

Terminverwaltung App Projekt

Programmierung Java 2

09.07.2019

Ziel

- ▶ REST API Server als Backend für eine Web App
- ▶ Terminmanagement und Patientenverwaltung

Web App

Angular Projekt für Webaufbau - SS19

Web Server

- ▶ Spring Boot
- ▶ Spring Framework + Embedded HTTP(s) Servers - XML Configuration
- ▶ Ökosystem von Spring (Spring ORM, Spring Security, ...)

Web Server

- ▶ Tomcat als HTTP(s) Server
- ▶ Endpoints (Pfade):
 - ▶ /api/{model} (GET/POST)
 - ▶ /api/{model}/{id} (PUT/DELETE)
- ▶ Request Type: "application/json"
- ▶ Response Type: "application/json"
- ▶ Authorization: Basic Auth

Datenbank

- ▶ MySQL als Datenbank
 - ▶ MySQL Connector
- ▶ Spring Data JPA als ORM
 - ▶ Hibernate/JPA
 - ▶ Entity, Repository, ...

Modelle

- ▶ User
- ▶ Patient
- ▶ Address
- ▶ Appointment
- ▶ AppointmentRecord
- ▶ Prescription
- ▶ Medicine
- ▶ ...

Dokumentation

- API Reference: Models

Beispiel: /patient Endpoint

PATIENT-CONTROLLER

| | |
|---------------------|--------|
| getAllPatients | GET |
| addNewPatient | POST |
| uniqueCheck | POST |
| getPatient | GET |
| updatePatient | PUT |
| deletePatient | DELETE |
| addAppointment | POST |
| getAllPrescriptions | GET |
| addPrescription | POST |

Dokumentation

- ▶ OpenAPI (Swagger) Spezifikation
- ▶ Standarte Dokumentationsnotation

OpenAPI

- Hilft als offenes Beschreibungsformat dabei, den Überblick und das Verständnis über die Fähigkeiten eines API zu erhalten.

Automatisierte Dokumentation

- ▶ Springfox - <https://springfox.io>
- ▶ Kontinuierliche Integration von Code und Dokumentation

Springfox

- ▶ Bietet die Integration von Swagger(OpenAPI) für Spring und Spring Boot Projekte.
- ▶ Überwiegend wird die Dokumentation aus den bereits vorhandenen Spring-Annotationen generiert.
- ▶ Swagger Annotation sind zusätzlich möglich.

TODO

- ▶ Weitere Validierung (Mehr als @NotNull)