API – Zugriff

Sensordaten

# Aufgabenstellung

Erfassen und Speichern von Sensorwerten mit dem Aeotec Multisensor

**Zielsetzung**: Deine Aufgabe ist es, zu lernen, wie du mit Hilfe einer REST API auf den Aeotec Multisensor zugreifen kannst, die Sensordaten (Luftfeuchtigkeit, Temperatur und Helligkeit) im Minutentakt erfassen und in einer Datenbank speichern kannst. Zusätzlich sollst du Diagramme der gespeicherten Daten erstellen und den Zusammenhang zwischen Temperatur und Luftfeuchtigkeit untersuchen.

**Voraussetzungen:**

- Du hast Zugriff auf die REST API des Aeotec Multisensors über das lokale Netzwerk.

- Die SSID und der PIN-Code für das Netzwerk sind bekannt.

- Ein JSON-String mit allen verfügbaren Sensoren des Aeotec Multisensors wurde bereitgestellt.

**Hinweise zur Durchführung:**

1. Vorbereitung:

- Analysiere die bereitgestellten JSON-Daten und verstehe, welche Sensoren verfügbar sind und wie diese in der API angesprochen werden können.

2. Code-Erweiterung:

- Erweitere den bereitgestellten Beispiel-Code in Python oder R so, dass die Sensordaten für Luftfeuchtigkeit, Temperatur und Helligkeit im Minutentakt abgerufen werden.

- Speichere jeden Messwert zusammen mit einem Zeitstempel in einer eigenen Tabelle in einer Datenbank.

- Führe die Messung über einen Zeitraum von 30 Minuten durch.

3. Maßnahmen zur Beeinflussung der Messwerte:

Die ersten fünf Schüler, die mit der Messung beginnen, sollen je eine Maßnahme ergreifen, um die Sensordaten zu beeinflussen.

Mögliche Maßnahmen könnten sein: Platzierung des Sensors in die Nähe einer Wärme- oder Kältequelle, Änderung der Umgebungsbedingungen usw.

Jede Maßnahme muss mit genauer Uhrzeit an der Tafel notiert werden.

4. Diagrammerstellung:

- Erstelle Diagramme der gespeicherten Daten für Luftfeuchtigkeit, Temperatur und Helligkeit im DB-Browser sowie in Python oder wahlweise in R

- Erstelle mindestens ein Diagramm, in dem die Kurven für Temperatur und Luftfeuchtigkeit gemeinsam dargestellt werden.

5. Bonus-Aufgabe:

- Schätze ein lineares Regressionsmodell für den Zusammenhang zwischen Temperatur und Luftfeuchtigkeit.

**Abgabe**:

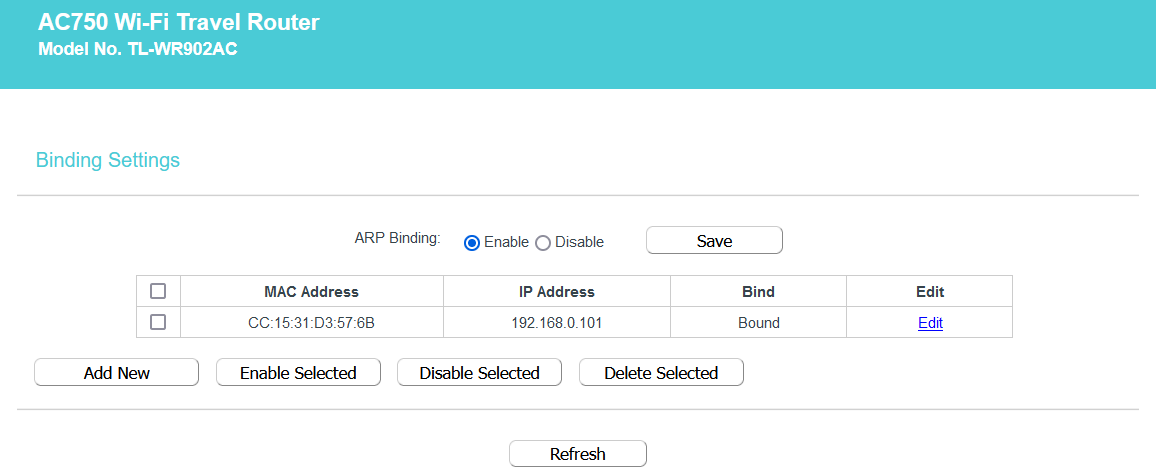
A) Reiche den erweiterten Code zusammen mit den erstellten Diagrammen ein.

