

SENSOTERRA

Moisture Data Integration Guide



Version 1.4
October 2022

Table of contents

Introduction.....3

- Overview.....3
- Data from private LoRaWAN networks.....4

Customer API.....5

- Documentation5
- Security.....6
 - Encryption6
 - Authentication7
 - Authorization7

Server push notifications8

- Format8
- Customer endpoint.....10
 - Testing.....10
 - Expiration.....10

Hosting locations of Sensoterra data.....11

Introduction

This document describes how you can integrate Sensoterra soil moisture data from the Sensoterra backend into your own environment. This can be as simple as downloading an Excel file with all soil moisture data up to using the Sensoterra Customer API and push notifications to fully control your own and/or your customers' sensors.

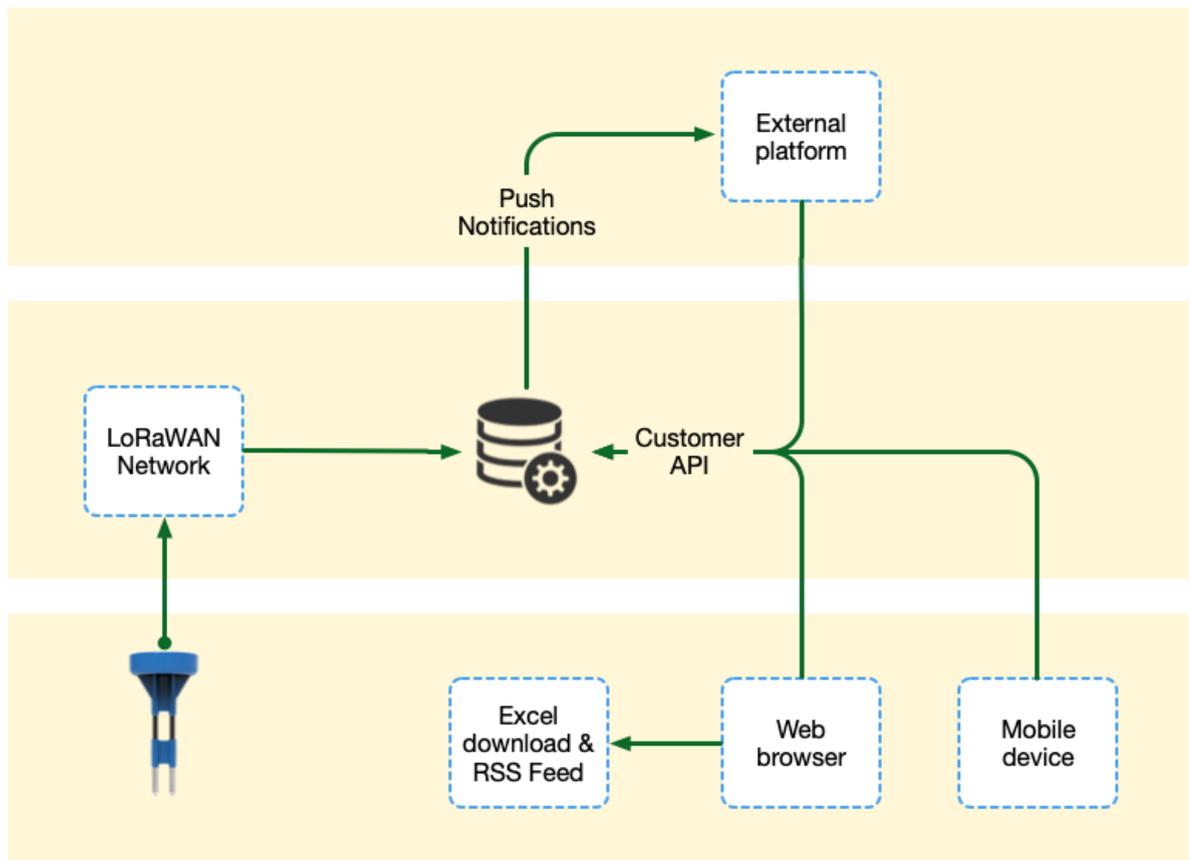
Intended audience are Sensoterra customers and resellers who want to integrate the Sensoterra sensors into their own platforms, applications and dashboards.

