

## Softwarepraktikum – Beispiel-Web-Anwendung Sudoku

In den beiden Ordnern finden Sie eine Beispielimplementierung einer Sudoku-Web-Anwendung mit verschiedenen Frameworks für Backend (den Server) und Frontend (die Web-Oberfläche). Dabei wurden die Implementierungen so ähnlich wie möglich gehalten, sodass Sie sich speziell die Unterschiede zwischen den Frameworks anschauen können. Damit Sie den Code der Anwendung nachvollziehen können, wird empfohlen zunächst die Tutorials der Frameworks durchzugehen, die auf der Website des Softwarepraktikums (<https://www.informatik.uni-wuerzburg.de/is/lehre/wintersemester-2021-22/praktikum-softwarepraktikum/>) verlinkt sind.

### Backend

Der Server wurde in den folgenden Frameworks implementiert:

- Express (<http://expressjs.com>)
- FastAPI (<https://fastapi.tiangolo.com>)
- Flask (<https://flask.palletsprojects.com/en/2.0.x/>)
- Spring (<https://spring.io>)

Nach dem Start egal welcher Implementierung ist das Backend unter <http://localhost:8081> erreichbar und stellt die folgenden Schnittstellen zur Verfügung:

- /difficulties gibt die Schwierigkeiten zurück, die für ein Sudoku zur Wahl stehen
- /board lädt ein Sudoku für eine gewählte Schwierigkeit
- /verify prüft, ob ein Sudoku korrekt gelöst wurde und gibt andernfalls aus, wo noch ein Fehler existiert

### Frontend

Für die Web-Oberfläche wurden die folgenden Frameworks verwendet:

- Angular (<https://angular.io>)
- React (<https://reactjs.org>)
- Vue.js (<https://vuejs.org>)

Um mit allen Frameworks ein einheitliches Layout zu erhalten, wurde außerdem Bootstrap 4 (<https://getbootstrap.com>) sowie eine Verbindung von Bootstrap mit dem jeweiligen Framework (Angular: <https://ng-bootstrap.github.io>; React: <https://react-bootstrap-v4.netlify.app>; Vue.js: <https://bootstrap-vue.org>) verwendet.

Diese Web-Oberflächen starten jeweils unter dem Standard-Port des Frameworks und stellen die folgenden Funktionen zur Verfügung:

- Anzeigen der verfügbaren Schwierigkeiten
- Anzeigen eines Sudoku für eine ausgewählte Schwierigkeit
- Senden eines Sudokus zum Überprüfen an das Backend und Anzeige des Überprüfungsergebnisses

## Voraussetzungen

Um die Anwendung zu starten, muss folgendes installiert sein:

- Node.js 14 (z.B. von <https://nodejs.dev>)
- Python 3.9 (z.B. von <https://www.python.org>) mit pip (wird beim Installations-Programm von <https://www.python.org> automatisch installiert)
- OpenJDK 11 (z.B. von <https://adoptium.net>)

## Starten

Die Anwendung wird in allen Implementierungen über die Konsole gestartet. Dafür wird im Folgenden davon ausgegangen, dass Sie in der Konsole bereits im Ordner des jeweiligen Frameworks sind und dort die jeweiligen Befehle ausführen. Sie können jede beliebige Kombination aus Front- und Backend verwenden.

### - Backend

Da das Backend standardmäßig in allen Implementierungen immer auf demselben Port läuft, können Sie nur eines davon gleichzeitig starten.

- **Express**
  - Befehl (einmalig vor erstem Start): *npm ci*
  - Befehl: *npm start*
- **FastAPI**
  - Befehl (einmalig vor erstem Start): *pip install -r requirements.txt*
  - Befehl: *python sudoku.py*
- **Flask**
  - Befehl (einmalig vor erstem Start): *pip install -r requirements.txt*
  - Befehl: *python sudoku.py*
- **Spring**
  - Befehl: *./mvnw spring-boot:run*

### - Frontend

Das Frontend läuft jeweils auf dem Standard-Port der einzelnen Frameworks, sodass auch mehrere der Implementierungen gleichzeitig gestartet werden können.

- **Angular**
  - Befehl (einmalig vor erstem Start): *npm ci*
  - Befehl: *npm start*
  - Erreichbar unter <http://localhost:4200/>
- **React**
  - Befehl (einmalig vor erstem Start): *npm ci*
  - Befehl: *npm start*
  - Erreichbar unter <http://localhost:3000/>
- **Vue.js**
  - Befehl (einmalig vor erstem Start): *npm ci*
  - Befehl: *npm run serve*
  - Erreichbar unter <http://localhost:8080/>

## Übungsaufgaben

Wenn Sie die Verwendung der verschiedenen Frameworks üben möchten, können Sie für eine beliebige Implementierung (oder auch für Mehrere) der Sudoku-Web-Anwendung die folgenden Funktionen ergänzen (oder auch nur einen Teil davon):

- **Unterstützungs-Modus**, in welchem beim Eintragen einer Zahl alle anderen Vorkommen der Zahl hervorgehoben werden (mit anderer Farbe, falls es die Zahl in der aktuellen Zeile/Spalte bzw. im aktuellen Teil-Quadrat bereits gibt).
- **Speichern und Laden von Sudokus**, sodass ein Zwischenstand beim Lösen im Backend gespeichert und später wieder geladen werden kann; ein gespeicherter Zwischenstand soll auch wieder gelöscht werden können.
- **Sudoku-Editor** zum Anlegen eines eigenen Sudokus mit Prüfung, ob das angelegte Sudoku lösbar ist; für das angelegte Sudoku soll dann eine noch nicht existierende Bezeichnung für die Schwierigkeit vergeben werden und es anschließend zusätzlich zu den existierenden Schwierigkeiten geladen werden können.