KOENIG
step forward

Build Reactive MicroServices using Spring WebFlux/SpringBoot

Table of Content

- We will explore lot of different operators in Flux and Mono.

Module 6: Build the first Non-Blocking RESTFUL API using Annotated Controllers - Hands On

- In this section we will build the first non-blocking API using the annotated controllers.
- This section covers the fundamentals of how the reactive API works.
- This section also covers the coding aspect of how to return a Flux/Mono from an endpoint.
- This section also covers how to write JUnit test cases using WebTestClient.

Module 7: Build Non-Blocking RESTFUL API using Functional Web - Hands On

- In this section we will build the non-blocking API using the Functional Web Module.
- This section explains about the RouterFunction and HandlerFunction which forms the foundation for Function Web Module.
- This section also covers how to write JUnit test cases using WebTestClient.

Module 8: Spring WebFlux & Netty - Execution Model

- This section explains about the different layers behind WebFlux to serve an HTTP Request/Response.
- This section covers the concepts of NETTY such as Channel, EventLoop , and some of the technical aspects of Netty.

Module 9: Overview of the Reactive API

- This section will give you an Overview of the Reactive API that we are going to build as part of this course.

Module 10: Reactive Programming in Databases - MongoDB - Hands On

- In this section we will learn about how to write reactive programming code with MongoDB.
- Define the Item Document for the project.

- This section covers about how to configure different profiles in Spring Boot.
- In this section we will set up the ItemReactive Mongo DB adapter.
- This section also covers how to write JUnit test cases for the reactive repository.

Module 11: Build the Item Reactive API Endpoint - Using RestController

- In this section we will learn about how to code the Item CRUD Reactive API using the @RestController approach.
- This section also covers how to write automated tests using JUNIT and the non-blocking test client WebTestClient.

Module 12: Build the Item Reactive API Endpoint - Using Functional Web

- In this section we will learn about how to code the Item CRUD Reactive API using the Functional Web approach.
- This section also covers how to write automated tests using JUNIT and the non-blocking test client WebTestClient.

Module 13: Build Non-Blocking Client using WebClient

- In this section we will explore the techniques to interact with Reactive API using the WebClient.
- Learn the techniques to Invoke the Reactive API
  using exchange() and retrieve() methods.
- We will explore the GET, PUT, POST and DELETE operations using the WebClient.

Module 14: Handling Exceptions in WebFlux - RestController

- In this section we will code and explore different approaches to handle the exceptions/errors that occurs in the reactive api that's built using RestController.
- Handle exceptions using @ExceptionHandler and @ControllerAdvice.
- This section also covers how to write JUNIT test cases for the Exception scenarios.

Module 15: Handling Exceptions in WebFlux - Functional Web

- In this section we will code and explore different approaches to handle the exceptions/errors that occurs in the reactive api that's built using Functional Web.
- Handle exceptions using WebExceptionHandler.
- This section also covers how to write JUNIT test cases for the Exception scenarios.

Module 16: WebClient - Exception Handling

- In this section we will code and explore how to handle the exceptions using the WebClient.
- Learn the techniques to handle the exceptions using exchange() and retrieve() methods.

Module 17: Streaming Real-Time Data using WebFlux - Server-Side Events (SSE)

- In this section we will code and learn about build an endpoint for Streaming RealTime Data using Mongo DB and Spring WebFlux.
- This section covers about the Tailable Cursors and Capped Collections in Mongo DB.
- Build a Non-Blocking Streaming Endpoint and interact with the Mongo DB using the @Tailable annotation.
- Learn to write Automated Tests using JUNIT for the Streaming Endpoints (SSE).