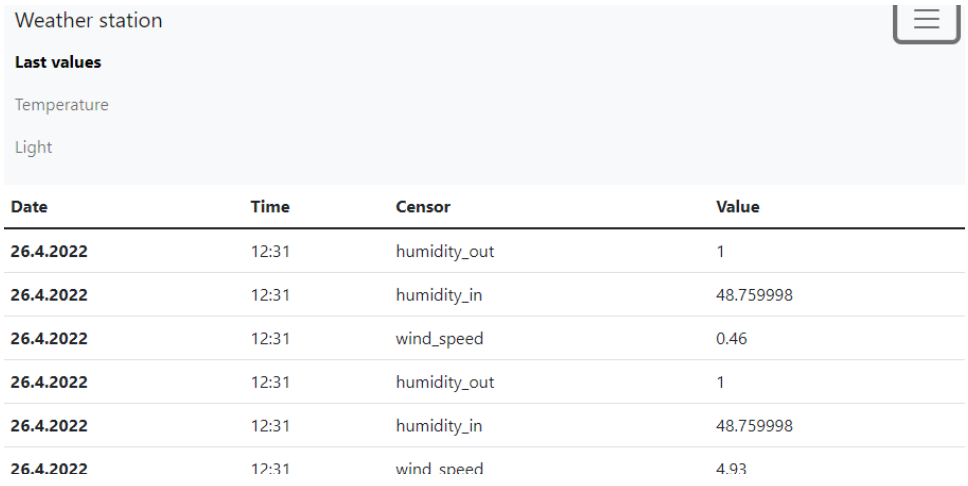
Sääasema projekti

Näkymä 1

Ensimmäisessä näkymässä on 50 viimeisintä mittausta kaikista sensoreista. Mittauksissa lukee päivämäärä, kellon aika, mikä sensori ja mittauksen arvo.

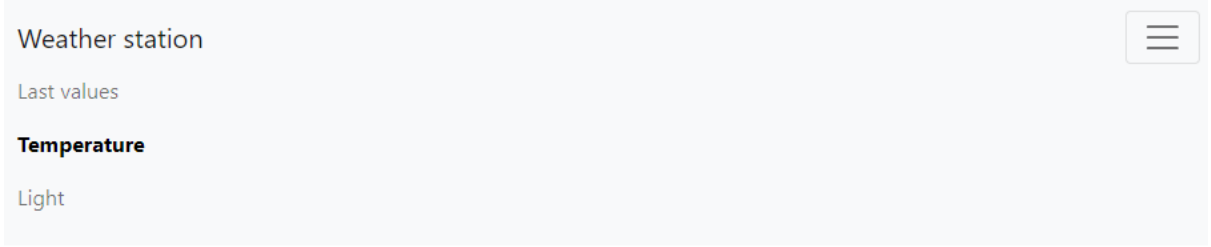


Date	Time	Censor	Value
26.4.2022	12:31	humidity_out	1
26.4.2022	12:31	humidity_in	48.759998
26.4.2022	12:31	wind_speed	0.46
26.4.2022	12:31	humidity_out	1
26.4.2022	12:31	humidity_in	48.759998
26.4.2022	12:31	wind_speed	4.93

```
const BASE_API = "https://webapi19sa-1.course.tamk.cloud/v1/weather/"
const LAST_50_DATA = BASE_API + "limit/50"

/*
Fetch 50 latest
{
"id":61387030,
"device_id":"ICT_2018",
"date_time":"2022-04-15T09:33:04.451Z",
"data":{
 "humidity_out":1
}
},
*/
const fetch50Latest = () => {
    return fetch(LAST_50_DATA).then(response => response.json()).then(data => {

        return data.map(d => {
            const { data, date_time } = d;
            let sensorName = Object.keys(data)[0];
            let sensorData = data[sensorName];

            return {
                ...parseStrToDateTime(date_time),
                sensorName,
                sensorData,
            }
        })


    })
}
```

Näkymä 2

Näkymässä 2 on lämpösensorin mittaukset. Siinä näkyy mittauksen päivämäärä, kellon aika ja mittauksen arvo.

