

Spring Boot Notes

♦ 1. What is Spring Boot?

- **Spring Boot** is a framework to create **standalone, production-grade** Spring-based applications easily.
 - No need for complex XML configuration.
 - Comes with embedded servers like **Tomcat, Jetty**, etc.
-

♦ 2. Key Features

- **Auto Configuration**
 - **Standalone** Spring applications
 - **Embedded Web Servers**
 - **Production Ready** (Actuator)
 - **Minimal Configuration**
 - **Spring Initializr** support
-

♦ 3. Spring Boot Starter Project

Use **Spring Initializr** (<https://start.spring.io/>) to create a project.

Add starter dependencies:

xml

CopyEdit

```
<dependency>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

◆ 4. Project Structure

```
css
CopyEdit
src/
├─ main/
│   ├── java/
│   │   └─ com.example.demo/
│   │       └─ DemoApplication.java
│   └─ resources/
│       └─ application.properties
```

◆ 5. Main Application Class

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

◆ 6. REST Controller

```
java
CopyEdit
@RestController
public class HelloController {
    @GetMapping("/hello")
    public String hello() {
        return "Hello, Spring Boot!";
    }
}
```

♦ 7. application.properties

properties
CopyEdit
server.port=8081
spring.application.name=DemoApp

♦ 8. CRUD using Spring Boot and JPA

java
CopyEdit
@Entity
public class User {
 @Id
 @GeneratedValue
 private Long id;
 private String name;
}

java
CopyEdit
public interface UserRepository extends JpaRepository<User, Long> {}

♦ 9. Dependency Injection

Spring Boot uses annotations like:

- @Autowired – for dependency injection
 - @Service, @Repository – for component scanning
-

♦ 10. Spring Boot Annotations

- `@SpringBootApplication`
 - `@RestController`
 - `@GetMapping`, `@PostMapping`, etc.
 - `@RequestBody`, `@PathVariable`
 - `@Entity`, `@Id`, `@GeneratedValue`
 - `@Autowired`
-

◆ 11. Spring Boot Actuator

Used for monitoring:

xml

CopyEdit

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-actuator</artifactId>  
</dependency>
```

Access endpoints like `/actuator/health`, `/actuator/info`

◆ 12. Connecting to Database (MySQL)

properties

CopyEdit

```
spring.datasource.url=jdbc:mysql://localhost:3306/mydb  
spring.datasource.username=root  
spring.datasource.password=admin  
spring.jpa.hibernate.ddl-auto=update
```

♦ 13. Running the App

bash

CopyEdit

```
mvn spring-boot:run
```

♦ 14. Testing with Spring Boot

java

CopyEdit

```
@SpringBootTest
class DemoTests {
    @Test
    void contextLoads() {}
}
```

♦ 15. Advanced Features

- Spring Security for authentication/authorization
- Spring Boot with Swagger for API docs
- Spring Boot with Kafka or RabbitMQ for messaging
- Spring Boot Microservices with Eureka, Zuul, Config Server