tRacket API v1

Database Schema

User

- **Username** (Text - typically same as Email)
- **Email** (Text)
- **FirstName** (Text)
- **LastName** (Text)
- **DateRegistered** (Date)

Location

- **Label** (Text - typically the name of two cross streets)
- **PublicLatitude** (Decimal - typically randomized to the nearest cross streets)
- **PublicLongitude** (Decimal - typically randomized to the nearest cross streets)
- **PublicRadius** (Int - specifying the radius in meters of the accuracy circle)

Device

- **DeviceID** (Text - Device ID of the device, auto-generated by the device based on its MAC Address)
- **Name** (Text - optional, user-specified, defaults to DeviceID)
- **Model** (Text - something like “Gabe Device”)
- **Version** (Text - typically 1.0.0 format representing the hardware version)
- **Notify** (True/False to determine if notifications are enabled for this device)

Device History

- **RevisionStart** (Timestamp)
- **RevisionEnd** (Timestamp - 0000-00-00 00:00:00 means no Revision End)
- **Device** (Relationship)
- **User** (Relationship)
- **Location** (Relationship)
- **SoftwareVersion** (Text - typically 1.0.0 format representing the software version)

Measurement

- **Timestamp** (Timestamp) - Should include timezone (assume UTC?)
- Duration/End Timestamp (currently hard coded to 5 min) - no for now.
- **Device** (Relationship)
- **User** (Relationship)

- **Location** (Relationship)
- **Min** (Decimal)
- **Max** (Decimal)
- **Mean** (Decimal)

Endpoints

All endpoints below fall under the base API URL:
`https://api.tracket.info/v1/`

Timestamps and Time Zones

Timestamps specified in requests may be in any format recognized by PHP strtotime(), which is just about every standard format, as well as relative times like “now” and “yesterday”. If a timezone is not specified, UTC will be assumed.

Timestamps provided in all responses are in ISO 8601 format with a timezone. For example, April 14, 2024 at 4:13pm in Eastern Time will be: `2024-04-14T16:13:00-04:00`

Responses

API responses typically include the following properties:

- **message** - a human readable message about the outcome of the request.
- **result** - a short and concise string to indicate if the action was executed (“ok”) or not (“error”).
- **timestamp** - a timestamp representing the time the response was sent, according to the API server.
- **log** - optional event log, included only if there were one or more events logged while handling the request. Each logged event will include the properties:
    - **timestamp** - When the event was logged.
    - **type** - The type of event (“Notice”, “Warning”, “Error”).
    - **message** - Details about the event, as a text string.

Errors

API errors are typically included in the API response under the **log** property. TODO: They are also logged to the API Error Log in the database for reference. In the future, logs may also be sent to the #proj-noisemeter Slack channel.

measurement (HTTP POST or GET)

Add decibel measurements for a time, device and user, which *may* correspond to an active Location (a Location referenced by a Device History with a matching Device, User and no RevisionEnd set).

NOTE: If the device is not associated with a Location at the time of the Timestamp, no location will be recorded for the measurement. The measurement will still be recorded for the device.

HTTP Headers

- **Authorization** - The request must specify an Authorization header with a valid token in the following format, where 1234 is a valid user token returned by the device register endpoint outlined below. The token must always be preceded by the word “Token” and a space.
  `Authorization: Token` *1234*
- **X-Tracket-Device** - Device ID (may be a hash or MDF based on the MAC Address), as an alternative to providing the email as an HTTP Post or Get Parameter, to keep it private and out of server logs. If provided, the device parameter below is not required, and will be ignored.
  `X-Tracket-Device:` *1234*

HTTP Post or Get Parameters

- **timestamp** - End time of the sample period for the reported measurements.
- **device** - Device ID (may be a hash or MDF based on the MAC Address)
- **min** - Minimum decibel level during the sample period.
- **max** - Maximum decibel level during the sample period.
- **mean** - Average decibel level during the sample period.
- **boottime** - Optional timestamp that represents when the device was (re)started. As an alternative, the number of seconds since the device booted can be specified.
- ~~**wifitime** - Optional timestamp that represents the last time it had a wifi connection.~~
- **version** - Optional software version number the device is running.

Example HTTP Request

https://api.tracket.info/v1/measurement?timestamp=2024-03-20%2012:00:00&device=574155&user=mitch@webcomand.com&min=1&max=2&mean=1.5&boottime=2024-04-14T10:30:00-04:00

measurements (HTTP POST)

Add one or more decibel measurements for a device and user, which may correspond to an active Location (Device History with a matching Device, User, Location and no RevisionEnd).

