

Introduction to Reactive Spring - TT3353

Explore Reactive Programming Basics, ReactiveX Essentials, Project Reactor, WebFlux, Spring Data / Databases & More

Duration: 3 Days

Skill Level: Intermediate

Available Format: Instructor-Led Online; Instructor-Led, Onsite In Person ; Blended; On Public Schedule

Introduction to Reactive Spring is a comprehensive Spring training workshop geared for experienced Java and Spring developers who wish to explore concurrent, asynchronous and reactive programming APIs and techniques using Spring. After an introduction to reactive programming, Reactive Streams and the Project Reactor APIs, this course will show how this APIs are integrated into Spring. Spring 5 includes Spring WebFlux, providing a reactive programming model for web applications, including support for Reactive REST APIs.

What You'll Learn

Overview

Spring Data reactive allow us to implement database operations relying on Reactive Programming APIs. The Spring R2DBC initiative gave us a reactive programming API to access our relational databases, allowing for a fully reactive development stack. From using R2DBC to access a relational database at the back-end to WebFlux for the development of a fully reactive presentation tier.

Introduction to Reactive Spring is a comprehensive hands-on intermediate and beyond-level Java training workshop geared for experienced developers who wish to explore concurrent, asynchronous and reactive programming APIs and techniques using Spring. After an introduction to reactive programming, Reactive Streams and the Project Reactor APIs, this course will show how this APIs are integrated into Spring. Spring 5 includes Spring WebFlux, providing a reactive programming model for web applications, including support for Reactive REST APIs. Spring WebSocket assists in the creation of

web applications which provide a full-duplex, two-way communication between client and server.

NOTE: We offer an alternative course for those who wish to use ReactiveSpring and MongoDB: Introduction to Reactive Spring with MongoDB (4 days) (TT3354). **Please inquire for details.**

Objectives

Reactive Spring focuses on providing an understanding of the fundamental principles and technologies that are used in reactive programming. This understanding is critical to being able to diagnose, troubleshoot, tune, and perform other lifecycle activities.

Working within in an engaging, hands-on learning environment, guided by our expert team, attendees will:

- Understand the ReactiveX specification
- Understand the basics of Reactive Programming
- Discuss the advantages and limitations of Observables
- Write a client application capable of handling Reactive events
- Apply operators to event streams to filter, modify and combine the objects emitted by event publishers
- Select the appropriate type of Event Source
- Use both Cold and Hot Observables
- Deal with backpressure problems in reactive programming
- Develop a reactive REST application using Spring WebFlux
- Define router functions using WebFlux.fn
- Use the WebClient API to handle non-blocking and reactive HTTP requests
- Develop Unit and Integration tests to test WebFlux endpoints
- Become familiar with the basics of WebSockets
- Create a WebSocket endpoint using Spring
- Create a WebSocket client

Need different skills or topics? *If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate. We offer additional Spring, Java / JEE programming, Microservices / Services, REST, TDD / testing, design, application security and other related topics that may be blended with this course for a track that best suits your needs. Our team will*

collaborate with you to understand your needs and will target the course to focus on your specific learning objectives and goals.

Audience

This is an intermediate-level Java development course geared for students experienced with Java and Spring programming essentials. **This course does not cover Java or Spring development basics.**

Pre-Requisites

Take Before: Students should have development skills at least equivalent to the following course(s) or should have attended as a pre-requisite:

- **TT3335** Mastering Spring 5.x
TT3335 Mastering Spring 5.x Developer Boot Camp

Agenda

Session: Introduction to Reactive Programming

Lesson: Introduction to Reactive Programming

- Reactive Manifesto
- Introduce ReactiveX
- ReactiveX implementations
- The Observer, Iterator pattern and functional programming
- Discuss hot and cold publishers

Lesson: Reactive Streams API

- Introduce the Reactive Streams specification
- Publisher and Subscribers
- `java.util.concurrent.Flow`
- Transformation of Messages (Processor)
- Controlling messages
- Tutorial: Setup Eclipse for Using Maven
- Tutorial: Setup IntelliJ for Using Maven

Session: Project Reactor

Lesson: Introduction

- Introduce the Reactor Building blocks
- Flux and Mono
- Creating observables
- Subscribing to a stream
- Lab: Project Reactor introduction

