**WEEK-4: Spring REST using Spring Boot 3(Module 7)**

**Introduction to Spring REST and Spring Boot 3**

Spring REST and Spring Boot 3 provide a robust framework for building RESTful web services. REST (Representational State of Resource) is an architectural style that emphasizes stateless communication between client and server. Spring Boot 3 offers a simplified way to build RESTful services with features like auto-configuration, embedded servers, and production-ready metrics.

- Overview of RESTful architecture

- Introduction to Spring REST

- Benefits of using Spring Boot for RESTful services

- Setting up a Spring Boot project for REST

- What's New in Spring Boot 3?

Building a Simple REST Controller

A REST controller is the core component of a RESTful service, handling HTTP requests and responses. In Spring Boot, you can create a REST controller using annotations like @RestController and @RequestMapping.

- Creating a basic REST controller

- Defining request mappings

- Handling HTTP methods (GET, POST, PUT, DELETE)

- Returning JSON responses

**Request and Response Handling**

Effective request and response handling is crucial for a RESTful service. Spring Boot provides features like path variables, query parameters, and request body processing to handle requests. You can also customize response status and headers.

- Handling path variables and query parameters

- Request body and form data processing

- Customizing response status and headers

- Exception handling in REST controllers

**RESTful Resource Representation with DTOs and HATEOAS**

Data Transfer Objects (DTOs) help represent resources in a RESTful service. HATEOAS (Hypermedia as the Engine of Application State) enables clients to navigate through resources using hypermedia links.

- Introduction to Data Transfer Objects (DTOs)

- Mapping entities to DTOs

- Customizing JSON serialization and deserialization

- Managing versioning and backward compatibility

- Understanding HATEOAS and adding links to resources

**Security, Testing, and Documentation**

Security, testing, and documentation are essential aspects of a RESTful service. Spring Boot provides features like Spring Security, testing frameworks, and API documentation tools.

- Securing RESTful endpoints with Spring Security

- Implementing authentication and authorization

- Token-based authentication (JWT)

- Unit testing and integration testing for REST services

- Documenting RESTful APIs with Swagger/OpenAPI

By following these topics, you can build robust and scalable RESTful web services using Spring Boot 3.