2021_25_IV_Rest API_CA2_G2_Slot 1

Test Summary

- No. of Sections: 2
- No. of Questions: 10
- Total Duration: 90 min

Section 1 - MCQ

Section Summary

- No. of Questions: 9
- Duration: 15 min

Additional Instructions:

None

Q1. Which HTTP methods are typically used in RESTful API for performing data operations?

- POST, GET, DELETE
- PUT, PATCH, DELETE
- POST, GET, PUT, PATCH, DELETE
- GET, PUT, DELETE

Q2. In a RESTful API architecture, which layer is responsible for handling HTTP requests, routing, and returning HTTP responses?

- Controller
- Service layer
- View
- Model

Q3. What is the main advantage of using a Service layer in a software application?

- Separation of concerns
- Simplified user interface
- Faster performance
- Reduced database load

Q4. What is the main purpose of the @JsonProperty annotation in Jackson library?

To specify the name of a JSON property during serialization and deserialization

To define a new JSON property in a JSON object

To specify the order of JSON properties in a JSON object

To indicate that a property should be ignored during serialization and deserialization

Q5. Which of the following statements about the @Repository annotation in Spring is NOT true?

The @Repository annotation is used to mark a class as a data access object (DAO) or repository in Spring.

The @Repository annotation is a specialization of the @Component annotation.

The @Repository annotation is used for transactional management in Spring.

The @Repository annotation is used to handle database-related exceptions in Spring.

Q6. Which of the following is true about the @RequestParam annotation in Spring?

The @RequestParam annotation is required to be used in every controller method that expects query parameters in the URL.

The @RequestParam annotation is optional and can be used selectively based on the specific requirements of the controller method.

The @RequestParam annotation is only used with GET HTTP method and cannot be used with other HTTP methods.

The @RequestParam annotation can only bind primitive data types as method parameters.

Q7. In JPA, how can you use the AND operator to combine multiple conditions in a query?

Using the & symbol

Using the AND keyword

Using the && symbol

Using the AND() method

Q8. What is the default behavior of methods in a class annotated with @RestController in terms of response type?

The return type is automatically serialized to JSON.

The return type is automatically serialized to XML.

The return type is automatically serialized to HTML.

The return type is not automatically serialized.

Q9. What is the typical request body for a DELETE API?

Empty or null

JSON or XML representation of the resource to be deleted

Query parameters with the resource ID to be deleted

Form data with the resource ID to be deleted

Section 2 - Coding

Section Summary

- No. of Questions: 1
- Duration: 75 min

Additional Instructions:

None

Q1. **Property Management - eMall**

Overview:
Develop a property management system for a shopping mall using CRUD operation

Functional Requirements:

Create 4 folders inside the **WORKSPACE/springapp/src/main/java/com/example/springapp**
1. Controller
2. Model
3. Repository
4. Service