Lesson: Testing Event Sources (introduction)

- Testing reactive implementations
- StepVerifier : test sequence of emitted items
- Defining expectations
- TestPublisher: produce test data to test downstream operators
- Lab: Project Reactor Testing

Lesson: Reactive Operators

- Introduce Operators
- Show the use of marble diagrams
- Explain some commonly used operators
- Callback operators
- Lab: Project Reactor Operators

Lesson: Schedulers (Multithreading)

- Thread usage of subscriber and consumer
- Using the subscribeOn method
- Introduce the Scheduler interface
- Using the observeOn method

Lesson: Backpressure

- Strategies for dealing with Backpressure
- 'reactive pull' backpressure
- Lab: Reactor Backpressure

Lesson: Exception Handling

- Handling errors in onError
- Exception handling strategies
- Using onErrorReturn or onErrorNext operators

- Using the retry operators
- The Global Error Handler
- Lab: Reactor Exceptions

Session: Spring Data R2DBC

Lesson: Spring Data Review

- Quick review of Spring Data repositories
- Query return types
- Defining Query methods
- Pagination and sorting

Lesson: R2DBC Repositories

- Reactive Relational Database Connectivity
- Overview of Spring R2DBC
- R2DBC Repositories
- Metadata-Based Mapping
- Querying data using query method
- @Query annotated method
- Query by Example
- Lab: Spring Data R2DBC Repositories

Lesson: R2DBC Template

- Using DatabaseClient and R2dbcEntityTemplate
- Mapping the result to entities
- Querying for data
- Lab: Spring Data R2DBC Templates

Session: Spring WebFlux

Lesson: Introduction

- Non-blocking request processing
- WebFlux configuration
- Annotated endpoint
- Creating a reactive REST endpoint
- Mapping the request
- Path variables and template patterns

- Content type negotiation
- Using ResponseEntity to build the response
- Lab: Introduction to Spring WebFlux

Lesson: WebFlux.fn: Router functions

- Explore WebFlux.fn
- Defining router functions
- Creating the server response
- Implementing handler methods
- RouterFunction and RequestPredicates
- Defining Routes
- Lab: WebFlux, Defining Flows

Session: Spring WebClient

Lesson: Introduction to WebClient

- Making non-blocking, reactive HTTP requests
- WebClient vs RestTemplate
- Creating the WebClient
- Making HTTP requests
- Introduce the UriBuilder
- Retrieve the response data
- Handling the response (exchangeToMono and exchangeToFlux)
- Handling errors
- Lab: WebClient

Lesson: WebTestClient

- Testing WebFlux server endpoints
- Testing controllers or functions
- Define integration tests
- Lab: WebTestClient

Session: Spring Reactive WebSockets

Lesson: Introduction to WebSockets

- Become familiar with the basics of WebSockets
- Understand the HTTP handshake and upgrade
- Name some of the advantages of WebSockets

Lesson: Defining the WebSocket

- Introduce Spring WebSockets
- Implementing the WebSocketHandler
- Configuring the WebSocket
- Creating a Browser WebSocket Client
- Broadcasting to all connected clients
- Lab: Spring WebSocket Introduction

Lesson: WebSocket STOMP

- Streaming (or Simple) text-orientated messaging protocol
- Introduce SockJS
- Connecting to the STOMP endpoint
- Configuring the message broker
- STOMP destinations
- Lab: Spring WebSocket STOMP

Lesson: Reactive WebSocket

- Reactive WebSocket support in Spring
- Implement the reactive WebSocketHandler

Additional Topics: Time Permitting

These topics will be included in your course materials but may or may not be presented during the live class depending on the pace of the course and attendee skill level and participation.

Lesson: View Technologies

- Rendering static and dynamic web pages
- Using Thymeleaf to create the view
- Defining the Controller
- View Configuration
- Lab: WebFlux View technologies

Each participant will receive a Student Guide with course notes, code samples, software tutorials, step-by-step written lab instructions, diagrams and related reference materials and resource links. Students will also receive the project files (or code, if applicable) and solutions required for the hands-on work.

For More Information

Please [contact us](#) or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.