# Web-Tech Project – Smart Home Manager

Authors: Dielacher Manuel / Macher Markus / Freislich Philipp

Table of contents

Inhalt

[Web-Tech Project – Smart Home Manager 1](#_Toc209695)

[Overview 2](#_Toc209696)

[Testing 2](#_Toc209697)

[Functionality 2](#_Toc209698)

[Installation instructions 3](#_Toc209699)

[Required software 3](#_Toc209700)

[Installation 3](#_Toc209701)

[Start the app 3](#_Toc209702)

## Overview

The „Smart Home Manager“ is an web based application to visualize and manage smart home devices.

The typical use case would be that each family member can see the current state of each device (ON/OFF) and toggle them.

On the devices page of the app you would usually create a room and add all smart home devices that are in that room. You can create as many rooms and devices as want to. Rooms and devices can be edited and deleted.

At the current state of the application each user can see all devices so there is no point for different user accounts as of now, but you are able to change your email and username.

## Testing

For testing the back-end / API emulates the devices and saves their statues but not permanently so if you restart the app the statuses will be back at their defaults.

For the test mode to work you will need the provided database because the test-mode has those already in it, but if you remove/add devices it will update those.

## Functionality

The database stores all the relevant device, room and user data. It does not hold the statues of the devices since they can change from outside of the application.

The back-end / API manages the database and user requests.

It provides the front-end with the device data from the database and updates the database by user request.

When a user is logged in and views the devices page the server will start a kind of service that checks all the devices that are in the database every second and saves their state temporarily in the memory. On the client side when viewing the devices page all devices will be fetched from the server and listed according to their rooms on the devices page and it also asks the server to send the device statues so it can update the view for the user every second.

The devices component creates a device for each device in the database. Then the device component checks what kind of device should be shown and picks the right child component to view it.

When the user leaves the devices page the “service” on the client for fetching the statues every second stops and on the server side the service stops as well when there is no user requesting device statues data for 5 seconds.

## Installation instructions

### Required software

* A MySQL Database (InnoDB) (recommended Oracle MySQL)
* A Database management tool (recommended MySQL Workbench)
* Node.js

### Installation

1. First unzip the project folder where ever you want
2. Setup the Database

* A configured MySQL Database with root access
* To create the database structure run the provided sql script “create\_SHM\_DB.sql”

You will find it in the project folder in the subfolder “root\mysql”

* To create the needed database user run the provided script “create\_DB\_User.sql”

You will find it also in the “root\mysql” subfolder

Or you could create your own user the default user connection information is in the “mysql\_connection.txt” also in “root\mysql” subfolder

If you want to use other user information make sure to set them in the “root\server\config\config\_db.json”

1. Install the node modules

* Open a new PowerShell or command prompt and navigate to the root of the unzipped project folder and run the “npm install” command

This will install all required modules for the front-end “Angular”

* Open a new PowerShell or command prompt and navigate to “root\server” and run “npm install” again to install all back-end modules “express server”

## Start the app

1. You could start both servers by starting the start.bat or start.ps1 file in the root folder

the powershell one might not work because of execution policy restrictions

or

1. Start the front and back-end server

* Open a new PowerShell or command prompt and navigate to “root\server” and run “node server.js” to start the backend express server
* Open a new PowerShell or command prompt and navigate to the root of the project and type “ng serve -o” to start the angular cli test server
* After the angular cli finished compiling it will open an new webpage with the application