Gateway Cloud API

The information below is a swagger definition for the SensorPush public API for the Gateway cloud. Download the swagger definition file here.

Note that requests can be made no more than once per minute. If you need support, please reach out to support@sensorpush.com, please be sure to preface your email subject with "[api]" so it reaches the correct team.

Important! To activate your API access, please log in to the Gateway Cloud Dashboard and agree to the terms of service. Once you've logged in that initial time, your account will have access. You can review the terms here.

Base URL:	https://api.sensorpush.com/api/v1
Version:	v1.0.20250413
Schemes:	https

Examples

The following illustrates how to interact with the API via simple curl commands.

Important FYI about tokens:

The authorization token is returned after a successful signin. This token identifies the user as a trusted client, and is valid for 60 minutes.

The authorization token is used to request two additional tokens: access and refresh tokens. The access token authorizes the user to begin using the API. This token is valid for 30 minutes, at which time the client must request a new access token using the refresh token.

The refresh token is valid for 60 minutes. Upon requesting a new access token, the client will receive new refresh token as well. The access token is valid for another 30 minutes, and the refresh token is again valid up to an additional 60 minutes.

These steps are in accordance with the oAuth2 specifications such that if any of the three tokens are lost, the tokens eventually expire, thus securing the account once again.

For additional information, please refer to the oAuth website.

Important FYI about Gateways:

A Gateway steadily relays sensor data roughly every minute. In addition to that, a Gateway will periodically check in with the cloud to make a record to indicate that that Gateway is on and connected to the SensorPush cloud via the internet. A Gateway's check in can be observed by reading the "last_seen" property of a Gateway. Consider evaluating the "last_seen" property of a Gateway roughly every 15 minutes for a reliable indication of its status.

Example 1 - Step 1: Authorization

Log in using a valid email/password to recieve an authorization code.

copy to clipboard

```
curl -X POST "https://api.sensorpush.com/api/v1/oauth/authorize" \
-H "accept: application/json" \
-H "Content-type: application/json" \
-d @- <<BODY
{
   "email": "<email>",
   "password": "<password>"
}
BODY
```

Example 1 - Step 2: OAuth Access

Request a temporary oauth access token. Use the result from the previous step for the authorization code in the body.

copy to clipboard

```
curl -X POST "https://api.sensorpush.com/api/v1/oauth/accesstoken" \
-H "accept: application/json" \
-H "Content-type: application/json" \
-d @- <<BODY
{   "authorization": "<authorization>"
}
BODY
```

Example 2: List Gateways

Request a list of gateways. Add the header "Authorization: " using the accesstoken returned in the OAuth Access step.

copy to clipboard

```
curl -X POST "https://api.sensorpush.com/api/v1/devices/gateways" \
-H "accept: application/json" \
-H "Authorization: <accesstoken>" \
-d @- <<BODY
{}
BODY
```

Example 3: List Sensors

Request a list of sensors. Add the header "Authorization: " using the accesstoken returned in the OAuth Access step.

copy to clipboard

```
curl -X POST "https://api.sensorpush.com/api/v1/devices/sensors" \
-H "accept: application/json" \
-H "Authorization: <accesstoken>" \
-d @- <<BODY
{}
BODY
```

Example 4: Query Samples

Request up to 20 samples occuring after a timestamp with this format YYYY-MM-DDThh:mm:ss.000Z, and also add the header "Authorization: " using the accesstoken returned in the OAuth Access step.

