JOHN ABBOTT
CEGEP/COLLEGE

Spring Boot with MySQL Database

Elie Mambou, Ph.D.

(420-JD5-AB) Programming III

Summer 2022

Spring Boot annotations

- Spring uses annotation-based programming
  - Simplifies coding requirements
  - Classes do not have to extend specific base classes or implement specific interfaces
- @SpringBootApplication
  - Enables Spring Boot auto configuration and component scanning
- @Controller
  - Marks the class as a web controller, capable of handling requests
- @Autowired
  - Marks a constructor, field, or setter method to be autowired by Spring dependency injection.
  - https://www.baeldung.com/spring-autowire
- @RequestMapping("/add")
  - Maps HTTP request at a particular URL onto a particular handler method or onto an entire class
  - @GetMapping
    - Shortcut for @RequestMapping(method=RequestMethod.GET)
  - @PostMapping
    - Shortcut for @RequestMapping(method=RequestMethod.POST)
- @RequestParam
  - Extract parameters for the handler from the HTTP request
  - https://www.baeldung.com/spring-request-param
- @ResponseBody
  - https://www.baeldung.com/spring-request-response-body

Hello World – Handler

- **Two steps to create the handler:**

- @RestController
  - Put annotation above "class" keyword.
  - This lets Spring know that this class should be managed by the Spring IoC container.

- Create handler method in that class

```
@GetMapping("/hello")
    public String sayHello(@RequestParam(value = "myName", defaultValue = "World") String name) {
            return String.format("Hello %s!", name); }
```

  - This handler will listen for a HTTP GET event for the URL "localhost:XXXX/hello"
  - When that event is triggered (i.e., by the user), this handler method (sayHello) executes.
  - Using @RequestParam, we can use the "myName" information sent in the GET request as the value for the name parameter of the handler.
    - If no myName info was provided in the GET request, then use the default value "World" for the name parameter.
  - The method returns a text string to display on the "/hello" page.

@RestController vs @Controller

JOHN ABBOTT CEGEP/COLLEGE

•We've now seen both @RestController and @Controller. What's the difference?

•Basically, @RestController is the same as using @Controller and then putting @ResponseBody before every handler in the class.

- i.e., just a short-cut convenience.

•In our CRUD, we use @Controller because sometimes we won't be returning information to display at the URL, but instead will be returning the view (i.e., actual html page) to display.

- If we use @ResponseBody, we can't return the html page to use.
- So, some handlers will return information and others html pages.

•https://www.baeldung.com/spring-controller-vs-restcontroller

Thymeleaf

JOHN ABBOTT CEGEP/COLLEGE

•Thymeleaf is a Java library.

•An XML/XHTML/HTML5 template engine able to apply a set of transformations to template files in order to display data and/or text produced by your applications.

•Main goal: provide an elegant and well-formed way of creating templates.

•Based on XML tags and attributes that define the execution of predefined logic, instead of explicitly writing that logic as code inside the template.

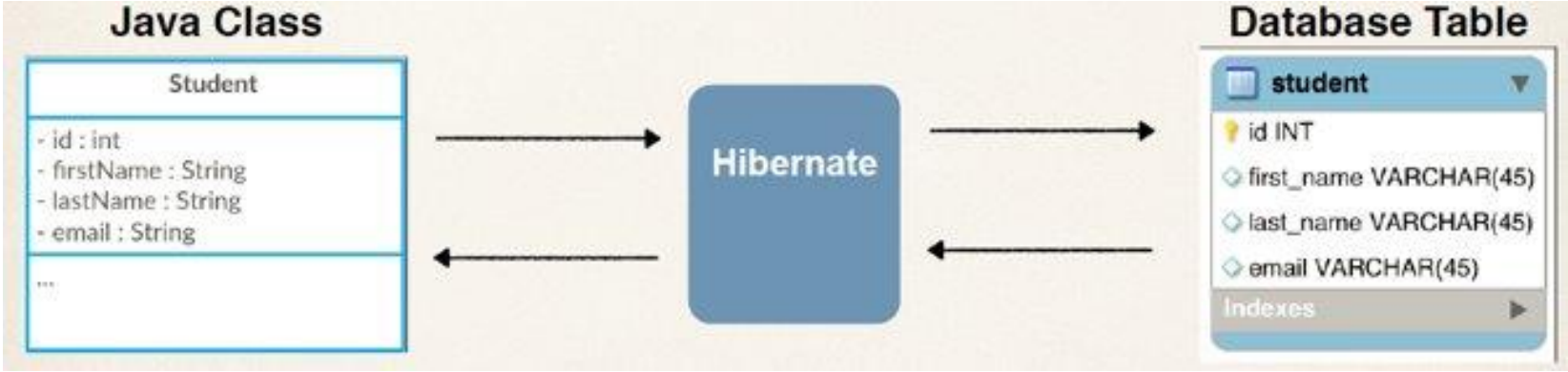
•https://www.baeldung.com/thymeleaf-in-spring-mvc

