

Backend Team Learning Topic Covered

Learning Duration: Start date: 27-Jan-2025 – 29-May-2025

1. Rest Api's:

Basics of REST

- HTTP Methods: GET, POST, PUT, DELETE, PATCH
- URI Design Principles (resource naming, plural nouns)
- Status Codes (200, 201, 204, 400, 401, 403, 404, 500)

Creating REST APIs using Spring Boot

- @RestController and @RequestMapping
- @GetMapping, @PostMapping, @PutMapping, @DeleteMapping
- @RequestParam, @PathVariable, @RequestBody

Response Handling

- Returning ResponseEntity<T>
- Custom response objects

Data Validation

- @Valid, @NotNull, @Size, @Email
- BindingResult for capturing validation errors

2.Spring Security:

Authentication vs Authorization

JWT Structure

- Header, Payload, Signature

Generating JWT in Spring Boot

- Use of libraries like jwt

- HS256 Signing algorithm

Securing Endpoints

- Spring Security Configuration
- Custom JwtFilter or OncePerRequestFilter
- AuthenticationManager, UserDetailsService, UsernamePasswordAuthenticationToken

Token Validation

- Token expiry, invalid signature.

3.Exception Handling:

Try-Catch Blocks (basic Java)

Spring Boot Exception Handling

- @ControllerAdvice, @ExceptionHandler
- ResponseEntityExceptionHandler

Custom Exception Classes

- ResourceNotFoundException, InvalidRequestException

Global Exception Handling

- Format error response with timestamp, status, message, path

Validation Error Handling

- MethodArgumentNotValidException
- Field-wise error response for @Valid

4.Multithreading:

Thread Basics

- Extending Thread vs implementing Runnable

Executor Framework

- ExecutorService, ThreadPoolExecutor

- `Executors.newFixedThreadPool()`, `submit()`, `shutdown()`

Synchronization

- `synchronized` blocks/methods
- `volatile`, `ReentrantLock`

Callable & Future

- Returning results from threads

5.Spring Boot :

MVC Structure

Spring Boot Basics

- Auto-configuration, `@SpringBootApplication`, `application.properties`

Project Structure

- Controller, Service, Repository, Model

Spring Data JPA

- `@Entity`, `@Id`, `@GeneratedValue`
- `JpaRepository`, `CrudRepository`

Database Configuration

- MySQL/PostgreSQLIntegration
- Schema auto-creation: `ddl-auto`

Dependency Injection

- `@Autowired`, constructor-based injection

Bean Lifecycle

- `@Component`, `@Service`, `@Repository`, `@Configuration`

6.WebSocket:

- Difference from HTTP (bi-directional)
- Use cases: Chat apps, live updates, games

Spring Boot WebSocket Support

- Add spring-boot-starter-websocket
- Use @EnableWebSocketMessageBroker

STOMP Protocol

- Message format and routing
- Client uses SockJS + STOMP

Server-side Setup

- @MessageMapping, @SendTo, SimpMessagingTemplate

Client-side Setup

- JS: SockJS + StompJS

7.HTML

HTML Boilerplate (<!DOCTYPE html>, <html>, <head>, <body>)

Semantic Tags: <header>, <nav>, <main>, <footer>, <section>, <article>

Forms: <form>, <input>, <select>, <textarea>, form attributes (action, method)

Multimedia: , <audio>, <video>

Tables: <table>, <thead>, <tbody>, <tr>, <td>, <th>

Lists: , ,

Links and Navigation: , internal vs external links