Data for temperature is in Fahrenheit.

copy to clipboard

```
curl -X POST "https://api.sensorpush.com/api/v1/samples" \
-H "accept: application/json" \
-H "Authorization: <accesstoken>" \
-d @- <<BODY
{ "limit": 20 }
BODY
```

Example 5: Query Samples for Specific Sensors and/or Specific Start and Stop Times

Similar to the "Query Samples" example, but with an added array for specific sensor IDs, and also added startTime and stopTime.

copy to clipboard

```
curl -X POST "https://api.sensorpush.com/api/v1/samples" \
-H "accept: application/json" \
-H "Authorization: <accesstoken>" \
-d @- <<BODY
{ "sensors": ["01234.01234567890123456"],
"limit": 10000,
"startTime": "2019-03-07T10:30:00-0400",
"stopTime": "2019-04-07T10:30:00-0400"
 }
BODY
```

Security

Oauth

Name:
Authorization

In:
Header

Description:
This header value grants temporary access to data resources. Use the accesstoken value returned by the accesstoken service.

Help

Paths

Path	*Operation*	*Description*
/	POST	SensorPush API status
/devices/gateways	POST	Lists all gateways.
/devices/sensors	POST	Lists all sensors.
/oauth/accesstoken	POST	Request a temporary oAuth access code.
/oauth/authorize	POST	Sign in and request an authorization code
/oauth/token	POST	oAuth 2.0 for authorization, access, and refresh tokens
/reports/download	POST	Download bulk reports.
/reports/list	POST	Lists reports available for download.
/samples	POST	Queries for sensor samples.
/tags	POST	Updates tags on devices.

POST */*
SensorPush API status

Description:
This service is used as a simple method for clients to verify they can connect to the API.

Responses:
application/json

200 response

Status

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

POST */devices/gateways*
Lists all gateways.

Description:
This service will return an inventory of all registered gateways for this account.

Responses:
application/json

200 response

Gateways

400 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

500 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

Security:
oauth-public

POST */devices/sensors*
Lists all sensors.

Description:
This service will return an inventory of all registered sensors for this account.

Responses:
application/json

200 response

Sensors

400 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

500 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

Security:
oauth-public

POST */oauth/accesstoken*
Request a temporary oAuth access code.

Description:
This is a simplified version of oAuth in that it only supports accesstokens and does not require a client_id. See the endpoint '/api/v1/oauth/token' for the more advanced oAuth endpoint. Once a user has been authorized, the client app will call this service to receive the access token. The access token will be used to grant permissions to data stores. An access token expires every hour. After that, request a new access token.

Responses:
application/json

200 response

AccessTokenResponse

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

400 response

Error

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

500 response

Error

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

POST */oauth/authorize*
Sign in and request an authorization code

Description:
Sign into the SensorPush API via redirect to SensorPush logon. Then signin using email/password, or an api id. This service will return an oAuth authorization code that can be exchanged for an oAuth access token using the accesstoken service.

Responses:
application/json

200 response

AuthorizeResponse

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

400 response

Error

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

invalid user

500 response

Error

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

POST */oauth/token*
oAuth 2.0 for authorization, access, and refresh tokens

Description:
This is a more advanced endpoint that implements the oAuth 2.0 specification. Supports grant_types: password, refresh_token, and access_token. If grant_type is null an access_token will be returned. (see oAuth Grant Types). A client_id is required for this endpoint. Contact support@sensorpush.com to register your application and recieve a client_id.

Responses:
application/json

200 response

TokenResponse

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

400 response

Error

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

500 response

Error

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

POST */reports/download*
Download bulk reports.

Description:
This service will download bulk generated reports.

Responses:
application/json

200 response

400 response

Error

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

500 response

Error

Header	Data type
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

Security:

oauth-public

POST ***/reports/list***
Lists reports available for download.

Description:

This service will list all bulk generated reports available to download.

Responses:

application/json

200 response

ListResponse

400 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

500 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

Security:

oauth-public

POST ***/samples***
Queries for sensor samples.

Description:

This service is used to query for samples persisted by the sensors. The service will return all samples after the given parameter (startTime). Queries that produce greater than ~5MB of data will be truncated. If results return truncated, consider using the sensors parameter list. This will allow you to retrieve more data per sensor. For example, a query that does not provide a sensor list, will attempt to return equal amounts of data for all sensors (i.e. ~5MB divided by N sensors). However, if one sensor is specified, than all ~5MB will be filled for that one sensor (i.e. ~5MB divided by 1). Another option is to paginate through results by time, using (startTime) as the last date in your previous page of results.

