

Expeditors – Java Academy

Capstone 02 Instructions & Requirements

Enhance the API created in Capstone 01. As a reminder, capstone 01 included the development of an application for a Music and Entertainment management system (tracks and artists). For Capstone 02, you will be refactoring your application into a REST API using Spring Boot. You should be able to successfully exercise your API via tests and via a tool like cURL, Postman, or an equivalent. For this capstone, data will still be maintained using in-memory data structures – during the third capstone, we will add support for persisting our data to a database (which will allow us to keep data in place across application restarts).

- Students will work in teams designing and building the application
 - Design work should be done as a team – each member of the team should be aligned on any design decision made
 - Implementation individually or as a group using paired or mob programming; if executed as a group, it is critical that every member of the team get an opportunity for “hands on keyboard” coding
 - Each team should have a short meeting at a fixed time every day to talk about status and any issues or blockers that team members are facing

Features:

- Continue to provide CRUD (Create/Retrieve/Update/Delete) functionality
- Allow for querying across domains (tracks and artists) – enable management of each domain model type individually as well as management of the composed type (i.e., support CRUD functionality for artists and for tracks which have an associated artist)
- Look for opportunities (if time allows) to enhance management of bi-directional relationships across the domain models used in your application (e.g., managing the many-to-many relationship between tracks and artists)
- Think about any enhancements you want to make to your data model to support those relationships
- Take advantage of annotations to configure your application components and reduce boilerplate code

Tech Stack:

- Java
- Spring/Spring Boot
- Data will be kept in memory (during our third capstone, we will leverage a database for persisting our data)

Deliverables:

- A functioning REST API built using Java and Spring/Spring Boot
- A suite of unit tests for all tiers of the application (including mocking where appropriate)
- “Slice” (or integration) tests for your controllers using testing facilities provided by Spring Boot
- A presentation on the completed application