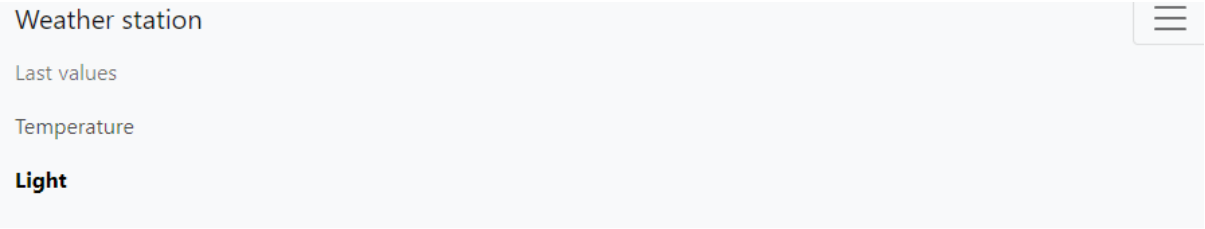


Date	Time	Value
26.4.2022	12:41	16.5
26.4.2022	12:41	16.5
26.4.2022	12:41	16.5
26.4.2022	12:41	16.59
26.4.2022	12:41	16.639999
26.4.2022	12:41	16.5

```
/*
Fetch by single sensor
{"device_id":"ICT_2018","date_time":"2022-04-15T11:47:10.596Z","temperature":"16.74"}
*/
const fetchBySensor = (sensorName) => {
    return fetch(BASE_API + sensorName).then(response => response.json()).then(data => {

        return data.map(d => {
            let sensorData = d[sensorName]

            return {
                ...parseStrToDateTime(d.date_time),
                sensorData
            }
        })
    })
}
```

Näkymä 3

Viimeisessä näkymässä on valo sensorin mittaukset. Se toimii samanlailla kuin lämpötilan näkymä.



Date	Time	Value
26.4.2022	12:46	79.269997
26.4.2022	12:46	79.269997
26.4.2022	12:46	79.120003
26.4.2022	12:46	79.120003
26.4.2022	12:46	79.120003
26.4.2022	12:46	79.120003

CSS/Bootstrap

```
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

<!-- Option 1: Bootstrap Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
    crossorigin="anonymous"></script>
<title>Sääasema</title>
<style>
    .active-item {
        color: □black !important;
        font-weight: bold;
    }
</style>
```

```
<!-- Boostrapin valmis template -->
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">Weather station</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarTogglerDemo02"
            aria-controls="navbarTogglerDemo02" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarTogglerDemo02">
            <ul id="ul-navbar" class="navbar-nav me-auto mb-2 mb-lg-0">
                <li class="nav-item">
                    <a class="nav-link active-item" id="index" href="#" onclick="updateView(this)">Last values</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" id="temperature" href="#" onclick="updateView(this)">Temperature</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" id="light" href="#" onclick="updateView(this)">Light</a>
                </li>
            </ul>
        </div>
    </div>
</nav>
<div class="table-container">
    <table class="table">
        <thead id="table-header">
        </thead>
        <tbody id="table-body">
        </tbody>
    </table>
</div>
```

Muita asioita

Muutetaan API:sta saatu string päivämääräksi.

```
/*
Get date and hour from string returned from api
*/
const parseStrToDateTime = (str) => {
    let dateObj = new Date(str)
    let date = dateObj.getDate() + "." + (parseInt(dateObj.getMonth()) + 1) + "." + dateObj.getFullYear()
    let hour = dateObj.getHours() + ":" + dateObj.getMinutes().toString().padStart(2, '0');
    return {
        date,
        hour
    }
}
```

Näkymä yksittäiselle sensorille.

```
/*
Create view for single sensor
*/
const createSensorDataView = async (sensorName) => {
    let data = await fetchBySensor(sensorName)

    let columnData = `
     <tr>
        <th scope="col">Date</th>
        <th scope="col">Time</th>
        <th scope="col">Value</th>
    </tr>
    `



    let tableHeaderContainer = document.getElementById("table-header")
    tableHeaderContainer.innerHTML = columnData

    let htmlStr = ""
    data.forEach(d => {
        const { date, hour, sensorData } = d;

        let html = (`<tr>
            <th>${date}</th>
            <td>${hour}</td>

            <td>${sensorData}</td>
        </tr>`)

        htmlStr += html;

    })
    let tableRowContainer = document.getElementById("table-body");
    tableRowContainer.innerHTML = htmlStr;

}
```

Näkymä indexille

```
/*
Create index page view
*/
const createIndexView = async () => {
    let data = await fetch50Latest();


    let columnData = `
     <tr>
        <th scope="col">Date</th>
        <th scope="col">Time</th>
        <th scope="col">Censor</th>
        <th scope="col">Value</th>
    </tr>
    `


    let tableHeaderContainer = document.getElementById("table-header")
    tableHeaderContainer.innerHTML = columnData

    let htmlStr = ""
    data.forEach(d => {
        const { date, hour, sensorName, sensorData } = d;

        let html = (`<tr>
            <th>${date}</th>
            <td>${hour}</td>
            <td>${sensorName}</td>
            <td>${sensorData}</td>
        </tr>`)

        htmlStr += html;

    })
    let tableRowContainer = document.getElementById("table-body");
    tableRowContainer.innerHTML = htmlStr;


}
```