Responses:

application/json

200 response

Samples

400 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

500 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

Security:

oauth-public

POST ***/tags***
Updates tags on devices.

Description:

This service allows users to add tags to devices.

Responses:

application/json

200 response

TagsResponse

400 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

500 response

Error

Header	*Data type*
Access-Control-Allow-Headers	string
Access-Control-Allow-Methods	string
Access-Control-Allow-Origin	string

Security:

oauth-public

Schema definitions

AccessTokenRequest
Type: object

Description:

Request object for an oAuth accesstoken code.

Properties:

authorization: string

Authorization code recieved from the oauth/authorize service.

AccessTokenResponse
Type: object

Description:

Response object for an oAuth authorization code.

Properties:

accesstoken: string

JWT oAuth accesstoken. Pass this code to the data services via the http header 'Authorization'. Example 'Authorization' : 'Bearer '.

AuthorizeRequest
Type: object

Description:

Request object for an oAuth authorization code.

Properties:

email: string

Email associated with a valid account.

password: string

Password associated with the email.

AuthorizeResponse
Type: object

Description:

Response object for an oAuth authorization code.

Properties:

authorization: string

JWT oAuth authorization code. Pass this code to the oauth/accesscode service to request an access token. The [jwt] (https://jwt.io/) website has a useful jwt viewer.

Error
Type: object

Properties:

message: string

Gateway
Type: object

Properties:

last_alert: string

Date last alert was sent

last_seen: string

Date the gateway was last seen

message: string

Detailed message associated with the gateway status.

name: string

Name associated with a gateway

paired: string

Gateway is paired with a bluetooth device

tags: object

List of tags associated with this device

version: string

Version of Sensorpush software

Gateways
Type: object

Description:

Map of registered SensorPush gateways

GatewaysRequest
Type: object

Description:

Request object for gateways.

Properties:

format: string

Returns the results as the specified format (csv|json). Defaults to json

ListResponse
Type: object

Properties:

files: array

ReportListing
Type: object

Properties:

last_modified: string

Date file was last modified

name: string

Name of the file

size: string

File size

ReportsRequest
Type: object

Description:

Request object for reports.

Properties:

path: string

The directory path to perform this operation.

Sample
Type: object

Description:
This represents one observation recorded by a given sensor. The fields listed below (except for observed) will be populated base on the measures parameter specified in the request, and if the given sensor version collects that particular measure. For example, barometric_pressure is not available in HT1 series sensors, and thus will not be reported.

Properties:

altitude: number
Value unit is feet (ft)

barometric_pressure: number
Value unit is inch of mercury (inHg)

dewpoint: number
Value unit is farenheit (°F)

humidity: number
Value unit is percentage (%)

observed: string
Date time when sample was observed.

tags: object
List of tags associated with this device

temperature: number
Value unit is farenheit (°F)

vpd: number
Value unit is kPa

Samples
Type: object

Description:
Map of registered SensorPush sensors

Properties:

last_time: string
ISO date time of the last sample returned. Use this as the start_ts to query for the next page of samples.

sensors: object
Map of sensors and the associated samples.

status: string
Message describing state of the api call.

total_samples: number
Total number of samples across all sensors

total_sensors: number
Total number of sensors returned

truncated: boolean
The query returned too many results, causing the sample list to be truncated. Consider adjusting the limit or startTime parameters.

SamplesRequest
Type: object

Description:
Request object for samples.