JPA (Java Persistence API)

•Java Persistence API provides a specification for persisting, reading, and managing data from your Java object to relational tables in the database.

•Object/Relational Mapping (ORM)

- Map application objects to relational database
- Allow application to focus on objects



•Hibernate is an implementation of JPA

•Spring Data JPA is not an implementation, it's just an abstraction used to significantly reduce the amount of boilerplate code

•JPA is a separate ORM container that instantiates and manages the lifecycles of entities

•Separate from Spring's IoC Container

•Does Dependency Injection based on @Entity relationships only.

Setup Data Object

JOHN ABBOTT CEGEP/COLLEGE

- Use annotations to inform the Hibernate framework

- @Entity
  - Annotation that we use on a "POJO" Plain Old Java Object that we have created
  - Simplest form of POJO is a Java Bean.

- @Id
  - Annotation to specify the field value that will be used to unique distinguish data being stored in the database

- @GeneratedValue
  - Annotation that lets the framework handle the creation of unique Id values for our POJO objects that we are storing
  - More approaches: https://www.baeldung.com/hibernate-identifiers

Resources/Application.properties

```
spring.mvc.view.prefix=/view/

spring.mvc.view.suffix=.jsp



spring.datasource.url=
jdbc:mysql://localhost:3306/springdatabase?autoReconnect=true&useSSL=false

spring.datasource.username=root

spring.datasource.password=eliemambou



spring.jpa.hibernate.ddl-auto=update
```

Potential topics for projects

JOHN ABBOTT CEGEP/COLLEGE

- Will be covered:
  - MySql
    - https://www.springboottutorial.com/spring-boot-with-mysql-and-oracle
    - https://springbootdev.com/2017/11/13/spring-boot-and-mysql-simple-crud-application-rest-apis-with-spring-data-jpa/
  - Hibernate – More complex data relationships (Composition, Joins, etc.)
    - @ManyToMany https://www.toptal.com/spring/beginners-guide-to-mvc-with-spring-framework
    - @OneToMany https://www.baeldung.com/hibernate-one-to-many
    - @OneToOne
    - @ManyToOne
    - https://docs.jboss.org/hibernate/orm/3.3/reference/en/html/associations.html
- More GUI Elements for I/O with Thymeleaf
  - https://www.baeldung.com/spring-thymeleaf-request-parameters
  - https://medium.com/@grokwich/spring-boot-thymeleaf-html-form-handling-762ef0d51327
- May explore based on project needs:

- Form Validation / Error Messages (Advanced)
  - https://howtodoinjava.com/spring-mvc/spring-mvc-display-validate-and-submit-form-example/
  - https://www.baeldung.com/spring-thymeleaf-error-messages
  - https://medium.com/@grokwich/spring-boot-thymeleaf-html-form-handling-part-2-b4c9e83a189c
- Misc links (Advanced)
- https://www.baeldung.com/spring-boot-start
- https://howtodoinjava.com/spring-mvc/spring-mvc-requestmapping-annotation-examples/

Misc Links

- https://spring.io/guides/gs/serving-web-content/
- https://www.youtube.com/watch?v=YywLS8XdxLQ
- https://www.youtube.com/watch?v=dVTh--j1suI
- https://www.codejava.net/frameworks/spring/spring-mvc-spring-data-jpa-hibernate-crud-example
- https://www.baeldung.com/spring-boot-h2-database
- https://www.springboottutorial.com/spring-boot-with-mysql-and-oracle

Q & A