This document describes the methods how to get data from the Sensoterra backend via the following methods:

- Using the Sensoterra Customer API (REST)
- Using the Sensoterra Push Notifications

Please note that the payloads from Sensoterra sensors cannot be directly used. The payloads need to be decoded and calibrated by the Sensoterra backend.

Overview



Data flow is as follows:

- Sensor measurements are forwarded by the LoRaWAN network to the Sensoterra backend.
- All sensor data is decoded and calibrated by the Sensoterra backend and stored in a central database operated by Sensoterra.
- Sensor measurements and configuration are accessible through the Customer API. This API is also used by the Sensoterra Android & iOS apps and by the web site <https://monitor.sensoterra.com>.
 - Measurements can be downloaded from the monitor website as an Excel file. A GeoRSS feed is also offered with all the latest sensor data.
 - Exceptions like soil too dry/wet can be sent by mail or pushed directly to your mobile device.
- *Measurements can be pushed in real time into an external platform. This is strongly preferred over polling the API, because it is less demanding for both your environment as the Sensoterra environment.*

Data from private LoRaWAN networks

If the LoRaWAN connectivity is not managed by Sensoterra or its partners, but by the customer and/or a third-party supplier, please consult our *Custom LoRaWAN Integration Guide* on how to link the Sensoterra sensors to the LoRaWAN network and get the payloads decoded by the Sensoterra backend.

Customer API

The API is intended for external platform integration. With this API, the sensor data can be retrieved, (customer) accounts can be managed and sensor settings can be changed.

This API is a RESTful webservice using JSON as the message format. The API can be used for various management tasks like:

- Get all sensor data
- (Customer) account creation
- Sensor registration
- Change calibration settings
- Sensor location management
- etc.

For retrieving data, we strongly advise to use Push Notifications instead of polling through the Customer API.

Documentation

The customer API documentation can be found at <https://monitor.sensoterra.com/api/v3/>. To test drive the API, the documentation can be used. Preparation:

1. Create an account in the Sensoterra app (can be done in the [API](#) too).
2. Get a temporary API key in the API
3. Authorize the API with this key

To get a temporary API key, use the [Customer Auth PUT](#) call, click on “Try it out” and fill in your credentials:

customer ▾

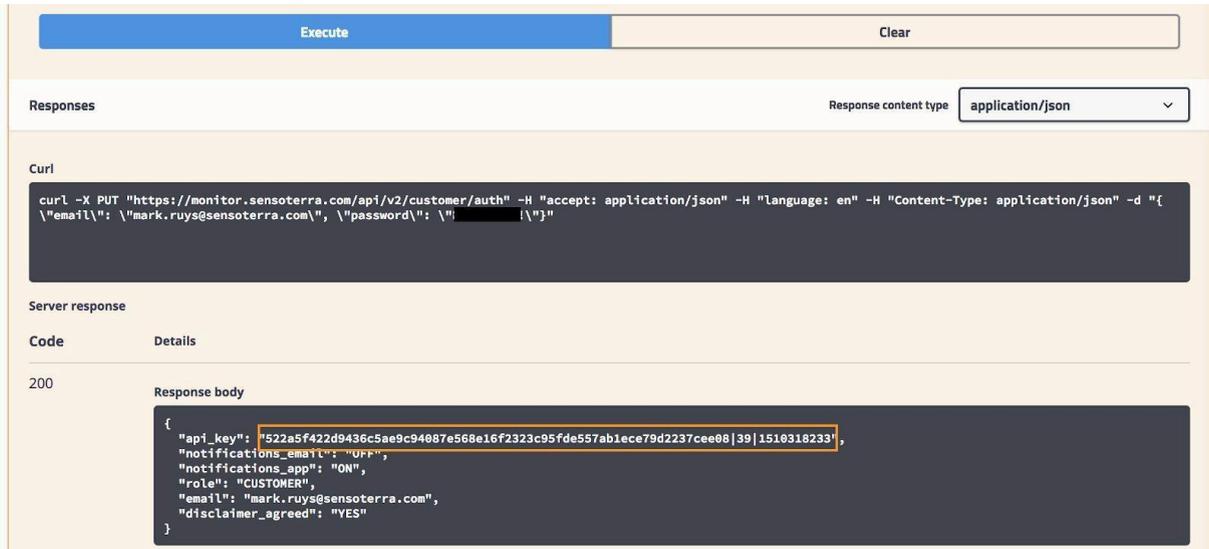
PUT /customer/auth Get an API key based on the user credentials

When the API key expires you have to renew it. Keys expire typically after 30 minutes.