Properties:

active: boolean
Filters sensors by active = (true|false). Defaults to true.

bulk: boolean
Queries that return large results are truncated (see comments on Samples endpoint). Set this flag to true for large reports. The report request will be queued and processed within 24 hours. Upon completion, the primary account holder will recieve an email with a link for download.

format: string
Returns the results as the specified format (csv|json). Defaults to json

limit: integer
Number of samples to return.

measures: array
Specifies which measures to include ("temperature"|"humidity"|"vpd"|"barometric_pressure"|"dewpoint"). Note some measures are not available on older devices.

sensors: array
Filters samples by sensor id. This will be the same id returned in the sensors api call. The parameter value must be a list of strings. Example: sensors: ["123.56789"].

startTime: string
Start time to find samples (example: 2019-04-07T00:00:00-0400). Leave blank or zero to get the most recent samples.

stopTime: string
Stop time to find samples (example: 2019-04-07T10:30:00-0400). Leave blank or zero to get the most recent samples.

tags: array
Filters samples by associated tags.

Sensor
Type: object

Properties:

active: boolean
Is the sensor active?

address: string
MAC address

alerts: object
Alert settings

> ***Properties:***
>
> humidity: object
> *alert settings for humidity*
>
> > ***Properties:***
> >
> > enabled: boolean
> > *Is enabled?*
> >
> > max: number
> > *Max threshold for alert*
> >
> > min: number
> > *Min threshold for alert*
>
> temperature: object
> *alert settings for temperature*
>
> > ***Properties:***
> >
> > enabled: boolean
> > *Is enabled?*
> >
> > max: number
> > *Max threshold for alert*
> >
> > min: number
> > *Min threshold for alert*

battery_voltage: number
Current battery voltage

calibration: object
Calibration settings

> ***Properties:***
>
> humidity: number
> *Humidity calibration*
>
> temperature: number
> *Temperature calibration*

deviceId: string
Short device Id

id: string
Long device Id

name: string
Name of the sensor sensor

rssi: number
Wireless signal strength in dB at last reading

tags: object
List of tags associated with this device

type: string
Type of device hardward

Sensors
Type: object

Description:
Map of registered SensorPush sensors

SensorsRequest
Type: object

Description:
Request object for sensors.

Properties:

active: boolean
filters sensors by active = (true|false). Defaults to true

format: string
Returns the results as the specified format (csv|json). Defaults to json

Status
Type: object

Required:

- message

Properties:

deployed: string
Date time when this service was last updated.

message: string
Greeting message.

ms: integer
Current date time on the server in milliseconds.

stack: string
Active stack hosting this service.

status: string
Current status of the api service.

time: string
Current date time on the server.

version: string
Version of this service currently deployed

Tags
Type: object

Description:
Map of registered devices and their tags.

Properties:

gateways: object

sensors: object

TagsRequest
Type: object

Description:
Map of registered devices and their tags.

Properties:

sensors: object

TagsResponse
Type: object

Description:
Response object resulting from updating tags on devices.

Properties:

status: string
Message indicating if the tags were successfully updated.

TokenRequest
Type: object

Description:
Request object for an oAuth accesstoken code.

Properties:

client_id: string
Client Id assigned to 3rd party applications. Contact support@sensorpush.com to register you app.

client_secret: string
Password associated with the client_id

code: string

This can be an authorization, access, or refresh token. Depending on which grant_type you are using.

grant_type: string
Accepted values are password, refresh_token, and access_token

password: string
User's password

redirect_uri: string
Redirection url to the 3rd party application once the user has signed into the sensorpush logon. This value should be URL encoded.

refresh_token: string
Refresh token used to request new access tokens.

username: string
Email of the user to sign in.

TokenResponse
Type: object

Description:
Response object for an oAuth authorization code.

Properties:

access_token: string
JWT oAuth access token. Pass this token to the data services via the http header 'Authorization'. Example 'Authorization' : 'Bearer '

expires_in: number
TTL of the token in seconds

refresh_token: string
JWT oAuth refresh token. Pass this token to the token service to retrieve a new access token.

token_type: string
Type of token returned