

Entwurfsbeschreibung  
Nachrichtenkommunikation für das THW

na17b

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>2</b>
<b>2</b>	<b>Produktübersicht</b>	<b>2</b>
<b>3</b>	<b>Grundsätzliche Struktur</b>	<b>2</b>
<b>4</b>	<b>Struktur einzelner Pakete</b>	<b>2</b>
<b>5</b>	<b>Datenmodell</b>	<b>3</b>
	<b>Glossar</b>	<b>4</b>

## 1 Allgemeines

Ziel des Vorprojekts ist es, eine Anwendung zu schreiben, welche es erlaubt, einen Vierfach-Vordruck abzufassen und anzuzeigen sowie den damit verbundenen Workflow zu anzudeuten.

## 2 Produktübersicht

Im oberen Bereich befindet sich eine Kopfleiste mit THW-Farben und -Logo, welche eine von zunächst drei verfügbaren Rollen anzeigt. Am linken Bildrand befindet sich ein Menü, welches den Wechsel zwischen Übersicht, Maske zur Erstellung eines neuen Formulars, sowie der Rollenauswahl ermöglicht.

Die Übersichtsseite vermittelt einen schnellen Überblick über alle hinterlegten Vordrucke, eingeschränkt auf Verfasser, Datum, Uhrzeit und eine gekürzte Version des Inhalts. In der ersten Spalte befindet sich ein Button, über den später der Vordruck zur genaueren Ansicht bzw. Bearbeitung ausgewählt werden können soll.

Die Formularmaske bildet den Vierfachvordruck realitätsnah nach. Die Leitrichtung (eingehend oder ausgehend) kann über einen Switch am oberen Rand ausgewählt werden und hat Einfluss auf die verfügbaren Felder. Am unteren Rand befindet sich ein provisorischer 'Abschicken'-Knopf, welcher die eingegebenen Daten zunächst nur temporär abspeichert. Es findet derzeit auch noch keine Überprüfung der Daten hinsichtlich Konsistenz und Vollständigkeit statt.

Die Rollenauswahl beschränkt sich im Vorprojekt auf ein simples Dropdown-Menü und dient nur zur technischen Demonstration.

## 3 Grundsätzliche Struktur

Der Quelltext der Anwendung ist unterteilt in Frontend und Backend. Um allen Entwicklern eine konsistente Arbeitsumgebung bereitzustellen, wird hierzu Docker verwendet. Die Konfigurationsdateien für Docker sind im Verzeichnis `docker` abgelegt. Diese müssen nicht manuell verwendet werden, stattdessen steht für die Installation der Container ein Skript `install.sh` bereit, welches einmalig ausgeführt werden muss.

## 4 Struktur einzelner Pakete

### Frontend

Das Frontend ist in Javascript geschrieben und verwendet das Vue.js-Framework. Als Erweiterungen kommen Vuex, sowie vue-router zum Einsatz. Die Verwaltung des Quelltextes sowie das kompilieren geschieht via webpack. Zur Vereinfachung wird auf die Bibliothek von Element-UI zurückgegriffen.

Der Quelltext des Frontends befindet sich im Ordner `src/frontend`. In diesem Ordner befinden sich neben einem weiteren Ordner `src` die von webpack benötigten Dateien

und Ordner; diese werden im Folgenden nicht weiter beschrieben und sind weitgehend Standard.

Der Ordner `src` unterteilt sich weiter in folgende Struktur:

```
./src/frontend/src/  
├── assets/  
├── components/  
├── router/  
├── store/  
│   ├── state.js  
│   ├── getters.js  
│   ├── mutations.js  
│   └── index.js  
├── App.vue  
└── main.js
```

- `assets` Mediendateien (aktuell nur das THW-Logo)
- `components` Vue-Komponenten
- `router` Konfiguration von vue-router
- `store` Konfiguration von Vuex
  - `state.js` deklariert globale Zustandsvariablen
  - `getters` stellt Funktionen zum Abrufen der Variablen bereit
  - `mutations` enthält Funktionen zum Bearbeiten des Zustandes
  - `index.js` fügt die obigen Dateien zusammen zu einem globalen Store
- `App.vue` Wrapper-Komponente für das Frontend
- `main.js` Einstiegspunkt für das Programm, enthält alle Importe

## 5 Datenmodell

Für das Frontend existiert kein Datenmodell, dieses entsteht erst mit der Implementierung des Backends.

## Glossar

**Element-UI** Bibliothek vorgefertigter Komponenten für Vue.js. 2

**Javascript** Skriptsprache, die hauptsächlich auf Webseiten Anwendung findet und zu-  
meist clientseitig ausgeführt wird, um den Server zu entlasten. 2, 4

**vue-router** Ermöglicht die Verwendung der Browserhistorie, sowie das gezielte Anzei-  
gen von Komponenten abhängig vom Pfad. 2

**Vue.js** Reaktives Javascript-Framework zur einfachen Erstellung interaktiver Frontends.  
2, 4

**Vuex** Erweitert Vue.js um einen globalen Speicher für Zustandsvariablen, sowie Funk-  
tionen zum einfacheren Bearbeiten und Methoden zum Debuggen. 2

**webpack** Build-Tool für Webseiten, hauptsächlich für Javascript. Erlaubt das Verwen-  
den nahezu beliebiger Dateiformate und Sprachstandards für den Entwickler und  
erzeugt eine mit allen gängigen Browsern kompatible und minimierte Webseite. 2