

API Reference

Vaisala Wx Beacon XML API
Wx Beacon

PUBLISHED BY

Vaisala Oyj
Vanha Nurmijärventie 21, FI-01670 Vantaa, Finland
P.O. Box 26, FI-00421 Helsinki, Finland
+358 9 8949 1

Visit our Internet pages at www.vaisala.com.

© Vaisala 2021

Permission is hereby granted, free of charge, to any person obtaining a copy of this document (the "Document"), to deal in the Document without restriction, including without limitation the rights to use, copy, modify, merge, publish and distribute copies of the Document, and to permit persons to whom the Document is furnished to do so, subject to the following conditions:

This permission notice shall be included in all copies or substantial portions of the Document.

THE DOCUMENT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENT OR THE USE OR OTHER DEALINGS IN THE DOCUMENT.

Table of contents

1.	About this document	3
1.1	Version information.....	3
1.2	Documentation conventions.....	3
1.3	Trademarks.....	4
2.	XML API endpoint	5
3.	Creating API key and checking device serial number from Wx Beacon UI	6
4.	HTTP request and response	7
4.1	HTTP status codes.....	8
4.2	Response schema.....	9
5.	Measurement parameters and units	10
6.	Request example with Python 3	11
	Warranty	13
	Technical support	13

1. About this document

1.1 Version information

This document provides a description of Vaisala Wx Beacon XML API data reporting interface.

Table 1 Document versions (English)

Document	Date	Description
M212639-B	May 2021	XML API endpoint updated
M212639-A	March 2021	First version.

1.2 Documentation conventions



WARNING! Warning alerts you to a serious hazard. If you do not read and follow instructions carefully at this point, there is a risk of injury or even death.



CAUTION! Caution warns you of a potential hazard. If you do not read and follow instructions carefully at this point, the product could be damaged or important data could be lost.



Note highlights important information on using the product.



Tip gives information for using the product more efficiently.



Lists tools needed to perform the task.



Indicates that you need to take some notes during the task.

1.3 Trademarks

Vaisala® is a registered trademark of Vaisala Oyj.

Python® is a trademark or registered trademark of the Python Software Foundation.

All other product or company names that may be mentioned in this publication are trade names, trademarks, or registered trademarks of their respective owners.

2. XML API endpoint

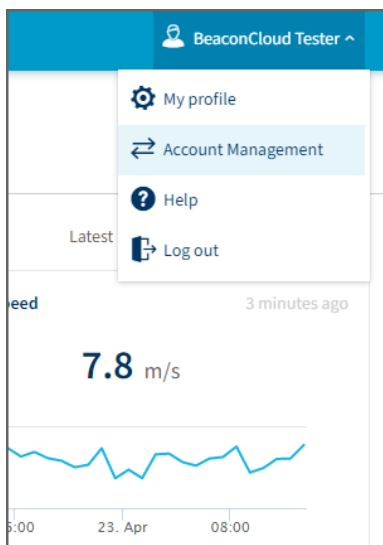
The XML API interface endpoint is:

- Wx Beacon XML API endpoint:
<https://wxbeacon.vaisala.com/api/xml>

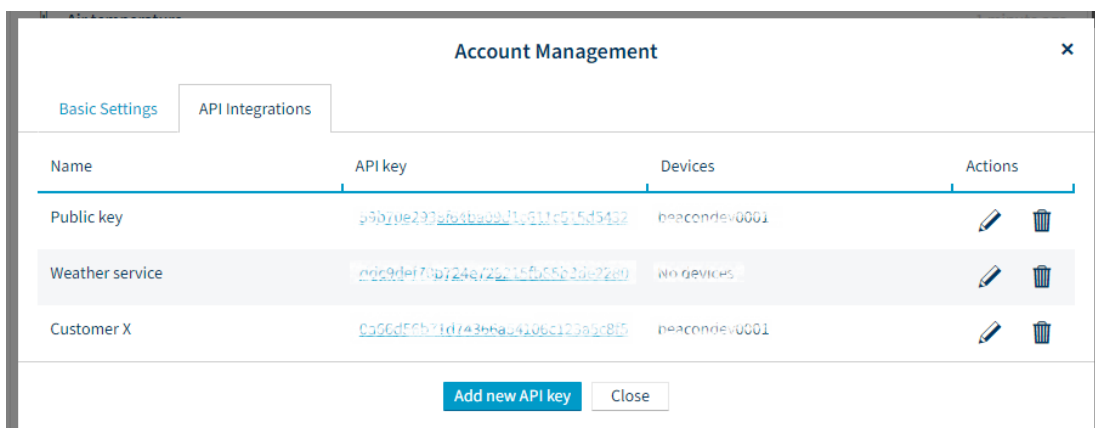
3. Creating API key and checking device serial number from Wx Beacon UI

