. In every case you must submit your API key using the **key** parameter. The parameters required by each method are listed in the respective manuals.

Currently, all API methods are accessible using HTTP GET requests.

Debug Mode

By setting the **debug** parameter to **1** you can retrieve a pretty-printed data output that contains additional debugging information.

Example: Calling 2017 Weather Summary for Frankfurt Airport

```
https://api.meteostat.net/v1/history/daily?station=10637&start=2017-01-01&end=2017-12-31&key=XXXXXXXX
```

In this example *history* is the package and *monthly* is the method. The parameters are *station*, *start* and *end*. Whenever you have to define a time range, *start* defines the start date and *end* defines the end date. Depending on the method you can provide dates in different formats:

- A year (e.g. *2009*)
- A month (e.g. *2009-01* for *January 2009*)
- A date (e.g. *2009-01-31* for *January 31st 2009*)
- A UNIX timestamp

Remember to replace the *key* placeholder with your personal API key.

Methods

The table shows all methods which are available through the meteostat API.

Package	Method	Description	Parameter(s)
stations	search	Search for weather stations by name, country or geographical specifications	<i>q</i> <i>country</i>
stations	meta	Retrieve meta information for any weather station using its meteostat, WMO, ICAO or IATA identifier	<i>id</i> <i>wmo</i> <i>icao</i> <i>iata</i>
history	monthly	Returns monthly history data for a specific weather station	<i>station</i> <i>start</i> <i>end</i>
history	daily	Returns daily weather statistics for a specific weather station	<i>station</i> <i>start</i> <i>end</i>
history	hourly	Returns hourly observations for a specific weather station	<i>station</i> <i>start</i> <i>end</i> <i>time_zone</i> <i>time_format</i> <i>exclude_model</i>
climate	normals	Returns climate normals for a weather station	<i>station</i>

Formats and Units

meteostat uses the metric system. All methods return values according to the formats and units listed below.

Time Format

Date and time statements follow the ISO 8601 standard (e.g. 2016-12-31 for December 31st 2016 / 23:59:58 for 23 hours, 59 minutes, and 58 seconds). The time zone used by meteostat is Coordinated Universal Time (UTC).

JSON Structure

All API methods return a JSON object that contains two properties: *meta* and *data*. The *meta* object provides general information about the data output and debugging information. For instance, methods which belong to the *history* and *climate* packages provide a *source* variable that holds the names of the organizations which provided the raw data. The *data* property is either an object or an array that contains the actual data output.

Measurement Formats

Data type(s)	Unit
Temperature	°C
Precipitation	mm
Sunshine Duration	hours
Air Pressure	hPa
Wind Speed, Peak Wind Gust	km/h
Wind Direction	degrees
Visibility, Cloud Height	m
Relative Humidity	%

Weather stations

- [Search for weather stations by name](#)
- [Retrieve meta information for a specific weather station](#)
- [Finding weather stations by geo location](#)

stations / search

Many API methods require a weather station identifier. Most weather stations are identified by their WMO ID. However, some weather stations may not be listed in the WMO index. Therefore, these weather stations are given five-digit IDs containing both numbers and letters. You'll need to call the *search* method of the *stations* package in order to get the ID of a weather station. Optionally, you can provide an ISO 3166-1 alpha-2 country code (e.g. *DE* for *Germany*) using the `country` parameter.

Parameter(s): `q` , `country`

Instead of the *stations / search* method you can use a copy of meteostat's weather station index provided in [CSV format](#).

Example: Search results for "toronto" using *stations / search* method

```
https://api.meteostat.net/v1/stations/search?q=toronto&key=XXXXXXXX
```

The request returns four results:

- Toronto City
- Toronto Pearson Airport
- Toronto Island
- Toronto / Buttonville

This is the JSON output:

```
{
  "meta": {
  },
  "data": [
    {
      "id": "71508",
      "country": "CA",
      "name": "Toronto City"
    },
    {
      "id": "71624",
      "country": "CA",
      "name": "Toronto Pearson Airport"
    },
    {
      "id": "71617",
      "country": "CA",
      "name": "Toronto Island"
    },
    {
      "id": "71615",
      "country": "CA",
      "name": "Toronto / Buttonville"
    }
  ]
}
```

Now you can use the `id` variable of your weather station of choice and pass it to another method.

stations / meta

This method provides meta information for each weather station. The output includes details like latitude, longitude and elevation.

Additionally, if you want to know which data is available for the weather station of your interest, you may set the `inventory` parameter to `1`.

Parameter(s): `station` or `wmo` or `icao` or `iata` , `inventory`

Example: Meta information for Toronto / The Annex (ID: 71508) with inventory

```
https://api.meteostat.net/v1/stations/meta?station=71508&inventory=1&key=XXXXXXXX
```

stations / nearby

The *stations / nearby* method enables users to find weather stations by geo location. When calling this method make sure to provide valid geographic coordinates (decimal degrees) using the `lat` (latitude) and `lon` (longitude) parameters. The `limit` parameter allows us to specify the maximum number of weather stations returned by this method.

