

ST2-Workshop am 30.06.20 und 05.07.2021

ST2-Workshop am 30.06.20 und 05.07.2021

In diesem Workshop werden wir uns mit der Implementierung von REST-APIs beschäftigen.

Repos

Hierfür dient wieder das "Textadventure" als Übungsbeispiel.

- Sie finden ein als **Aufgabenstellung** ein vorbereitetes Repo mit noch nicht implementierten Controller-Methoden hier: <https://git.st.archi-lab.io/students/st2/ss21/textadventure-example/textadventure-v4a.git>.
- Hier gibt es dazu auch eine **Beispiel-Lösung**: <https://git.st.archi-lab.io/students/st2/ss21/textadventure-example/textadventure-v4>

Bitte pullen Sie das/die Repos erneut, wenn Sie sie nach längerer Zeit noch einmal anschauen, der Code wird weiter gepflegt.

Übungen

- [Aufgabe 1 - Dungeon CRUD](#)
- [Aufgabe 2 - Items im Dungeon](#)
- [Aufgabe 3 - Player CRUD](#)
- [Aufgabe 4 - Monster CRUD](#)
- [Aufgabe 5 - Commands](#)

Agenda für 05.07.21

- Allgemeine Fragen zu Veranstaltung, Praktikum, Klausur
 - @RepositoryRestController - bitte ersetzen durch @RestController
 - Bitte nutzen Sie den Debugger!
 - Themenbereich 5) "Architektur großer Systeme: Strategisches DDD" entfällt aus Zeitgründen
 - wird Thema im Informatikprojekt im 5. Semester, wen es interessiert
 - Verbleibende Videos
- zum Übungsbeispiel:
 - Refactorings der Aufgabenstellung
 - kein Exposing von Field als inneres Entity mehr - ersetzt durch DungeonPosition. Dort wird jetzt die Nachbarschaft berechnet.
 - Checked => unchecked Exceptions
 - Rolle und Feinheiten des DTOMappers
 - siehe z.B. Auflösung von DungeonPosition
 - Rolle Application Service
 - siehe z.B. PlayerApplicationService - executeCommand
 - Veröffentlichung des Beispiels

Aufgabe 1 - Dungeon CRUD

Benötigte Controller-Methoden	<pre> @GetMapping("/dungeons") @GetMapping("/dungeons/{id}") @DeleteMapping("/dungeons/{id}") @PostMapping("/dungeons") </pre>
Welcher Test muss grün sein?	DungeonRESTTests1_GetPostDelete

Die vollständige Beispiellösung wird nach Abschluss der Aufgaben veröffentlicht. Hier sind vorab schon einmal Lösungen für die ersten vier Controller-Methoden.

```

@RestController
@Transactional
public class DungeonController {
    private final DungeonApplicationService dungeonApplicationService;
    private DungeonDtoMapper dungeonDtoMapper = new DungeonDtoMapper();

    @Autowired
    public DungeonController( DungeonApplicationService dungeonApplicationService ) {
        this.dungeonApplicationService = dungeonApplicationService;
    }

    /**
     * Get all
     */
    @GetMapping( "/dungeons" )
    public Iterable<DungeonDto> getAllDungeons() {
        Iterable<Dungeon> allDungeons = dungeonApplicationService.findAll();
        List<DungeonDto> allDtos = new ArrayList<>();
        for ( Dungeon dungeon : allDungeons ) {
            allDtos.add( dungeonDtoMapper.mapToDto( dungeon ) );
        }
        return allDtos;
    }

    /**
     * Get one
     */
    @GetMapping( "/dungeons/{id}" )
    public ResponseEntity<DungeonDto> getOneDungeon( @PathVariable UUID id ) {
        Dungeon dungeon = dungeonApplicationService.findById( id );
        DungeonDto createdDungeonDto = dungeonDtoMapper.mapToDto( dungeon );
        return new ResponseEntity( createdDungeonDto, OK );
    }

    /**
     * Delete
     * Deletes the dungeon and all its items. Monsters and Players live on, but are not
     * placed on any dungeon anymore.
     */
    @DeleteMapping( "/dungeons/{id}" )
    public ResponseEntity<?> deleteOneDungeon( @PathVariable UUID id ) {
        dungeonApplicationService.deleteById( id );
        return new ResponseEntity( OK );
    }

```

```

}

/**
 * Create New
 * Create a new dungeon with given size. It already contains one item of each available kind.
 */
@PostMapping("/dungeons")
public ResponseEntity<DungeonDto> createNewDungeon( @RequestBody DungeonDto dungeonDto )
    throws NoFreeFieldsException, DungeonCreationException, CoordinateException {
    Dungeon dungeon = dungeonApplicationService.createDungeonFromDto( dungeonDto );
    URI returnURI = ServletUriComponentsBuilder
        .fromCurrentRequest()
        .path("/{id}")
        .buildAndExpand( dungeon.getId() )
        .toUri();
    DungeonDto createdDungeonDto = dungeonDtoMapper.mapToDto( dungeon );
    return ResponseEntity
        .created(returnURI)
        .body( createdDungeonDto );
}

/**
 * Get field
 * @param dungeonId
 * @param fieldId - ID of a dungeon field in the shape "(1,2)"
 */
@GetMapping("/dungeons/{dungeonId}/fields/{fieldId}")
public ResponseEntity<FieldDto> getField( @PathVariable UUID dungeonId, @PathVariable String fieldId ) {
    throw new UnsupportedOperationException();
}

/**
 * Put item on field
 * @param dungeonId
 * @param fieldId - ID of a dungeon field in the shape "(1,2)"
 */
@PostMapping("/dungeons/{dungeonId}/fields/{fieldId}/item")
public ResponseEntity<?> putItemOnField(
    @PathVariable UUID dungeonId, @PathVariable String fieldId, @RequestBody Item item ) {
    throw new UnsupportedOperationException();
}

/**
 * Delete item from field
 * @param dungeonId
 * @param fieldId - ID of a dungeon field in the shape "(1,2)"
 */
@DeleteMapping("/dungeons/{dungeonId}/fields/{fieldId}/item")
public ResponseEntity<?> deleteItemFromField( @PathVariable UUID dungeonId, @PathVariable String fieldId ) {
    throw new UnsupportedOperationException();
}
}

```

Aufgabe 2 - Items im Dungeon

Benötigte Controller-Methoden	<pre>@GetMapping("/dungeons/{dungeonId}/fields/{fieldId}") @PutMapping("/dungeons/{dungeonId}/fields/{fieldId}/item") @DeleteMapping("/dungeons/{dungeonId}/fields/{fieldId}/item")</pre>
Welcher Test muss grün sein?	DungeonRESTTests2_FieldAndItems

Aufgabe 3 - Player CRUD

Benötigte Controller-Methoden	<pre>@GetMapping("/players") @GetMapping("/players/{id}") @DeleteMapping("/players/{id}") @PostMapping("/players") @PatchMapping("/players/{id}")</pre>
Welcher Test muss grün sein?	PlayerRESTTests3_GetPostDeletePatch

Aufgabe 4 - Monster CRUD

Benötigte Controller-Methoden	<pre>@GetMapping("/monsters") @GetMapping("/monsters/{id}") @DeleteMapping("/monsters/{id}") @PostMapping("/monsters") @PatchMapping("/monsters/{id}")</pre>
Welcher Test muss grün sein?	MonsterRESTTests4_GetPostDeletePatch

Aufgabe 5 - Commands

Benötigte Controller-Methoden	<pre>@PostMapping("/players/{id}/commands")</pre>
Welcher Test muss grün sein?	CommandRESTTests5

