

Entwurfsbeschreibung  
Nachrichtenkommunikation für das THW

na17b

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>1</b>
<b>2</b>	<b>Produktübersicht</b>	<b>1</b>
<b>3</b>	<b>Grundsätzliche Struktur</b>	<b>1</b>
<b>4</b>	<b>Struktur- und Entwurfsprinzipien einzelner Pakete</b>	<b>1</b>
4.1	Frontend . . . . .	1
<b>5</b>	<b>Datenmodell</b>	<b>2</b>
<b>6</b>	<b>Glossar</b>	<b>2</b>
6.1	Docker . . . . .	2
6.2	Wrapper . . . . .	2

# 1 Allgemeines

Die Anwendung soll den gegebenen Nachrichtenfluss in einer Notfallzentrale des THW nachbilden.

## 2 Produktübersicht

Die Anwendung besitzt auf der linken Seite ein Auswahlmenü, welches aus- und eingeklappt werden kann. Über dieses Menü gelangt man entweder zur Listenanordnung der schon erstellten Vordrucke, zur Erstellung eines neuen Vordrucks oder zur Rollenauswahl. In der Liste der Formulare ist für das jeweilige Dokument das Kürzel des Erstellers, sowie Erstellungsdatum und -zeit und außerdem eine kurze Zusammenfassung des Inhalts zu sehen. Bei der Erstellung des Formulars sind alle Eingabefelder ähnlich dem Original angeordnet. Die Rollenauswahl-Seite besteht aus einem Dropdown-Menü mit den unterschiedlichen besetzbaren Rollen. Bei Auswahl wird der Rollename in der linken oberen Ecke der Seite im Titel verändert.

### 3 Grundsätzliche Struktur

Der Quelltext der Anwendung ist service-orientiert aufgebaut. Hierzu werden Container verwendet welche die jeweiligen Services kapseln. Zwingende Voraussetzung in der Entwicklungsumgebung ist:

- Docker
  - Bevor die Anwendung ausgeführt werden kann müssen auf dem ausführenden Computer (Host) die folgenden Programme installiert sein:
    - \* docker
    - \* docker-compose
  - Die Anwendung ist auf zwei Docker-Container verteilt:
    - \* Server- Container der das Frontend und Controller-Logik implementiert
    - \* Daten-Container der den RDF-Endpoint implementiert

## 4 Struktur- und Entwurfsprinzipien einzelner Pakete

## 4.1 Frontend

Ausgehend von `na17b/Project/src/Frontend`.

## Frontend - Arbeitsverzeichnis von Webpack

**l**\_src - Quelltext

`_assets` - enthält Bilder, Videos, ... (aktuell nur THW-Logo)

- components - Komponenten (Bauteile), die Oberfläche wird (bis auf wenige begründete Ausnahmen) mit Elementen aus der Vue-Bibliothek Element-UI gebaut

- Layout.vue: legt das Design der Seite fest (Header, Menüleiste)

- └─<router-view> - Hauptbereich, dadurch wird vom Router die passende Komponente 'injiziert'

```
_router
```

- index.js - legt fest, wann welche Ansicht angezeigt wird

store - Globale Daten

-App.vue - ist ein Wrapper, muss nicht geändert werden (Existenz nur aus technischen Gründen)

`_main.js` - Einstiegspunkt, hier sind alle zu importierenden Komponenten definiert, die dann automatisch geladen werden

Frontend - Arbeitsverzeichnis von Webpack

└─ dist - Kompilierte und minimierte Webapp

└─ restliche - Konfiguration des Webpack oder node-module, wird in der Regel  
nicht angefasst

## 5 Datenmodell

Ist noch nicht Vorhanden

## 6 Glossar

### 6.1 Docker

Die Dockersoftware dient der Isolierung von Anwendungen. In diesem Projekt wird sie genutzt um eine plattformunabhängige Entwicklungsumgebung für die heterogenen System der Teammitglieder bereitzustellen. Hierzu wird die zur Entwicklung benötigte Software in einem Dockercontainer (virtuelles Betriebssystem) virtualisiert.

### 6.2 Wrapper

Software welche andere Software umgibt oder einbettet.