Parul® University

NAAC A++

Enterprise Programming using JAVA

Chapter-6: Spring Boot

Prof. ARNIKA PATEL

Assistant Professor

Department of CSE

Content

INDEX

Spring Boot-AOP

AOP(Aspect Oriented Programming) breaks the full program into different smaller units. In numerous situations, we need to log, and audit the details as well as need to pay importance to declarative transactions, security, caching, etc., Let us see the key terminologies of AOP

Aspect: It has a set of APIs for cross-cutting requirements. Logging module is an example of the AOP aspect of logging.
Joint Point: AOP aspect plug in place
Advice: Via this, the actual implementation of code is taken care for the AOP approach. It can be either before/after/after returning/after throwing. In this article let us see the examples related to this.

Spring Boot-AOP

Pointcut: Set of one or more join points where an advice need to be executed.

Introduction: This is the place where we can add new methods or attributes to the existing classes.

Target Object: One or more aspects will be there to provide advice for the target object.

Weaving: Linking aspects with other application types or objects. It can be done at compile-time/runtime/loadtime.

Spring Boot-AOP

In web applications, each of the layers(web, business, and data layer) are responsible for different tasks and they perform these tasks individually. But there are a few **common aspects** that are applied to each layer such as security, caching, validation, etc. These common aspects are known as **Cross-Cutting Concerns**.

Spring Boot- Database

In modern application development, integrating a database is crucial for persisting and managing data.

Spring Boot simplifies this process by providing seamless integration with various databases through JPA (Java Persistence API) and Hibernate ORM (Object-Relational Mapping).

Spring Boot- Database

Database Integrations (MySQL and H2)

MySQL: A widely used relational database management system known for its reliability, performance, and ease of use. It is ideal for production environments where data needs to be persistently stored.

H2 Database: An in-memory database that is extremely fast and useful for development and testing. As it is in-memory, data is lost once the application is stopped. H2 provides excellent support and allows you to switch between databases with minimal configuration changes.

Spring Boot- Database

Steps to Integrate MySQL and H2 in a Spring Boot Application

Step 1: MySQL Integration

1. Add the Dependencies: Include spring-boot-starter-data-jpa and mysql-connector-java in your pom.xml.

2. Configure MySQL Connection: Update application.properties with MySQL database configuration.

3. Database Initialization (Optional): Use schema.sql and data.sql files to initialize the database schema and data.

Spring Boot- Database

Steps to Integrate MySQL and H2 in a Spring Boot Application

Step 2: H2 Database Integration

1. Add the Dependencies: Include spring-boot-starter-data-jpa and h2 in your pom.xml.

2. Configure H2 Connection: Update application-dev.properties with H2 database configuration. Enable the H2 console for easy access to the in-memory database via the web interface.

3. Auto Schema Generation: H2 will automatically generate the schema based on JPA entities.

Spring Boot- Database

Switching Between Databases

Spring Boot makes it easy to switch between MySQL and H2 or other databases by changing the configuration in the application.properties file. This flexibility allows developers to use H2 for local development and MySQL in production without altering the codebase.

Spring Boot- REST

Representational State Transfer (REST) is a software architectural style that defines a set of constraints for creating web services. **RESTful web services** allow systems to access and manipulate web resources through a uniform and predefined set of stateless operations.

Unlike **SOAP**, which exposes its own set of operations, RESTful web services rely on simple **HTTP methods** such as GET, POST, PUT, and DELETE.

Spring Boot- REST

Why Spring Boot?

Spring Boot is built on top of Spring Framework, simplifying project setup and configuration. It provides default configurations to avoid boilerplate code and is ideal for beginners looking to work with Spring.

Spring Boot- REST

Steps to Create a REST API

Step 1: Define the Employee Entity

Step 2: Create a Storage Class

Step 3: Create the DAO Class

Step 4: Create the Controller

Step 5: Run the Application

PPT Content Resources Reference Sample:

1. **Book Reference**

   Jim Farley, William Crawford, David Flanagan. Java Enterprise in a Nutshell, O'Reilly

2. **Book Reference**

   Rocha, R., Purificação, J. (2018). Java EE 8 Design Patterns and Best Practices: Build Enterprise-ready Scalable Applications with Architectural Design Patterns. Germany: Packt Publishing..

3. **Website Reference**

   https://www.scribd.com/document/268349254/Java-8-Programming-Black-Book .

4. **Sources**

   https://developers.redhat.com/topics/enterprise-java

5. **Article**

   https://www.researchgate.net/publication/276412369_Advanced_Java_Programming

Parul®
University

NAAC
GRADE A++

https://paruluniversity.ac.in/