SensorServer

Node.js server for collecting sensor data

This server allows various sensor apps to send their data via REST to store them in a DB. The server processes the raw sensor data into history tables and offer a UI to view the data.

TODO

- Move the DB logic out of the API router
- Compete REST API with some JSON object for Domotica and history
- Auth?
- Implement CronJob for cleaning up historys.
- Prevent SQL injection, aka clean up some SQL statements

Database Tables

The databse holds the following tables - Sensors - Id: Unique Id for the sensor - Name: Sensor name - Units: kWh, M3 - High: The counter for low rate energy from 7:00 till 23:00 - Low: The counter for low rate energy from 23:00 till 7:00 - Volume: The volume of each pulse, in my case 1/375 for electricity
- SensorEvents - Id: Unique Id for the event - SenorId: Unique Id for the sensor - Time: Unix Epoch in milliseconds - Rate: 1 for low, 2 fro high - MinuteHistory, HourHistory, DayHistory, YearHistory - Id: Unique Id for the event - SenorId: Unique Id for the sensor - Time: Unix Epoch in milliseconds - Rate: 1 for low, 2 fro high - Usage: Amount of usage for tables timespan
e.g. minutes, hours, days, months

Rest API

The Rest API can insert, update and query the tables

The following endpoinst are available: - /api/sensors - POST: Inserts a new Sensor record. `json { "id": sensorId, "name": sensorName, "units": sensorUnits, "high": sensorHigh, "low": sensorLow, "volume": sensorVolume }` - GET: Returns all sensors from the Sensors tabel - PUT: Updates the sensor information in the table `json { "id": sensorId, "name": sensorName, "units": sensorUnits, "high": sensorHigh, "low": sensorLow, "volume": sensorVolume }`

- /api/sensors/:id
  Returns the sensor information from the Sensors tabel for the specified Id

- /api/sensorevents
- POST: Inserts a new SensorEvent record. `json   sensordata: { "id": sensorId, "time": currentTime, "rate": currentRate   }`
- GET:Returns the last 10 sensorevents from the SensorEvents tabel
- /api/sensorevents/:id
  Returns the last 10 sensorevents from the SensorEvents tabel for the specified Id
- /api/sensors/:id/now
  Returns the following JSON:
  `json   { "Total" : Total Usage, "High" : High Usage, "Low" : Low Usage, "UsageTime" : Timestamp of the last known usage, "Usage" : Last known usage based on the last 2 sensorevents }`

Builtin Cron Jobs

A job runs every minute to consolidate the sensorevents table into the miniute-history table and clean up the consolsidated sensorevents.

A second job runs every hour to fill and update all history tables.

Running in the background

Use the module https://www.npmjs.com/package/forever to run the server in the background.