📘 Spring Boot – Day 4 Interview Documentation

1️⃣ Why is a Service Layer is Required?

Definition

The **Service layer** contains the **business logic** of the application and acts as a **bridge between Controller and Repository**.

Why We Need It

- Maintains **separation of concerns**
- Keeps controllers **lightweight**
- Central place for **business rules**
- Improves **code reusability**
- Easier **unit testing**
- Supports future scalability

Flow

Controller → Service → Repository → Database

Interview Answer (Short)

> “Service layer is required to separate business logic from controller logic and to make the application maintainable, testable, and scalable.”

2️⃣ What is Constructor Injection?

Definition

Constructor Injection is a type of **Dependency Injection** where dependencies are provided through the **constructor**.

Example

```
@Service
public class StudentService {

    private final StudentRepository studentRepository;

    public StudentService(StudentRepository studentRepository) {
        this.studentRepository = studentRepository;
    }
}
```

Why Constructor Injection is Preferred

- Makes dependencies **mandatory**
- Supports **immutability**
- Better for **unit testing**
- Avoids `NullPointerException`
- Recommended by Spring

Interview Answer

> "Constructor injection injects dependencies through the constructor and is preferred because it ensures immutability and better testability."

3️⃣ Difference Between `@Controller` and `@RestController`

`@Controller`

- Used in **Spring MVC**
- Returns **view names** (HTML, JSP, Thymeleaf)
- Requires `@ResponseBody` to return JSON

`@RestController`

- Used to build **REST APIs**
- Returns **JSON/XML directly**
- No view resolver needed
- Automatically adds `@ResponseBody`

Key Point

@RestController = @Controller + @ResponseBody

Interview Answer (One Line)

> "@Controller returns views, whereas @RestController returns JSON data for REST APIs."

4️⃣ How `findAll()` Works Internally?

Source

`findAll()` comes from **JpaRepository**

Internal Working Steps

1. Controller calls service method
2. Service calls `repository.findAll()`
3. Spring Data JPA creates a **proxy implementation**
4. Hibernate generates **SQL query**
5. Query executes on database
6. ResultSet is mapped to **Entity objects**
7. List of entities is returned

SQL Generated (Example)

SELECT * FROM student;

Interview Answer

> "findAll() is provided by JpaRepository, internally Hibernate generates SQL, fetches data from database, and maps it to entity objects."

5️⃣ How Java Object Converts to JSON?

Component Used

Jackson Library (default in Spring Boot)

Conversion Process

1. Controller returns Java object
2. Spring uses `HttpMessageConverter`
3. Jackson converts object → JSON
4. JSON sent as HTTP response

Example

Java Object

```
Student {
  id: 1,
  name: "Rahul"
}
```

JSON Output

```
{
  "id": 1,
  "name": "Rahul"
}
```

Interview Answer

> “Spring Boot uses Jackson library to automatically convert Java objects into JSON using HttpMessageConverters.”