NOTE: If the device is not associated with a Location at the time of the Timestamp, no location will be recorded for the measurement. The measurement will still be recorded for the device.

HTTP Headers

- **Authorization** - The request must specify an Authorization header with a valid token in the following format, where 1234 is a valid user token returned by the device register endpoint outlined below. The token must always be preceded by the word “Token” and a space.
  `Authorization: Token` *`1234`*
- **X-Tracket-Device** - Device ID (typically a hash of the MAC Address).
  `X-Tracket-Device:` *`1234`*

HTTP Body

JSON array of one or more measurement objects, where each measurement object has the following properties.

- **timestamp** - End time of the sample period for the reported measurements.
- **min** - Minimum decibel level during the sample period.
- **max** - Maximum decibel level during the sample period.
- **mean** - Average decibel level during the sample period.
- **boottime** - Optional timestamp that represents when the device was (re)started. As an alternative, the number of seconds since the device booted can be specified.
- **version** - Optional software version number the device is running.

Example HTTP Request

URL: https://api.tracket.info/v1/measurements
Headers:

- `Authorization: Token` *`1234`*
- `X-Tracket-Device:` *`1234`*
- `Content-Type: application/json`

Body:

```
[
      {"timestamp": "2024-06-15T16:30:00-04:00", "min": 1, "max": 2, "mean": 1.555,
      "boottime": "2024-06-14T16:30:00-04:00", "version": "0.0.4"},
      {"timestamp": "2024-06-15T16:31:00-04:00", "min": 2, "max": 3, "mean": 2.666,
      "boottime": "2024-06-14T16:30:00-04:00", "version": "0.0.4"}
]
```

locations (HTTP GET or POST)

Get a list of locations, including their id, label, latitude, longitude and if there is a device actively configured to collect measurements at that location.

HTTP Post or Get Parameters

- **page** - Optional zero-based page index into long result sets (100 results per page).

Example HTTP Request

https://api.tracket.info/v1/locations

HTTP Response Example (JSON)


```
{
    "locations":[
        {
            "id": 572227,
            "label": "Scott St. & Wellington St.",
            "latitude": 43.6481,
            "longitude": -79.376,
            "radius": 25,
            "active": true,
            "lastChecked": "2024-09-28T15:08:08-04:00",
            "latestTimestamp": "2024-07-06T14:03:00-04:00",
            "hourStart": "2024-07-06T13:03:00-04:00",
            "hourEnd": "2024-07-06T14:03:00-04:00",
            "hourMin": 2,
            "hourMax": 5,
            "hourMean": 2.555
        },{
            "id": 572234,
            "label": "Ossington Ave. & Dupont St.",
            "latitude": 43.6701,
            "longitude": -79.4293,
            "radius": 25,
            "active": true,
            "lastChecked": "2024-09-28T15:08:09-04:00",
            "latestTimestamp": "2024-08-08T15:04:30-04:00",
            "hourStart": "2024-08-08T14:04:30-04:00",
            "hourEnd": "2024-08-08T15:04:30-04:00",
            "hourMin": 56,
            "hourMax": 83,
            "hourMean": 65.308
        }
```

```
    ]
}
```

locations/<*Location ID*>/noise (HTTP GET or POST)

Return measurements for the given location. The timestamps will be returned for the timezone where the device is located.

HTTP Post or Get Parameters

- **granularity** - “raw”, “hourly”, “life-time”
- **start** - Optional start time in format recognized by strtotime (e.g. 2024-03-19 14:00:12).
- **end** - Optional end time in format recognized by strtotime (e.g. 2024-03-19 14:00:12).
- **page** - Optional zero-based page index into long result sets (100 results per page).

Example HTTP Requests

https://api.tracket.info/v1/locations/572250/noise
https://api.tracket.info/v1/locations/572250/noise?granularity=hourly
https://api.tracket.info/v1/locations/572250/noise?granularity=life-time

HTTP Response Example (JSON) - no parameters


```
{
    "measurements":[
        {
            "timestamp": "2024-03-10T20:33:08+04:00",
            "min": 48.7052803,
            "max": 67.45218658,
            "mean": 56.614048
        },{
            "timestamp": "2024-03-10T20:38:10+04:00",
            "min": 47.83936691,
            "max": 62.44270706,
            "mean": 54.13415527
        }
    ]
}
```