To use the XML API, you need an API key and serial numbers of your devices.

- ▶ 1. To create or browse the API keys, select from the Wx Beacon user menu **Account Management**.



2. In **Account Management**, select **API Integrations**



Here you can:

- Create API keys.
- Edit existing API keys.
- Manage which devices are accessible with different API keys.

4. HTTP request and response

The XML API supports only HTTP GET. There is also a limitation that a user can fetch a maximum of 100 000 measurements with a single request. If more requests are needed, then subsequent requests need to be made.

The following is a cURL example of making a request with GET:

```
curl https://beacon.vaisala.com/api/xml?
d=R1234567&k=752cdd06eb1c412cb26abeb2164b980f&t0=2020-12-29T00:
00:00&t1=2020-12-29T00:01:00&c=100
```

Table 2 Request parameters

Name	Value	Example	Description
d	Serial number	R1234567	Serial number of the device.
k	API key	752cdd06ed1c412cb26abeb2164b980f	Device specific API key. Needs to be generated from Wx Beacon application to be able to use the API.
t0	Start timestamp in UTC time	2020-12-29T00:00:00	Start timestamp in ISO 8601 format. Time must be UTC time.
t1	Stop timestamp in UTC time	2020-12-29T12:00:00	End timestamp in ISO 8601 format. Time must be UTC time.
c	Max count	1000	Parameter to restrict the amount of measurements retrieved. Optional parameter.

Example response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<envitemsdata>
  <device>
    <name>Laaksolahti</name>
    <serial>R1234567</serial>
    <description>EGW501 in Laaksolahti</description>
    <location></location>
    <lat xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true" />
    <lon xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true" />
    <alt xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:nil="true" />
  </device>
  <measurements>
    <meas>
      <timestamp>2020-12-29T00:00</timestamp>
      <type>Air temperature</type>
      <value>0.4</value>
    </meas>
    <meas>
      <timestamp>2020-12-29T00:00</timestamp>
      <type>Air humidity</type>
      <value>91.5</value>
    </meas>
    <meas>
      <timestamp>2020-12-29T00:00</timestamp>
      <type>Air pressure</type>
      <value>1004.1</value>
    </meas>
  </measurements>
</envitemsdata>
```

If authentication fails, the XML API returns the following:

