Sprint 4.MSD: RESTful Service Model Implementation

S4 ● 2020

Overview

REST Architectural Components Implementation:

- **DTO Model Implementation with @Xml and @Json declarative serialization mapping**
- **RESTful Services Implementations with @RestController annotation**

Course Coverage:

- **Theme_3: Course_6, Course_7, Course_8**

DTO Model Implementation

DTO Classes

- Entities with serialization annotation for
  - XML:
    - @XmlRootElement
    - @XmlElement
    - @XmlElementWrapper
  - JSON (optional)
    - @JsonRootName
    - @JsonIgnoreProperties
    - @JsonProperty

DTO Aggregates

- Root Entity relationships. e.g.: @OneToMany
  - XML: @XmlElementWrapper

- Component Entities relationships: @OneToMany to avoid cycling:
  - XML: @XmlTransient

RESTful Services' Model Implementation

Spring RESTful Resource Services

- REST implementation classes to expose entities from *Repository* operations.

- REST implementation classes will be annotated with

  - @RestController

  - @RequestMapping

- *Repository* will be @Autowired into RestControllers

(Bonus) Spring REST-RPC Services

- REST implementation classes to expose Business Workflow Service Operations

- Business Workflow Services will be @Autowired into RestControllers

Deliverables

SpringBoot Projects

@SpringBootApplication main class

RestController definition services classes annotated with

- **@RestController**
- **@RequestMapping**

@Entity classes annotated with @Xml and @Json for DTO serialization/deserialization

SpringBoot configuration file:

- resource/application.properties

JUnit Test Classes

- REST agnostic CRUD tests with Apache HTTP Client library.