Parameter(s): `lat` , `lon` , `limit`

Example: Five weather stations nearby Singapore (`lat` = 1.283 and `lon` = 103.83)

```
https://api.meteostat.net/v1/stations/nearby?  
lat=1.283&lon=103.83&limit=5&key=XXXXXXXX
```

Weather History

- [Hourly Records](#)
- [Daily Averages](#)
- [Monthly Statistics](#)

history / hourly

The *history / hourly* method returns hourly weather records based on synoptical observations and METAR data. Additionally, gaps in the time series are filled with statistically optimised [MOSMIX model data](#).

The time zone of all hourly records is Coordinated Universal Time (UTC). The datasets are ordered by time.

Parameter(s): `station` , `start` (date, e.g. 2017-01-01) and `end` (date, e.g. 2017-01-31),
`time_zone` , `time_format` , `exclude_model`

The `time_zone` parameter accepts a time zone as defined by the IANA time zone database ([tz database](#)). Each dataset comes with a `time` parameter that holds the time of the observation in UTC. The `time_local` parameter provides the observation time according to the time zone defined using the `time_zone` parameter. The `time_format` parameter can be used to define the format of the `time` and `time_local` parameters.

In case you want to exclude MOSMIX model data from the API output just set the `exclude_model` parameter to `1`.

Example: Hourly weather data for London Heathrow Airport

```
https://api.meteostat.net/v1/history/hourly?station=03772&start=2019-05-02&end=2019-05-11&time_zone=Europe/London&time_format=Y-m-d%20H:i&key=XXXXXXXX
```

Each dataset has the following structure:

```
{
  "time": "2019-05-01 23:00:00",
  "time_local": "2019-05-02 00:00",
  "temperature": 12.2,
  "dewpoint": 7.9,
  "humidity": 75,
  "precipitation": 0.1,
  "precipitation_3": null,
  "precipitation_6": null,
  "snowdepth": null,
  "windspeed": 9.3,
  "peakgust": 16.7,
  "winddirection": 270,
  "pressure": 1016,
  "condition": 4
}
```

history / daily

Daily data provided by the meteostat API is heavily based on the Global Historical Climatology Network (GHCN). Additionally, daily statistics may include estimated data based on hourly observations (SYNOP and METAR data).

Parameter(s): `station` , `start` (date, e.g. 2017-01-01) and `end` (date, e.g. 2017-01-31)

Example: Calling 2017 Weather Summary for Frankfurt Airport

```
https://api.meteostat.net/v1/history/daily?station=10637&start=2017-01-01&end=2017-12-31&key=XXXXXXXX
```

history / monthly

The *history / monthly* method returns monthly climate summaries. You can query data for any time range from one month to 30 years.

Parameter(s): `station` , `start` (e.g. 2009-01), `end` (e.g. 2009-12)

Example: Calling 2016 Weather Summary for Whitehorse, Yukon

```
https://api.meteostat.net/v1/history/monthly?station=71964&start=2009-01&end=2009-12&key=XXXXXXXX
```

Climate Data

climate / normals

The *climate / normals* method returns multiannual climate normals for any weather station. In order to retrieve data you'll need to define the weather station identifier using the `station` parameter.

Parameter(s): `station`

Example: Climate Normals for Frankfurt Airport, Germany

```
https://api.meteostat.net/v1/climate/normals?station=10637&key=XXXXXXXX
```

This is the JSON output:

```
{
  "meta": {
    "source": "National Oceanic and Atmospheric Administration, Deutscher Wetterdienst"
  },
  "data": {
    "temperature": {
      "JAN": "0.7",
      "FEB": "1.8",
      "MAR": "5.2",
      "APR": "9.2",
      "MAY": "13.7",
      "JUN": "17.1",
      "JUL": "18.9",
      "AUG": "18.3",
      "SEP": "14.8",
      "OCT": "10.2",
      "NOV": "5.7",
      "DEC": "1.2"
    }
  }
}
```

Terms of Service

The meteostat API can be used by non-commercial projects for free. Currently, we cannot offer meteostat data for commercial purposes due to WMO resolution 40. Furthermore, meteostat cannot guarantee for the availability of this service. The API may be shut down at any time without prior notification. Please make sure to mention all data sources which are used by meteostat. We recommend you to use this HTML code:

```
Data provided by <a href="https://www.meteostat.net" title="meteostat"
target="_blank">meteostat</a>. Meteorological data: Copyright &copy; National
Oceanic and Atmospheric Administration (NOAA), Deutscher Wetterdienst (DWD). Learn
more about the <a href="https://www.meteostat.net/sources" title="meteostat
Sources" target="_blank">sources</a>.
```

Data provided by [meteostat](https://www.meteostat.net). Meteorological data: Copyright © National Oceanic and Atmospheric Administration (NOAA), Deutscher Wetterdienst (DWD). Learn more about the [sources](https://www.meteostat.net/sources).

meteostat uses data of various meteorological services and organizations. Each of them has its own copyright restrictions. Please inform yourself about the provisions of the different organizations. meteostat is not responsible for any infringement of a copyright.

Copyright © meteostat. [Legal Disclosure](#) & [Privacy](#).