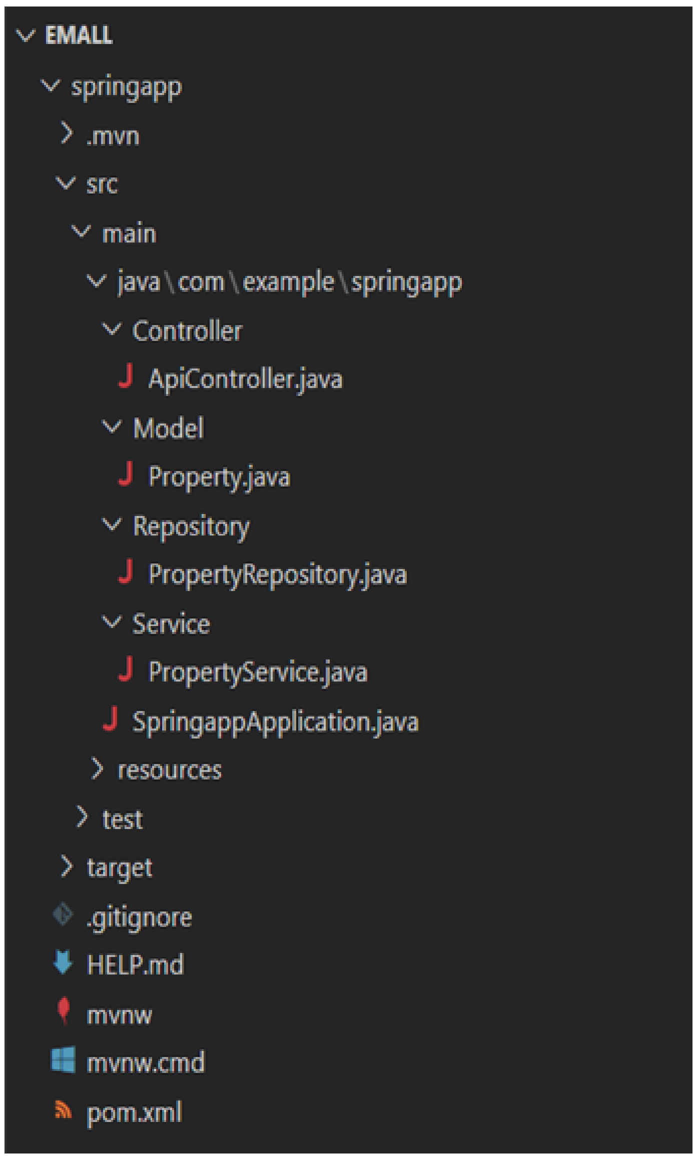
Inside the controller, create a Java file named “ApiController.java”
Inside the model, create a Java file named “Property.java”

Create 6 variables

1. id - int
2. name - string
3. address - string
4. description - string
5. price - int
6. availability - string

as well as create getters and setters and constructors for the corresponding variables.

- Inside the Repository, create a Java file named "PropertyRepository.java"
- Inside the service, create a Java file named "PropertyService.java"
The project structure looks like this image



```
EMALL
  springapp
    .mvn
    src
      main
        java\com\example\springapp
          Controller
            ApiController.java
          Model
            Property.java
          Repository
            PropertyRepository.java
          Service
            PropertyService.java
          SpringappApplication.java
        resources
      test
    target
    .gitignore
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
```

Core Platform

OpenJDK 11

API:

Question 1 (16 Marks)

POST - "/" --> true/false (**8 Marks**)
GET - "/{id}" --> Property object (**4 Marks)**
GET - "/" --> List of Property object **(4 Marks)**

Question 2 (16 Marks)
PUT - “/{id}” --> Property object (**8 Marks**)
DELETE - “/{id}” --> true/false (**8 Marks**)

Note:

Copy and paste it into the **application.properties** file

spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost/emall?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=examly
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.show-sql= true
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect

API endpoint:
8080

Platform Guidelines:
To run the command use Terminal in the platform.

Spring Boot:
Navigate to the springapp directory => **cd springapp**
To start/run the application '**mvn spring-boot:run**'
Click on the Run Test Case button to pass all the test cases

Answer Key & Solution

Section 1 - MCQ

Q1

POST, GET, PUT, PATCH, DELETE

Solution

No Solution

Q2

Controller

Solution

No Solution

Q3

Separation of concerns

Solution

No Solution

Q4

To specify the name of a JSON property during serialization and deserialization

Solution

No Solution

Q5

The @Repository annotation is used for transactional management in Spring.

Solution

No Solution

Q6

The @RequestParam annotation is optional and can be used selectively based on the specific requirements of the controller method.

Solution

No Solution

Q7

Using the AND keyword

Solution

No Solution

Q8

The return type is automatically serialized to JSON.

Solution

No Solution

Q9 Empty or null

Solution

No Solution

Section 2 - Coding

Q1 Solution cannot be displayed