API Version 1.4 2024-08-05

SPARROW

Measurements API

Environmental monitoring and management turnkey solutions for Cities

Introduction

Get Measurements
- Endpoint URL
- Request Parameters
- Successful JSON Response
- Unsuccessful JSON Response
- Available Filters
- Datetime Format
- Result Codes
- Request Example
- Successful Response Example
- Unsuccessful Response Example

SPARROW

ITU ASSOCIATE

UNEP

SWISS MADE

Introduction

The Measurements API is a web service which allows you to request and retrieve environmental measurements data collected by Sparrow nodes fleet, separated by four categories:

1. Fine particulate matter: PM_1, $PM_{2.5}$, PM_{10}
2. Gases: CO_2, NO_2, CO, O_3
3. Weather conditions: temperature, pressure, humidity
4. Road quality: IRI, road bumps

The API is using REST-Like operations over HTTP GET requests with parameters encoded into the URL address. The response is encoded in JSON format, using HTTP status codes to verify successful data retrieval. It is designed to be easy to use and implement in any third-party software or service that is using web requests for data interchange. Web service is using a scalable cloud infrastructure to be responsive and sustainable for numerous simultaneous connections.

Data from the nodes is constantly transferred to the cloud database and available for retrieval in real-time. Please, notice: geographical location from node GPS module is mandatory for Measurements API, so in case of node location is unavailable due to any reasons including but not limited to poor GPS signal – measurements data will not be stored and available.

> Please, note that dates and times in the API are always in GMT (UTC +0) timezone.

> In order to use the Measurements API you must receive a private authentication API key attached to one or several Sparrow nodes. Please make sure to keep the API key safe and unexposed to the public.

Get Measurements

Endpoint URL

`GET` `https://api.sparrow.city/get/`

Request Parameters

All parameters are required.
Please, notice: period between starting and ending datetimes should not exceed 31 days.

Parameter	Type	Description
api_key	String	Private API key for authorization
filter	String	Measurements filter (see Available Filters)
start_date	String	Period starting datetime (see Datetime Format)
end_date	String	Period ending datetime (see Datetime Format)
start_lat	Float	Area starting latitude
start_lon	Float	Area starting longitude
end_lat	Float	Area ending latitude
end_lon	Float	Area ending longitude

Successful JSON Response

Parameter	Type	Description
result	Integer	Result code (see Result Codes)
body	Array	Array of measurement objects
↳ i	Integer	Measurement identification number
↳ t	Integer	Measurement timestamp in Unix time format
↳ n	String	Measurement node name

↳ x	Float	Measurement longitude
↳ y	Float	Measurement latitude
↳ s	Float	Node detected speed (km/h)
↳ a	Float	Node detected altitude over sea level (m)
↳ v	Float	Measurement value for selected filter

Unsuccessful JSON Response

Parameter	Type	Description
result	Integer	Result code (see Result Codes)
message	String	Short error description

Available Filters

Filter	Description
pm1	Fine particulate matter with a size of 1 µm (µg/m^3)
pm25	Fine particulate matter with a size of 2.5 µm (µg/m^3)
pm10	Fine particulate matter with a size of 10 µm (µg/m^3)
co2	Carbon dioxide concentration (ppm)
no2	Nitrogen dioxide concentration (ppb)
co	Carbon monoxide concentration (ppb)
o3	Ozone concentration (ppb)
temperature	Air temperature (°C)
humidity	Relative humidity (%)
pressure	Atmospheric pressure (hPa)
iri	International road roughness index (m/km)
bumps	Relative level of road bumps (lvl)

Datetime Format

Datetime is using an ISO-8601 format (YYYY-MM-DDThh:mm:ss). It is a combination of date and time separated by the “T” literal. Example, for the noon of the 31st of January 2022:

```
2022-01-31T12:00:00
```

Result Codes

Result Code	Description
200	Successful request
400	Bad request
401	Unauthorized
403	Forbidden
404	Not found
405	Method not allowed
429	Too many requests
500	Internal server error
501	Not implemented
503	Service unavailable

Request Example

```
$ curl -X GET -H 'Accept-encoding: gzip'
'https://api.sparrow.city/get/?filter=pm10&start_date=2023-01-08T12:00:00&end_date=2
023-01-10T15:00:00&start_lat=46.20391241&start_lon=6.145815253&end_lat=46.20585782&e
nd_lon=6.154720187&api_key=d9e560e9e693a3a032637bfacae5fd16'
```

Successful Response Example


```
{
  "result": 200,
  "body": [
    {
      "i": 17432,
      "t": 1673955047,
      "n": "SPA-007",
      "x": 6.1439123,
      "y": 46.203412,
      "s": 52.41,
      "a": 105.39,
      "v": 53.1342
    }
  ]
}
```


Unsuccessful Response Example


```
{
  "result": 401,
  "message": "Unauthorized"
}
```