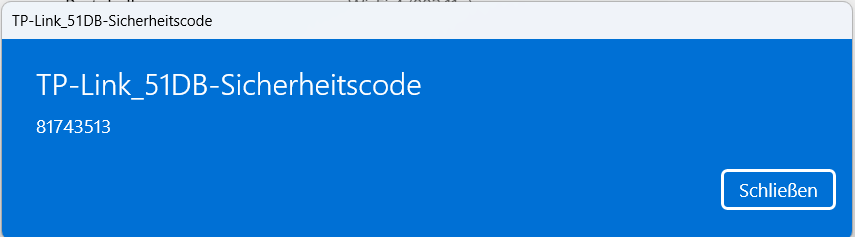
B) Die Diagramme sollen die gespeicherten Daten aus der Datenbank verwenden und übersichtlich gestaltet sein. Die Interpretation und Dokumentation der Diagramme sollte die Art und Uhrzeit der Maßnahmen und die erwarteten Auswirkungen auf die Sensordaten enthalten.

\*) Zusätzlich kannst du das lineare Regressionsmodell als Bonus-Aufgabe durchführen und die Ergebnisse dokumentieren.

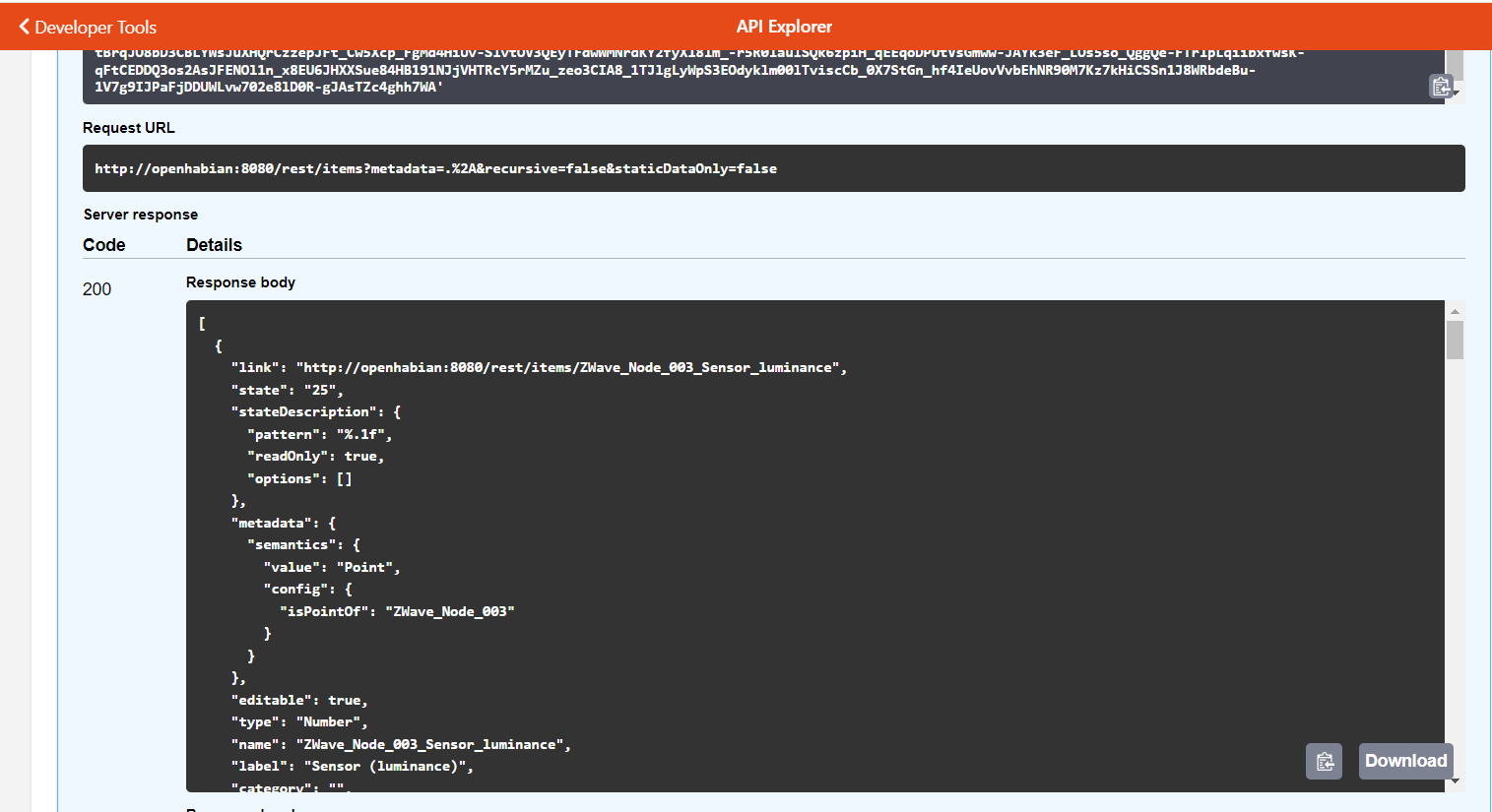
## Informationen zur Durchführung

### Netzwerk





### Abfrage aller Sensoren bzw. Kanäle



**[**

**{**

**"link": "http://openhabian:8080/rest/items/ZWave\_Node\_003\_Sensor\_luminance",**

**"state": "25",**

**"stateDescription": {**

**"pattern": "%.1f",**

**"readOnly": true,**

**"options": []**

**},**

**"metadata": {**

**"semantics": {**

**"value": "Point",**

**"config": {**

**"isPointOf": "ZWave\_Node\_003"**

**}**

**}**

**},**

**"editable": true,**

**"type": "Number",**

**"name": "ZWave\_Node\_003\_Sensor\_luminance",**

**"label": "Sensor (luminance)",**

**"category": "",**

**"tags": [**

**"Point"**

**],**

**"groupNames": [**

**"ZWave\_Node\_003"**

**]**

**},**

**{**

**"link": "http://openhabian:8080/rest/items/ZWave\_Node\_003\_Sensor\_ultraviolet",**

**"state": "0",**

**"stateDescription": {**

**"pattern": "%.1f",**

**"readOnly": true,**

**"options": []**

**},**

**"metadata": {**

**"semantics": {**

**"value": "Point",**

**"config": {**

**"isPointOf": "ZWave\_Node\_003"**

**}**

**}**

**},**

**"editable": true,**

**"type": "Number",**

**"name": "ZWave\_Node\_003\_Sensor\_ultraviolet",**

**"label": "Sensor (ultraviolet)",**

**"category": "",**

**"tags": [**

**"Point"**

**],**

**"groupNames": [**

**"ZWave\_Node\_003"**

**]**

**},**

**{**

**"members": [],**

**"link": "http://openhabian:8080/rest/items/ZWave\_Node\_003",**

**"state": "NULL",**

**"metadata": {**

**"semantics": {**

**"value": "Equipment"**

**}**

**},**

**"editable": true,**

**"type": "Group",**

**"name": "ZWave\_Node\_003",**

**"label": "Z-Wave Node 003",**

**"category": "",**

**"tags": [**

**"Equipment"**

**],**

**"groupNames": []**

**},**

**{**

**"link": "http://openhabian:8080/rest/items/ZWave\_Node\_003\_Tamper\_Alarm",**

**"state": "OFF",**

**"stateDescription": {**

**"readOnly": true,**

**"options": [**

**{**

**"value": "OFF",**

**"label": "OK"**

**},**

**{**

**"value": "ON",**

**"label": "Alarm"**

**}**

**]**

**},**

**"commandDescription": {**

**"commandOptions": [**

**{**

