

Katzenfütterungsanlage

**HTBLA Kaindorf an der Sulm
Grazer Straße 202, A-8430 Kaindorf an der Sulm
Ausbildungsschwerpunkt Mechatronik und Automatisierungstechnik**

Florian Greistorfer, Florian Harrer, Dominik Pichler, Julian Wolf

Abgabedatum: 05.04.2018

Betreut von:
Dipl.-Ing. Manfred Steiner
Dipl.Ing. Dr. Gerhard Pretterhofer
Otto

Inhaltsverzeichnis

1	Webserver und Client	1
1.1	Begriffserklärungen	1
1.1.1	Server	1
1.1.2	Client	1
1.2	Anforderungen	1
1.2.1	Webserver	1
1.2.2	Client	1
1.3	Voruntersuchung	2
1.3.1	Typescript	2
1.3.2	Node.js	2
1.3.3	Angular 2/4	2
1.3.3.1	Modules	2
1.3.3.2	Components	2
1.3.3.3	Templates	2
1.3.3.4	Data binding	2
1.3.3.5	Services	2
1.3.4	Bootstrap	2
1.4	Umsetzung	3
1.4.1	Client	3
1.4.1.1	Design	3
1.4.1.2	Funktion	4
1.4.2	Server	4
1.4.2.1	Funktion	4
1.4.2.2	Mongodb	4
1.4.2.3	Kommunikation mit dem Java Programm	4
1.5	Zusammenfassung und Verbesserungsmöglichkeiten	4
A	Abbildungsverzeichnis	7
B	Tabellenverzeichnis	9
C	Listings	11

KAPITEL 1

Webserver und Client

1.1 Begriffserklärungen

1.1.1 Server

Ein Computer oder Programm, der oder das Zugriff auf eine Resource oder einen Dienst in einem Netzwerk ermöglicht

1.1.2 Client

Ein Computer oder Programm, der oder das auf einen Server Zugreift

1.2 Anforderungen

1.2.1 Webserver

Auf der Katzenfütterungsanlage läuft ein Webserver, der es ermöglicht, dass der Benutzer das Gerät über das Internet erreichen kann. Hauptaufgaben des Servers sind dabei, Daten bereitzustellen, zu verarbeiten und zu speichern und den Webclient zur Verfügung zu stellen.

1.2.2 Client

Der Client soll dem Benutzer ermöglichen, die Katzenfütterungsanlage über einen Webbrowser zu steuern. Ein Benutzername und ein Passwort sind erforderlich, damit man das Gerät bedienen kann. Das Design soll eindeutig und übersichtlich gehalten sein. Auf der Startseite sollen die eingestellten Fütterungszeiten zu sehen sein und eine allgemeine Übersicht. Über eine Navigationsleiste sollen die weiteren Seiten erreichbar sein:

- Fütterungszeiten
- Positionsinfo
- Geräteinfo
- Update

1.3 Voruntersuchung

1.3.1 Typescript

Typescript ist eine Weiterentwicklung der Sprache Javascript, die strenge Datentypen hat. Typescript muss von einem Transpiler (=Übersetzer) in Javascript übersetzt werden. Javascript kann direkt von jedem herkömmlichen Browser ausgeführt werden.

1.3.2 Node.js

Node.js ist eine Laufzeitumgebung, die es ermöglicht, dass Javascript direkt auf einem Rechner ausgeführt werden kann.

1.3.3 Angular 2/4

Angular ist ein Typescript Framework, das aus dem Javascript-Framework AngularJS weiterentwickelt wurde. Es wird von Google entwickelt. Angular ist gegliedert in Komponenten. Die grobe Struktur wird in der Abbildung 1.1 dargestellt.