HTTP Response Example (JSON) - ?granularity=hourly


```
{
    "measurements":[
        {
            "timestamp": "2024-03-10T20:00:00+04:00",
            "min": 48.53691864,
            "max": 62.83390045,
```

```
            "mean": 49.0464433
        },{
            "timestamp": "2024-02-05T00:00:00+04:00",
            "min": 48.42551422,
            "max": 52.37109756,
            "mean": 49.25240326
        }
    ]
}
```


HTTP Response Example (JSON) - ?granularity=life-time


```
{
    "measurements":[
        {
            "start": "2024-02-04T23:32:58+04:00",
            "end": "2024-03-19T15:53:49+04:00",
            "count": 12368,
            "min": 35.77902985,
            "max": 92.33490753,
            "mean": 46.83609481
        }
    ]
}
```


device/register (HTTP POST or GET)

Register a user to a device. The device must be registered in the system, which is done by Noise Meter team before it is shipped. When the device is registered to a new/different email address, a Registration Confirmation email will be sent to the email address to let them know. If the device was previously associated with a different email address, an email will also be sent to the previous address. If the email for a device changes, the previous location associated with the device will be cleared and must be set again.

HTTP Headers

- **X-Tracket-Device** - Device ID (may be a hash or MDF based on the MAC Address), as an alternative to providing the email as an HTTP Post or Get Parameter, to keep it private and out of server logs. If provided, the device parameter below is not required, and will be ignored.
  `X-Tracket-Device: 1234`
- **X-Tracket-Email** - Email address of the user, as an alternative to providing the email as an HTTP Post or Get Parameter, to keep it private and out of server logs. If provided, the device parameter below is not required, and will be ignored.

HTTP Post or Get Parameters

- **device** - The device ID, which is an MD5 hash of the device's MAC Address.
- **email** - Email address of the user that would like to register this device. If the email address is not already in the system, a user with that email address will be added, which can be used to login to the device manager to access and configure the device.

Example HTTP Requests

https://api.tracket.info/v1/device/register?device=eb341720ca3a3485461a61b1e97d31b1&email=name@domain.com

HTTP Response Example (JSON)

```
{
    "message": "Device registered.",
    "result": "ok",
    "timestamp": "2024-04-14T12:04:26-04:00",
    “token”: “1234”
}
```

device/set_location (HTTP POST or GET)

Set the location of a device, and add the location, or update an existing location with an identical latitude, longitude and radius previously set for the same device and user.

HTTP Headers

- **Authorization** - The request must specify an Authorization header with a valid token in the following format, where 1234 is a valid user token returned by the device register endpoint outlined just above. The token must always be preceded by the word “Token” and a space.
  `Authorization: Token 1234`

HTTP Post or Get Parameters

- **device** - The device ID, which is an MD5 hash of the device's MAC Address.
- **latitude** - Public latitude of the location.
- **longitude** - Public longitude of the location.
- **radius** - Radius around the provided public longitude and longitude that contains the actual (private) latitude and longitude only known to the device owner.
- **publicLabel** - Public label of the location, which is typically a cross-street.
- **privateLabel** - Private label of the location, which is only visible to the device owner.

Example HTTP Requests

https://api.tracket.info/v1/device/set_location?device=eb341820cd3a3485461a61b1e97d31b1&latitude=43.648&longitude=-79.376&radius=50&publicLabel=Scott St. %26 Wellington St.&privateLabel=My home.

HTTP Response Example (JSON)

```
{
    "message": "Device location set.",
    "result": "ok",
    "timestamp": "2024-04-14T12:04:26-04:00"
}
```

software/latest (HTTP GET or POST)

Get the latest version of the software, including a URL where the software can be downlaoded.

Example HTTP Requests

https://api.tracket.info/v1/software/latest

HTTP Response Example (JSON)

```
{
    "message": "Found latest software.",
    "result": "ok",
    "timestamp": "2024-04-14T12:04:26-04:00",
    “version”: “0.0.4”,
    “releaseTime”: “2024-04-13T17:25:44-04:00”,
    “url”: “https://live.noisemeter.webcomand.com/updates/update-0.0.4.bin”
}
```

Original proposal from Gabe:

API Questions & Answers

- Device Registration (from device web server set up)
    - Collect user email?
      (Clyne) We can have the user input their email on the device's setup form. Once the device is on WiFi, it could PUT to a registration endpoint with the device's ID and the email address.
    - Can we get and send the device MAC address?
      (Clyne) Yes. We'll decide this weekend on using MAC addresses for unique device IDs vs. generating random IDs that we burn into the device.
    - Should we allow the user to override the default API endpoint URL, so it can be used for other projects or updated manually if needed in the future? This could be baked into a future software update, but the updated API endpoint could impact where we look for updates and downloads.
    - What should we call the model and hardware version?
      (Clyne) We should come up with a model name, even if it's as simple as "NM". I'm identifying hardware revisions as "rev1", "rev2", and so on; we only need to store the number if we want this info in the database.
- User Device Registration (on website)
    - Collect user email for user account (can use for one-time-passwords and/or forgot password feature). What about Google, X, Facebook login? We can potentially get email from those if authorized to share in the future.
    -
- Measurements
    - Should we store measurements when device and user are known, but the device location is not known? I vote no. Would have to go back and correct retroactively.
    - Should we require a User Token or Device Token to add a measurement? User Token would require communication with API at set up (user would need to be registered). Device Token would be added to device by us when shipped somehow? We could register valid MAC addresses before shipping a device.
      (Clyne) My idea is to have the server respond to a successful registration request (when we send the user's email) with a device-specific API token that the device stores into its memory.
    - Should we also record the start time or sample duration with Measurements?
    - Should we send Device ID or Mac Address?
      (Clyne) The device ID should be the only unique hardware identifier we need. It might be based on the MAC address.
    - Should we send User ID or Email?
    - Should the Timestamp be within 24 hours or a week of the current time?
    - Should we store just the Location, or the Device and/or User as well?
- API User / Device Details
    - Should we require a simple User Token (expires or never expires) or more complex oAuth Access Token? If a simple user token, how will the API validate a

user? Do they have to login from the website/app making the API calls, which then obtains the token until the token expires (after no requests for more than x minutes?), and then require another login?

- API workflow for simple token will be, send user to web page with callback URL, where the user will login and authorize the “app” at the hostname of the callback URL. Once authorized, the login will ping the callback URL with the User Token (that could expire, requiring another login). oAuth is similar, but more complex.

Original proposal from Daniel:

Noisemeter project API endpoint documentation (beta)

Status: work in progress (partial implementation detailed above)

DB schema reference:
https://www.figma.com/file/jZWxG4VOAATuRVIhlarWBt/NoiseMeter-Dashboard-%26-DB-Ideation?type=whiteboard&node-id=0-1&t=Y1G7UGbfswIE37y2-0

Data loading queries
https://github.com/danieltsoukup/noise-dashboard/blob/main/app/src/data_loading.py#L13

Sensor adding data

Public Data Loading API

Intent: Load list of locations

`/locations`

Parameters: none.

-

Returns: list of all locations with relevant information.

- **LocationID**
- **ANONYMIZED lat/lon** [we should only store anonymized location for better privacy protection - when user submits a new location, noise is added?]
- **LocationActive** [mark if the location active by checking the device-history table end date]
- Name? [not sure if this is user submitted, and if necessary - could expose real location if the user is not careful saw that we have a location name in the DB schema]

Intent: Load Noise Data for Specific Location

```
/locations/123/noise?
    granularity=...
    &start=...
    &end=...
```

Parameters:

- **Granularity**: “raw” [default?], “hourly”, “life-time”
    - “raw” is the original 5-minute measures from the location
    - “hourly” is data being aggregated for hours at 0 minutes.
    - “life-time” means a single value
- **Optional Start Time** [default None]
- **Optional End time** [default None]

Outputs: noise measurements (min, max, average) between start and end date, rolled up to the specified granularity (take minimum of min-measures, maximum of max-measures and the average of averages).

- **Date-time**
- **LocationID** (123 in the example or whatever specified by the request already)
- **Min, Max, Average**

Intent: Load System Level Stats

```
/noise?
    start=...
    &end=...
```

Parameters:

- **Optional Start Time** [default None]
- **Optional End time** [default None]

Outputs: system level aggregate statistics based on all locations (take minimum of min-measures, maximum of max-measures and the average of averages).

- **Start Date** [user provided]
- **End Date** [user provided]
- **Min, Max, Average**

Note: In a v2 of the API this `/noise` interface can be extended for much greater flexibility (pull a subset of locations, aggregate per location, set granularity for time, etc).

Dashboard API

Average Ambient Noise over Past 2 Weeks

On Mitch's API: Load Noise Data (granarity='lifetime', start_time='2 weeks ago', don't specify location, measure='average')

Account page loading data

Account page editing data