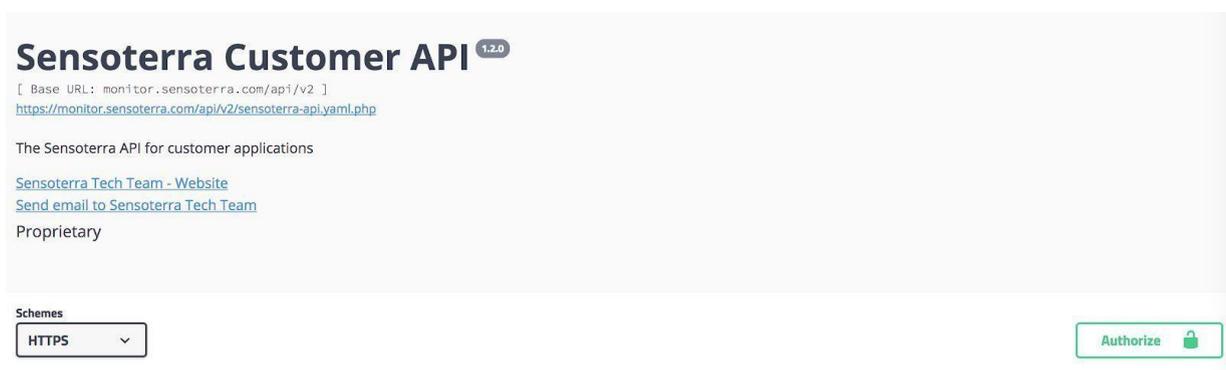
Parameters Cancel

Name	Description
body * required (body)	Customer object with login credentials. Example Value Model <pre>{ "email": "joe@somewhere.org", "password": "my_secret" }</pre>

Press “Execute” to get the key:



Click on “Authorize” to paste in the API key.



From now on, all API calls are performed in the selected account.

Security

Although the API does not reveal privacy sensitive data, it does contain sensitive business data as the geolocation of sensors, soil type, and historical soil moisture data. Therefore, the API is protected through a series of security measures.

Encryption

All communication to the servers providing the API is encrypted by TLS. Qualys Labs rates the encryption at the highest [A+ level](#).

Authentication

The Sensoterra Customer API is protected by dynamic token authentication. To get access to the API the user requests a security token by handing over its username (typically email address) and password. After the credentials are verified, the customer receives a token. For each API request, the user must add the token. A token has a limited lifetime of 30 minutes.

Authorization

The API provides three different access levels:

1. Customers
2. Resellers
3. Partners

A customer only has access to their own data. A value-added reseller has access to all data of all their customers. A partner is a reseller with read only access.

Server push notifications

As a value-added reseller or individual customer, you can have your decoded and calibrated sensor measurements forwarded in real time to your own platform by the Sensoterra backend. To be able to receive these Push Notifications, it is necessary to create an endpoint in your environment, to receive each sensor reading processed by Sensoterra. The end point will be called using a http(s) POST call containing a JSON payload.

Server push notifications use the standardized SenML format. As value-added reseller or individual customer, you set up an endpoint to which the SenML measurements are sent from the Sensoterra backend.