**"command": "OFF",**

**"label": "OK"**

**},**

**{**

**"command": "ON",**

**"label": "Alarm"**

**}**

**]**

**},**

**"metadata": {**

**"semantics": {**

**"value": "Point",**

**"config": {**

**"isPointOf": "ZWave\_Node\_003"**

**}**

**}**

**},**

**"editable": true,**

**"type": "Switch",**

**"name": "ZWave\_Node\_003\_Tamper\_Alarm",**

**"label": "Tamper Alarm",**

**"category": "",**

**"tags": [**

**"Point"**

**],**

**"groupNames": [**

**"ZWave\_Node\_003"**

**]**

**},**

**{**

**"link": "http://openhabian:8080/rest/items/ZWave\_Node\_003\_Batterieladung",**

**"state": "100",**

**"stateDescription": {**

**"minimum": 0,**

**"maximum": 100,**

**"step": 1,**

**"pattern": "%.0f %%",**

**"readOnly": true,**

**"options": []**

**},**

**"metadata": {**

**"semantics": {**

**"value": "Point\_Measurement",**

**"config": {**

**"relatesTo": "Property\_Energy",**

**"isPointOf": "ZWave\_Node\_003"**

**}**

**}**

**},**

**"editable": true,**

**"type": "Number",**

**"name": "ZWave\_Node\_003\_Batterieladung",**

**"label": "Batterieladung",**

**"category": "Battery",**

**"tags": [**

**"Measurement",**

**"Energy"**

**],**

**"groupNames": [**

**"ZWave\_Node\_003"**

**]**

**},**

**{**

**"link": "http://openhabian:8080/rest/items/ZWave\_Node\_003\_Motion\_Alarm",**

**"state": "ON",**

**"stateDescription": {**

**"readOnly": true,**

**"options": [**

**{**

**"value": "OFF",**

**"label": "OK"**

**},**

**{**

**"value": "ON",**

**"label": "Alarm"**

**}**

**]**

**},**

**"commandDescription": {**

**"commandOptions": [**

**{**

**"command": "OFF",**

**"label": "OK"**

**},**

**{**

**"command": "ON",**

**"label": "Alarm"**

**}**

**]**

**},**

**"metadata": {**

**"semantics": {**

**"value": "Point",**

**"config": {**

**"isPointOf": "ZWave\_Node\_003"**

**}**

**}**

**},**

**"editable": true,**

**"type": "Switch",**

**"name": "ZWave\_Node\_003\_Motion\_Alarm",**

**"label": "Motion Alarm",**

**"category": "Motion",**

**"tags": [**

**"Point"**

**],**

**"groupNames": [**

**"ZWave\_Node\_003"**

**]**

**},**

**{**

**"link": "http://openhabian:8080/rest/items/ZWave\_Node\_003\_Sensor\_temperature",**

**"state": "24.5 °C",**

**"stateDescription": {**

**"pattern": "%.1f %unit%",**

**"readOnly": true,**

**"options": []**

**},**

**"unitSymbol": "°C",**

**"metadata": {**

**"semantics": {**

**"value": "Point",**

**"config": {**

**"isPointOf": "ZWave\_Node\_003"**

**}**

**}**

**},**

**"editable": true,**

**"type": "Number:Temperature",**

