

Dienstplan

About

Diese Anwendung ist ein eigenständiger Webserver der zusammen mit einer MySQL-Datenbank betrieben wird.

Bei der ersten Ausführung wird der Server sich mit dem eingestellten Datenbank-Server verbinden, die Datenbank auswählen und wenn nicht vorhanden selbst erstellen, genauso wie jede Tabelle die benötigt wird.

Außerdem werden zum Testen Mitarbeiter direkt hinzugefügt, damit man sich gleich einloggen kann.

Für den Administratorzugang:

- Username: admin
- Passwort: admin

Das Webpanel ist über den jeweiligen Port verfügbar, standardmäßig: `http://localhost:8080`

Es empfiehlt sich allerdings für externe Verbindungen https mithilfe eines Proxy einzurichten (wie z.B. nginx)

Decisions

Die meisten Abweichungen von der Planung sind durch vorherige Fehler in der Planungsphase oder Implementationslimitation der gewählten Sprache entstanden (z.B. die Node.js Umgebung erlaubt keine Circular Dependencies).

Wichtige Änderungen die daraus resultierten:

TODO

Getting Started

Zur Installation und zum Ausführen wird [Node \(https://nodejs.org\)](https://nodejs.org) und der dazugehörige [Node Package Manager \(https://www.npmjs.com/\)](https://www.npmjs.com/) benötigt, welcher mit Node mitgeliefert wird. Beide Tools sollten über die Pfadvariable des Systems verfügbar sein.

Setup

```
$ git clone https://github.com/IZEDx/SWT-17-18.git
$ cd Prototyp/
$ npm install
```

NPM installiert nun alle Dependencies für die Entwicklung, inklusive dem Typescript Compiler.

Beachte: Alle npm Commands müssen im "Prototyp" Ordner ausgeführt werden.

Build

```
Prototyp$ npm run build
```

Dies führt nun den Typescript Compiler aus, welcher das Projekt anhand der tsconfig.json zu Javascript übersetzt. Die fertigen Dateien sind im "dist" Ordner.

Run

```
npm start
```

Startet den Server mit den Einstellungen aus der config.json, sollte keine vorhanden sein werden die Standardeinstellungen aus der config-sample.json verwendet.

Configuration

Es empfiehlt sich die config-sample.json nicht direkt zu ändern sondern umzubenennen in config.json, damit die gespeicherten Einstellungen bei einem Update nicht überschrieben werden.