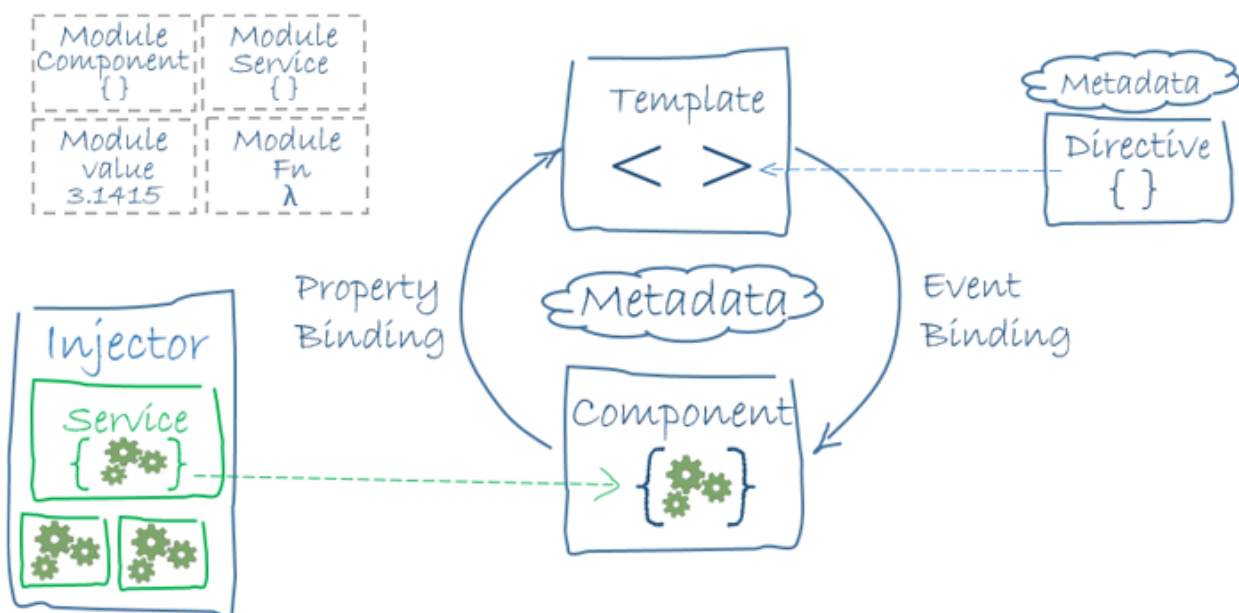


Abbildung 1.1: Angular Struktur

1.3.3.1 Modules

1.3.3.2 Components

1.3.3.3 Templates

1.3.3.4 Data binding

1.3.3.5 Services

1.3.4 Bootstrap

Bootstrap

1.4 Umsetzung

1.4.1 Client

1.4.1.1 Design

Das Design sollte übersichtlich und einfach gestaltet werden. Der Benutzer soll auf den ersten Blick die wichtigsten Funktionen und Informationen erkennen können.

Auf der Startseite sind alle wichtigen Informationen übersichtlich dargestellt. Auf der linken Seite werden die Uhrzeit der letzten erfolgreichen Fütterung, die Zeit der nächsten Fütterung und die Zeit bis zur nächsten Fütterung dargestellt. Darunter werden Fehler und Warnungen, falls welche auftreten sollten, angezeigt. Da unbekannt ist, wie viele Fehler und Warnungen auftreten, werden



Abbildung 1.2: Startseite

diese in einem *ngFor aufgelistet. Auf der rechten Seite sind die aktiven Fütterungszeiten aufgelistet.

Listing 1.1: Errors and Warnings in einem *ngFor

```

1 <ul style="list-style-type: none; margin-left: -40px;">
2   <li *ngFor="let warning_message of ErrorsAndWarnings.Warnings">
3     <div class="alert alert-warning alert-dismissible fade show" [hidden]="
4       warning_message.hidden">
5       <button type="button" class="close">
6         <span aria-hidden="true" (click)="ack(warning_message)">&times;</span>
7       </button>
8       <strong i18n>Warning! </strong>{{warning_message.message}}
9     </div>
10  </li>
11  <li *ngFor="let error_message of ErrorsAndWarnings.Errors">
12    <div class="alert alert-danger alert-dismissible fade show" [hidden]="
13      error_message.hidden">
14      <button type="button" class="close">
15        <span aria-hidden="true" (click)="ack(error_message)">&times;</span>
16      </button>
17      <strong i18n>Error! </strong>{{error_message.message}}
18    </div>
19  </li>

```

18

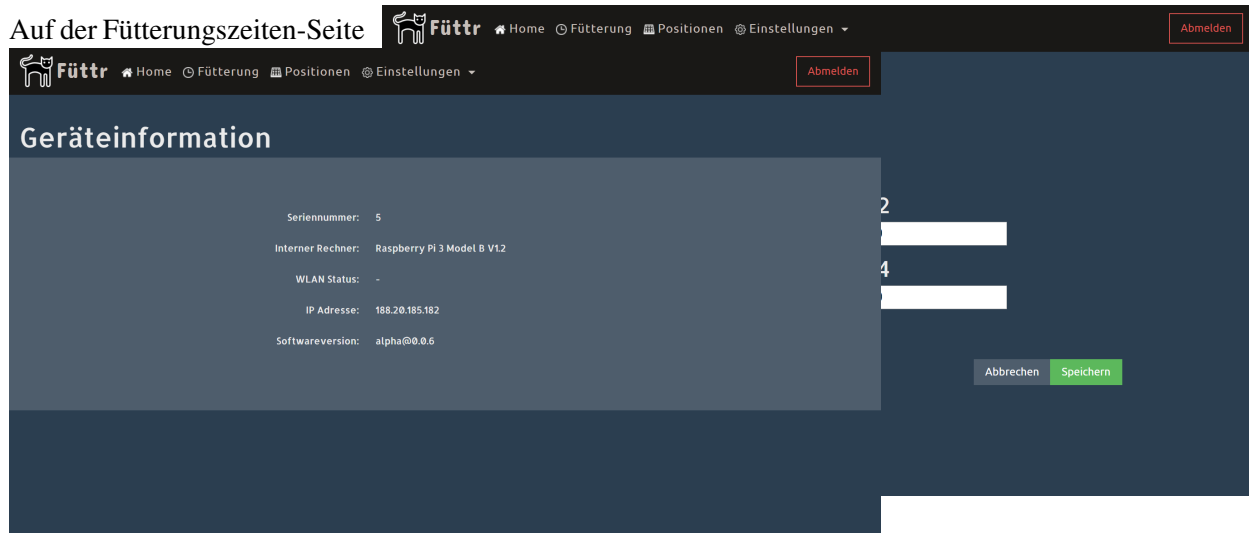


Abbildung 1.3: Fütterungszeiten

Abbildung 1.4: Geräteinformationen

1.4.1.2 Funktion

1.4.2 Server

1.4.2.1 Funktion

1.4.2.2 MongoDB

1.4.2.3 Kommunikation mit dem Java Programm

1.5 Zusammenfassung und Verbesserungs- möglichkeiten

Anhang

ANHANG A

Abbildungsverzeichnis

1.1	Angular Struktur	2
1.2	Startseite	3
1.3	Fütterungszeiten	4
1.4	Geräteinformationen	4

ANHANG **B**

Tabellenverzeichnis

ANHANG C

Listings

1.1	Errors and Warnings in einem *ngFor	3
-----	---	---