**"name": "ZWave\_Node\_003\_Sensor\_temperature",**

**"label": "Sensor (temperature)",**

**"category": "Temperature",**

**"tags": [**

**"Point"**

**],**

**"groupNames": [**

**"ZWave\_Node\_003"**

**]**

**},**

**{**

**"link": "http://openhabian:8080/rest/items/ZWave\_Node\_003\_Sensor\_relative\_humidity",**

**"state": "41",**

**"stateDescription": {**

**"pattern": "%.1f",**

**"readOnly": true,**

**"options": []**

**},**

**"metadata": {**

**"semantics": {**

**"value": "Point",**

**"config": {**

**"isPointOf": "ZWave\_Node\_003"**

**}**

**}**

**},**

**"editable": true,**

**"type": "Number",**

**"name": "ZWave\_Node\_003\_Sensor\_relative\_humidity",**

**"label": "Sensor (relative humidity)",**

**"category": "Humidity",**

**"tags": [**

**"Point"**

**],**

**"groupNames": [**

**"ZWave\_Node\_003"**

**]**

**}**

**]**

### Beispiel – Response Temperature

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte Beschreibung

### Code Beispiel: API – Zugriff Python

import requests

def get\_temperature():

    url = "http://openhabian:8080/rest/items/ZWave\_Node\_003\_Sensor\_temperature"

    try:

        response = requests.get(url)

        if response.status\_code == 200:

            data = response.json()

            temperature = data['state']

            return temperature

        else:

            print("Failed to retrieve data. Status code:", response.status\_code)

            return None

    except Exception as e:

        print("Error occurred:", str(e))

        return None

# Example usage

temperature = get\_temperature()

if temperature is not None:

    print("Current temperature:", temperature)

### Code Beispiel: API – Zugriff R

# Check if the package is already installed

if (!requireNamespace("httr", quietly = TRUE)) {

  # If not installed, install it

  install.packages("httr")

}

library(httr)

get\_temperature <- function() {

  url <- "http://openhabian:8080/rest/items/ZWave\_Node\_003\_Sensor\_temperature"

  tryCatch({

    response <- GET(url)

    if (status\_code(response) == 200) {

      data <- content(response, "parsed")

      temperature <- data$state

      return(temperature)

    } else {

      cat("Failed to retrieve data. Status code:", status\_code(response), "\n")

      return(NULL)

    }

  }, error = function(e) {

    cat("Error occurred:", conditionMessage(e), "\n")

    return(NULL)

  })

}

# Example usage

temperature <- get\_temperature()

if (!is.null(temperature)) {

  cat("Current temperature:", temperature, "\n")

}