Chapter 4 Getting Started with Spring Boot

Lecture Duration: 90 Minutes

🕒 *Goal:* Build conceptual understanding and prepare for coding workshop.

1 Understanding Spring Boot

What is Spring Boot?

- A **framework** simplifying Java backend development.
- Built on top of the **Spring Framework**.
- Provides:
  - **Auto-configuration**
  - **Embedded servers** (Tomcat, Jetty)
  - **Dependency management**

🧠 *Concept*: "Convention over configuration" — Spring Boot automatically sets up what you need.

Why Spring Boot?

- Reduces setup time
- Encourages **microservice architecture**
- Includes production tools:
  - Metrics, Health Checks, Security
- Excellent for REST API & backend system design

2 Creating a Project

Spring Initializr: Quick Start

URL: https://start.spring.io

Steps

1. Choose **Project:** Maven
2. **Language:** Java
3. Add dependencies:
   - `Spring Web`
   - `Spring Boot DevTools`
   - (Optional) `Spring Data JPA`, `MySQL Driver`
4. Generate → Import into IDE (IntelliJ / Eclipse / VS Code)

Project Structure Overview

```
demo/
├── src/
│   ├── main/java/com/example/demo/
│   │       ├── controller/
│   │       ├── service/
│   │       └── repository/
│   └── resources/
│           ├── static/
│           ├── templates/
│           └── application.properties
└── pom.xml
```

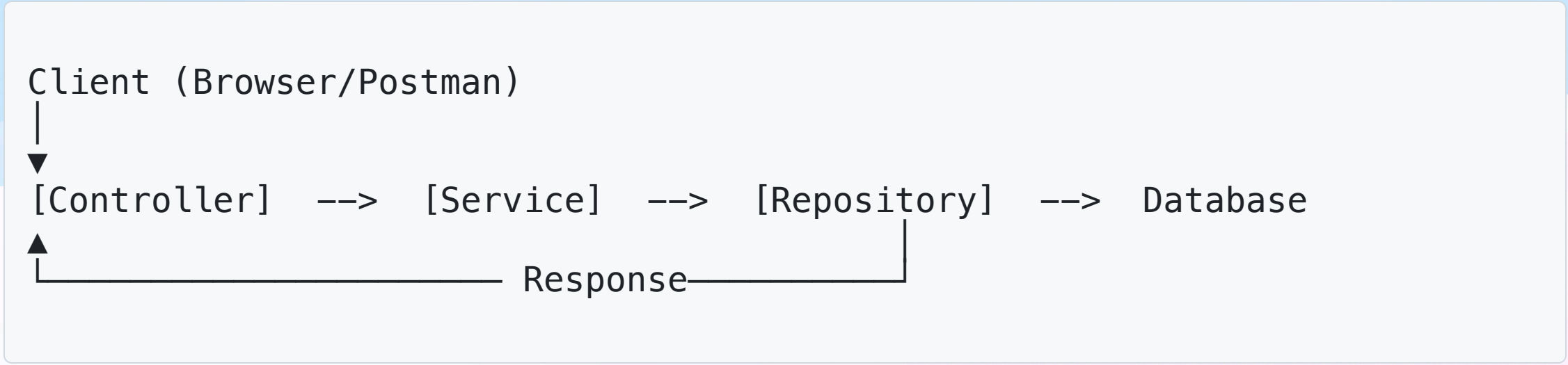
Observation: Separation of concerns → Clean architecture.

3 Backend Architecture

Spring Boot Layered Design

🧠 *Concept:* Each layer communicates only with the one directly below it.

Request–Response Flow Diagram



💡 *Key idea:* Controller = entry point, Service = logic, Repository = data access.

4 Core Components in Detail

Controller Layer

```java
@RestController
public class HelloController {

    @GetMapping("/hello")
    public String hello() {
        return "Hello, Spring Boot!";
    }
}
```

Notes:

- Uses `@RestController` annotation.
- Handles HTTP methods like `GET`, `POST`.
- Returns data directly (no view templates).

Service Layer

```
@Service
public class HelloService {
    public String getMessage() {
        return "Hello from Service Layer!";
    }
}
```

- Contains **business logic**.
- Promotes **reusability** and **testing**.
- Injected into Controller using `@Autowired`.

Repository Layer

```
@Repository
public interface UserRepository extends JpaRepository<User, Long> { }
```

- Defines CRUD operations for database entities.
- Spring automatically generates implementation.

Application Entry Point

```java
@SpringBootApplication
public class DemoApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

- Starts embedded Tomcat server.
- Scans packages for annotated components.

5 Demo: Hello World API

application.properties

```
server.port=8080
spring.application.name=demo
```

Run the Application

Steps

1. Run `DemoApplication`.
2. Visit → http://localhost:8080/hello
3. Output → `Hello, Spring Boot!`

🎯 *Learning outcome:* You have deployed your first API endpoint.

Example Workflow Visualization

```
User → Browser → http://localhost:8080/hello
 ↓
Spring Boot → DispatcherServlet
 ↓
HelloController → HelloService
 ↓
Return Response: "Hello, Spring Boot!"
```

Debugging Common Issues

Problem	Possible Cause	Solution
Port in use	Another process on 8080	Change in `application.properties`
404 Not Found	Wrong endpoint path	Check `@GetMapping`
Dependency Error	Missing library	Update `pom.xml`

6 Summary & Workshop Preview

Key Takeaways

- Spring Boot simplifies Java backend creation.
- You learned:
  - Project structure
  - Controller–Service–Repository pattern
  - Request–response flow
  - API deployment
- You are now ready to code in the **next workshop**.

Next Workshop

“Building a CRUD API with Spring Boot and MySQL”

- Implement RESTful CRUD endpoints.
- Connect to MySQL Database.
- Use `@Entity`, `@Repository`, `@Service`.
- Test with **Postman**.

🧩 *Preparation Tip:* Ensure your IDE and JDK are installed before the session.

Thank You!

Questions?

💬 *Next:* Hands-on Workshop

📧 Instructor: Asst. Prof. Chanankorn Jandaeng

🏫 Walailak University — IIT67-272