

# REST-full API für den Raspi

## Genrelles Verhalten

### Sensor-Daten

Der Raspi muss einen HTTP-REST Server zur Verfügung stellen, über den verschiedene Daten abgefragt werden können.

Wenn eine Anfrage kommt, dann liest der Raspi die Sensordaten aus, bringt sie in das richtige Format (heißt die Daten von den Sensoren müssen entsprechend umgerechnet werden)

Sensor	Zielwert	Kommentar
Licht	Lumen	ganzzahlig von 0 bis ?
Luftfeuchtigkeit	Prozent	ganzzahlig von 0 bis 100
Bodenfeuchtigkeit	Prozent	ganzzahlig von 0 bis 100
Temperatur	Grad	ganzzahlig
Größe	cm	ganzzahlig (über Kamera ermitteln?)

### History

Der Raspi soll nicht nur die Sensor-Daten auslesen, wenn er eine REST-Anfrage bekommt. Er soll zusätzlich alle 15 Minuten die Sensor-Daten abfragen und in einer Historie speichern.

Empfehlenswert sind wahrscheinlich einfache JSON-Dateien, in die die Historie so gespeichert wird, wie sie auch vom REST-Client abgefragt werden kann.

Wenn ein Topf nicht bepflanzt ist, soll auch keine Historie abrufbar sein. Heißt, wenn ein Topf leer wird, kann ebenfalls die Historie gelöscht werden, um Speicherplatz zu sparen.

Ein JSON-History File könnte z.B. so hier aussehen:

```
{
  "plantedAt": unix timestamp, // Zeitpunkt der Einpflanzung
  "history": [ // Befüllen von vorne nach hinten, also letzter Wert im Array ist der aktuellste
    {
      "createdAt": unix timestamp, // Zeitpunkt an dem die Messwerte erfasst worden
      "light": 0-99999, // Wert in Lumen, heißt 0..
      "airHumidity": 0-100, // in Prozent, ganzzahlig
      "soilHumidity": 0-100, // in Prozent, ganzzahlig
      "temperature": -20-100, // in Grad, ganzzahlig
      "size": 8, // Höhe der Pflanze in cm, über Kamera ermitteln => optional
    },
    // ...
  ]
}
```

Der Raspi muss selbstständig erkennen können, ob eine Topf bepflanzt oder leer ist.

- Wechselt sich der Zustand von leer zu bepflanzt, muss eine neue History angelegt werden (ganz wichtig: "plantedAt" muss dann gefüllt werden)
- Wechselt der Zustand von bepflanzt zu leer, dann soll die entsprechende Historie gelöscht werden

## API-Endpunkte

Der Raspi soll eine HTTP-REST API zur Verfügung stellen. Die folgenden Endpunkte sollen unterstützt werden:

### GET '/'

Diese Route soll die aktuellen Daten der Sensoren auslesen, aufbereiten und zurückgeben.

```
{
  "timestamp": unix timestamp, // timestamp wann die Daten ausgelesen wurden, also JETZT
  "pots": [
    {
      "id": 1|2|3, // die Nummer des Pots
      "used": true|false, // ob er gerade genutzt wird oder nicht

      // ab hier nur ausgeben, wenn der Pot used ist ?
    }
  ]
}
```

```

    "plantedAt": unix timestamp, // Zeitpunkt der Einpflanzung
    "light": 150, // Wert in Lumen, heißt 0..
    "airHumidity": 0-100, // in Prozent, ganzzahlig
    "soilHumidity": 0-100, // in Prozent, ganzzahlig
    "temperature": -20-100, // in Grad, ganzzahlig
    "size": 8, // Höhe der Pflanze in cm, über Kamera ermitteln => optional
  },
  // ...
]
}

```

## GET '/history'

Dieser Endpunkt gibt die History der Daten zu allen Töpfen aus.

```

{
  "pots": [ // ein Eintrag je Pot
    {
      "id": 1|2|3, // die Nummer des Pots
      "used": true|false, // ob er gerade genutzt wird oder nicht

      // ab hier sind es eins zu eins die Daten aus der jeweiligen Historie
      "plantedAt": unix timestamp, // Zeitpunkt der Einpflanzung
      "history": [ // Befüllen von vorne nach hinten, also letzter Wert im Array ist der aktuellste
        {
          "createdAt": unix timestamp, // Zeitpunkt an dem die Messwerte erfasst worden
          "light": 0-99999, // Wert in Lumen, heißt 0..
          "airHumidity": 0-100, // in Prozent, ganzzahlig
          "soilHumidity": 0-100, // in Prozent, ganzzahlig
          "temperature": -20-100, // in Grad, ganzzahlig
          "size": 8, // Höhe der Pflanze in cm, über Kamera ermitteln => optional
        },
        // ...
      ]
    },
    // ...
  ]
}

```

## GET '/history/{id}'

Gibt die History zu einem bestimmten Pot aus.

Wenn keine History vorhanden:

```

null

```

Wenn History vorhanden:

```

{
  "plantedAt": unix timestamp, // Zeitpunkt der Einpflanzung
  "history": [ // Befüllen von vorne nach hinten, also letzter Wert im Array ist der aktuellste
    {
      "createdAt": unix timestamp, // Zeitpunkt an dem die Messwerte erfasst worden
      "light": 0-99999, // Wert in Lumen, heißt 0..
      "airHumidity": 0-100, // in Prozent, ganzzahlig
      "soilHumidity": 0-100, // in Prozent, ganzzahlig
      "temperature": -20-100, // in Grad, ganzzahlig
      "size": 8, // Höhe der Pflanze in cm, über Kamera ermitteln => optional
    },
    // ...
  ]
}

```