

Reactive Programming with Spring 5

Course 6976 – 16 Hours

Overview

Reactive programming is more and more relevant, as concurrencies grows and data streaming becomes the favorite way to integrate with DBs and Web clients (repsponsive pages and IoT). The traditional 'Request-Oriented' way of handling client requests causes serious back-pressure and requires lot of computing power in order to supply enough threads. Reactive programming solves this MVC Controllers issue by splitting and forking requests in a transparent manner, which allows limited number of threads to handle multiple requests. This is mostly addresses the new 'Reactor' approach which also fully supported in HTTP.

This course explores the building blocks of Reactive Programming in general, and the need. The course then relates to Java 9 Flow API and focuses on practical reactive programming with Spring 5 WebFlux (Flux & Mono). The course allows you to define reactive services, use reactive REST web-services, work reactively with NoSQL (MongoDB) and consume reactive message streams from MQ (RabbitMQ).

Who Should Attend

- Java developers that wants to master reactive programming
- Some background in Java 8 Functional programming
- Spring developers that wants to exploit Spring5 reactive capabilities
- Developers that wants to build HTTP2 compliant, Reactive REST web-services
- Developers that wants to build Reactive applications

Prerequisites

- Experience in basic Java & Spring
- Some background in REST
- Some experience in working with SpringMVC & SpringBoot

Course Contents

Introduction to reactive programming

- What is?
- The need
- Back-pressure
- Data streaming
- Evolution in Java (7,8,9)



RxJava

- Introduction
- Flow
- Publisher
- Subscriber
- Processors
- Signals & signal types

Spring Reactive Programming

- Supported implementations
- Installing / setting environment
- Mono API
- Flux API
- From Mono to Flux
- From Flux to Mono
- Parallel Flux
- Connectable Flux

Spring WebFlux

- Introduction
- SpringBoot2 configuration
- Relevance to HTTP2
- SpringMVC vs WebFlux
- Reactive controllers
- Browser support

Reactive DAO

- Introduction
- SpringBoot2 configuration
- Reactive MongoDB
- Reactive Spring Repositories
- Creating reactive services

Reactive MQ

- Introduction
- Configuring Reactive RabbitMQ
- Setting MessageListenerContainer
- Creating Message Emitter
- Publishing & Consuming