

MidEng 7.1 Warehouse REST & Dataformats [GK] – 4h

Verfasser: **Danilo Stoilovski**

Datum: **24.09.2024**

GK-V

Einführung

Es steht die Nationalratswahl vor der Tür und um die Wahlergebnisse möglichst rasch vom Wahllokal an die nächste Bezirksstelle übertragen zu können, soll eine REST Schnittstelle implementiert werden, wo die Ergebnisse (Anzahl der Stimmen pro Partei) dieses Wahllokals ausgegeben werden. Um die Möglichkeiten der Anbindung flexibel zu gestalten, sollen die Datenformate JSON und XML unterstützt werden.

Daten-Generierung beim Wahllokal (inkl. Standortdaten)

Hierfür erweitere ich die Klasse `ElectionData`, indem ich eine Methode hinzufüge, die mir die fest encodierten Daten zurückgibt.

Entwicklung einer REST Schnittstelle

Um das automatische Kompilieren in XML in meiner REST-Schnittstelle zu ermöglichen, musste ich in den Gradle-Abhängigkeiten die folgende Zeile hinzufügen:

```
implementation 'com.fasterxml.jackson.dataformat:jackson-data'
```

Darüber hinaus musste ich nur den Code so anpassen, dass meine `ElectionSimulation`-Klasse aufgerufen wird. Dies habe ich wie folgt umgesetzt:

```
@RequestMapping(value = "/election/{id}/xml", produces = MediaType.  
    public ElectionData getElectionDataXML(@PathVariable int id)
```

```
        return electionService.getElectionData(id);  
    }
```

Empfangen der Daten bei einem Controller mittels HTTP Request / Controller veröffentlicht Daten (Standortdaten, Auszählungsergebnis) per Client Request

Hierfür habe ich vorerst zwei HTTP Endpunkte definiert an welchen meinen HTTP-Requestst verarbeitet werden sollen.

Beide Methoden akzeptieren eine HTTP-GET-Anfrage für den Pfad

`/election/{id}/{format}` und verwenden den `@PathVariable`, um die `id` dynamisch aus der URL zu extrahieren.

Die Annotation

`@RequestMapping` verwendet das Attribut `produces`, um festzulegen, welches Medienformat (JSON oder XML) an den Client zurückgegeben wird.

```
produces = MediaType.APPLICATION_JSON_VALUE
```

Neu für mich waren die folgenden Annotationen

- **@RestController**: Kennzeichnet eine Klasse als Controller, die RESTful Webservices bereitstellt. Methoden innerhalb dieser Klasse geben in der Regel JSON- oder XML-Daten zurück.
- **@RequestMapping**: Wird verwendet, um HTTP-Anfragen mit einer bestimmten URL einem Controller oder einer Methode zuzuordnen.