```
<error>Access denied!</error>
```

4.1 HTTP status codes

The XML API returns the following HTTP status codes.

HTTP status code	Status	Description
200	Success	Successful request
403	Forbidden	Wrong API key or serial number is given

4.2 Response schema

The XML API returns an XML message that is based on the following schema.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="envitemsdata">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="device">
          <xs:complexType>
            <xs:choice maxOccurs="unbounded" minOccurs="0">
              <xs:element type="xs:string" name="name"/>
              <xs:element type="xs:string" name="serial"/>
              <xs:element type="xs:byte" name="type"/>
              <xs:element type="xs:string" name="description"/>
              <xs:element type="xs:string" name="location"/>
              <xs:element type="xs:string" name="lat"/>
              <xs:element type="xs:string" name="lon"/>
              <xs:element type="xs:string" name="alt"/>
            </xs:choice>
          </xs:complexType>
        </xs:element>
        <xs:element name="measurements">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="meas" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="timestamp"/>
                    <xs:element type="xs:string" name="type"/>
                    <xs:element type="xs:float" name="value"/>
                  </xs:sequence>
                  <xs:attribute type="xs:byte" name="id" use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

5. Measurement parameters and units

Here is a list of all possible parameter types from the XML API and their fixed units.

Parameter	Unit	Unit description
Air temperature	°C	Degrees Celsius
Air pressure	hPa	Hectopascal
Relative humidity	%RH	Percentage 0 ... 100 %
Wind speed	m/s	Meters per second
Wind direction	°	Degrees 0 ... 359°
Maximum wind speed	m/s	Meters per second
Rain accumulation	mm	Millimeters
Rain intensity	mm/h	Millimeters per hour
NO ₂	ppb	Parts per billion
NO	ppb	Parts per billion
O ₃	ppb	Parts per billion
CO	ppb	Parts per billion
SO ₂	ppb	Parts per billion
H ₂ S	ppb	Parts per billion
PM ₁	µg/m ³	Micrograms per cubic meter
PM _{2.5}	µg/m ³	Micrograms per cubic meter
PM ₁₀	µg/m ³	Micrograms per cubic meter

6. Request example with Python 3

The following example retrieves data from the XML API and prints the response.

xml-api-request.py file

```
import requests
import pandas as pd
import xml.etree.ElementTree as et
from datetime import datetime, timedelta
from collections import defaultdict

def wxBeaconApiRequest(startUTC, endUTC, serial, apikey):

    endpoint = 'https://beacon.vaisala.com/api/xml/'
    query = '?d='+serial+'&k='+apikey+'&t0='+startUTC+'&t1='+endUTC
    URL = endpoint + query

    observations = et.fromstring(requests.get(URL).content)[1]

    d = defaultdict(dict)
    for obs in observations:
        timestamp = obs.find("timestamp").text
        type = obs.find("type").text
        value = obs.find("value").text
        d[timestamp][type] = float(value)

    df = pd.DataFrame.from_dict(d, orient='index')
    # convert index to datetime objects
    df.index = pd.to_datetime(df.index)

    return df

# get last 24h data
utcNow = datetime.utcnow()
startTimeUTC = (utcNow-timedelta(days=1)).strftime('%Y-%m-%dT%H:%M:%S')
endTimeUTC = utcNow.strftime('%Y-%m-%dT%H:%M:%S')
serial = 'R1234567'
apikey = '752cdd14eb1c412cb62abee2164b970f'

df = wxBeaconApiRequest(startTimeUTC, endTimeUTC, serial, apikey)

print(df)
```

Example when run in command line interface:

```
C:\proj> python .\xml-api-request.py
      Air temperature Air humidity Air pressure
2021-01-20 07:43:00      -11.7       91.1      1007.7
2021-01-20 07:44:00      -11.7       92.1      1007.7
2021-01-20 07:45:00      -11.6       92.6      1007.7
2021-01-20 07:46:00      -11.5       92.2      1007.7
2021-01-20 07:47:00      -11.7       90.1      1007.7
...
2021-01-21 05:38:00       -4.9       85.7       999.0
2021-01-21 05:39:00       -4.9       85.8       999.0
2021-01-21 05:40:00       -4.9       85.8       999.0
2021-01-21 05:41:00       -4.9       85.9       998.9
2021-01-21 05:42:00       -4.9       85.9       998.9

[1320 rows x 3 columns]
```

Warranty

For standard warranty terms and conditions, see www.vaisala.com/warranty.

Please observe that any such warranty may not be valid in case of damage due to normal wear and tear, exceptional operating conditions, negligent handling or installation, or unauthorized modifications. Please see the applicable supply contract or Conditions of Sale for details of the warranty for each product.

Technical support



Contact Vaisala technical support at helpdesk@vaisala.com. Provide at least the following supporting information as applicable:

- Product name, model, and serial number
- Software/Firmware version
- Name and location of the installation site
- Name and contact information of a technical person who can provide further information on the problem

For more information, see www.vaisala.com/support.

VAISALA

www.vaisala.com