Format

Measurements adhere to the formal SenML specifications. We use JSON as the message format. An example of a payload of two soil measurements is:

```
[
  {
    "bver": 4,
    "bn": "urn:dev:mac:0024befff0_18000125819",
    "u": "rssi", "v": -116,
    "bt": 1505747862
  },
  { "u": "lat", "v": 52.5495116 },
  { "u": "lon", "v": 5.6128824 },
  { "u": "%EL", "v": 100 },
  { "u": "%vol", "v": 32.3, "depth": 30, "soil": "SAND", "ut": 5400 },
  { "u": "%vol", "v": 32.0, "depth": 30, "soil": "SAND", "ut": 5400, "t": -3600 },
]
```

Basically, you get a list of measurements of various sensors in time. Each record can contain the following data:

<i>Label</i>	<i>Usage</i>
bver	Base version: payload layout format, either 3 or 4.
bn	Base name to identify the probe. In case of v3 the format is urn:dev:com.sensoterra:<serial>. In case of v4 the DevEUI is added as follows: urn:dev:mac:<serial>_<deveui>. Both the serial number as the DevEUI uniquely identify a probe. A serial number is an integer of 11 positions. DevEUI is a hex string 16 characters in length, representing 8 bytes.
bt	Base time is in Unix timestamps from 1970-1-1 in UTC and contains the date/time of the measurement.

ut	Update time: within how many seconds a new measurement can be expected.
t	Time: offset relative to the base time in seconds to set the time of the measurement.
u	Unit: <ul style="list-style-type: none"> - %vol is the volumetric soil moisture - Cel: temperature in degree Celsius - rssi which represents the radio signal strength indicator in dBm (higher is better) - esp which stands for estimated signal power, roughly the sum of RSSI and SNR (higher is better) - lat, lon for the probe geolocation - %EL indicates the battery voltage (0 = min, 100 = max)
v	Value: floating point value of the measurement.
depth	SenML extension to indicate the sensor depth in cm relative to the top soil level.
soil	SenML extension to indicate the soil type used for calibration as configured by the Customer API.

Hence the example above encodes the following measurements:

- At 18 September 2017 15:17:42 (base time 1505747862) we have:
 - Radio signal strength (RSSI) of -116 dBm
 - Geolocation sensor of 52.5495116, 5.6128824
 - Soil moisture in sand of 32.3% at a depth of 30 cm
 - We expect the next value within an hour and a half
- At 18 September 2017 14:17:42 (time -3600) we have:
 - Soil moisture in sand of 32.0% at a depth of 30cm

An example of a multi depth reading in v3 format is:

```
[
{
  "bver": 3,
  "bn": " urn:dev:com.sensoterra: 24000125828",
  "u": "rssi", "v": -116,
  "bt": 1505747862
},
{ "u": "lat", "v": 52.5495116 },
{ "u": "lon", "v": 5.6128824 },
{ "u": "%EL", "v": 100 },
{ "u": "%vol", "v": 32.3, "depth": 10, "soil": "SAND", "ut": 5400 },
{ "u": "%vol", "v": 32.0, "depth": 20, "soil": "SAND", "ut": 5400 },
{ "u": "%vol", "v": 30.0, "depth": 30, "soil": "SAND", "ut": 5400 },
{ "u": "%vol", "v": 28.0, "depth": 45, "soil": "SAND", "ut": 5400 },
{ "u": "%vol", "v": 32.1, "depth": 60, "soil": "CLAY", "ut": 5400 },
```

```
{ "u": "%vol", "v": 32.9, "depth": 90, "soil": "CLAY", "ut": 5400 },
{ "u": "Cel", "v": 16.9, "depth": 15 }
]
```

Customers created after 2021-4-1 receive v4 messages by default. Older accounts get v3. Reach out to Sensoterra support in case you want to switch.

For more details on SenML, please check <https://tools.ietf.org/html/draft-ietf-core-senml-10>.

Customer endpoint

When you have created an endpoint to get the sensor measurements delivered from the Sensoterra backend, please contact support@sensoterra.com in order to activate forwarding to this endpoint.

*Typically, an endpoint will be some web service which accepts POST data. The content type is **application/senml+json**.*

Basic authentication or an optional header can be configured at request.

Testing

For testing, Sensoterra can set up a mail forward of SenML data to the customer. Please contact support@sensoterra.com to set it up.

Expiration

If a notification delivery times out, the measurements will be offered again to the end point, until the delivery succeeds, or after 35 hours when the measurements expire.

Hosting locations of Sensoterra data

Sensoterra production servers are spread over data centers in Amsterdam, the Netherlands, and Frankfurt, Germany. These data centers are operated by TelecityGroup and Interxion respectively and ISO/IEC 27001:2013 and PCI-DSS certified.