

교육 과정 소개서.

Spring Webflux 완전 정복 : 코루틴부터 리액티브 MSA
프로젝트까지



강의정보

강의장	온라인 강의 데스크탑, 노트북, 모바일 등
수강 기간	평생 소장
상세페이지	https://fastcampus.co.kr/dev_online_webflux
강의시간	54시간
문의	고객센터

강의특징

나만의 속도로 **낮이나 새벽이나 내가 원하는 시간대**에 나의 스케줄대로 수강

원하는 곳 어디서나 **시간을 쪼개 먼 거리를 오가며 오프라인 강의장을 찾을 필요 없이 어디서든 수강**

무제한 복습 **무엇이든 반복적으로 학습해야 내것이 되기에 이해가 안가는 구간 몇번이고 재생**



강의목표

- 체계적인 커리큘럼을 통해 Webflux를 통한 리액티브 웹 개발을 마스터 할 수 있습니다.
- 기본인 스프링과 비동기 코드의 가독성을 높여주는 코루틴을 한 강의에서 만날 수 있습니다.
- 기획부터 부하 테스트까지 실무 프로세스로 고성능 리액티브 MSA를 구축할 수 있습니다.

강의요약

- Webflux 완벽 마스터 탄탄한 커리큘럼
- Java와 Kotlin(Coroutine)기반으로 학습
- 리액티브 MSA 실무 프로젝트
- r2DBC와 Spring Data로 고효율 데이터베이스 구축
- 유지보수 학습으로 Webflux 완벽 커버



강사

김태우

약력

- 현) 당근마켓
로컬 커머스 아키텍처 설계 및 구축, Webflux 등 Reactive Stack을 활용하여 리소스 효율과 응답 속도 향상
- 전) 카카오
Spring mvc, JPA 등 Servlet stack을 활용하여 챗봇 빌더 개발, 리팩토링과 클린 아키텍처 도입으로 구조적 문제 해결
- 전) 쿠팡
Spring boot, memcached, react, redux를 활용한 풀스택 개발

강사의 한 마디

안녕하세요. 수강생 여러분, 김태우입니다.
지역 기반의 커머스 서비스를 만들며, 설계부터 개발, 운영에 이르기까지 다양한 업무에 참여하고 있습니다. 제가 작업하던 프로젝트에 Webflux를 도입했더니 훨씬 적은 서버로 적은 리소스를 활용하여 서비스를 제공할 수 있었고, 요청량이 늘어나도 안정적으로 그리고 탄력적으로 요청을 처리하는 좋은 성능에 놀랐습니다. Webflux에 대한 자료 부족과 비동기 개발의 복잡성 때문에 걱정이라면 이 번 강의를 통해 새로운 패러다임을 도전해보시고, 저와 함께 Webflux를 통해서 서비스의 성능을 향상시키고 개발자로서 크게 성장한 경험을 얻어가세요!

CURRICULUM

01.

Reactive
programming

파트별 수강시간 08:08:41

Ch 01. 전체 커리큘럼과 강의진행 소개
01. 전체 커리큘럼과 강의 진행 소개
Ch 02. Reactive Programming
01. 챕터 소개
Ch 03. 비동기 Programming
01. 함수 관점에서 동기와 비동기의 차이
02. 함수 관점에서 blocking과 non-blocking의 차이
03. IO 관점에서 blocking과 non-blocking의 차이
Ch 04. CompletableFuture
01. Future 인터페이스
02. CompletionStage 인터페이스
03. CompletableFuture 클래스
04. CompletableFuture 사용해서 비동기 로직 처리하기
05. CompletableFuture 정리
Ch 05. Reactive Streams
01. Reactive streams의 역사
02. Reactive manifesto
03. Reactive programming
04. Reactive streams
05. Cold & Hot Publisher 구현
06. Reactive streams 정리
Ch 06. Reactive Streams 구현 라이브러리 소개
01. Reactor 소개, Publisher 중심
02. RxJava 소개, Publisher 중심
03. Mutiny 소개, Publisher 중심



CURRICULUM

01.

**Reactive
programming**

파트별 수강시간 08:08:41

Ch 07. Java NIO
01. Java IO - InputStream
02. Java IO - OutputStream
03. Java NIO - Buffer
04. Java NIO - DirectByteBuffer와 Channel
05. Java IO, NIO 사용해서 소켓 서버 구현
06. Java AIO(NIO2)
Ch 08. Reactor 패턴
01. Selector 소개
02. Selector 사용해서 소켓 서버 고도화
03. epoll 소개
04. Reactor pattern 소개
05. Reactor pattern 사용해서 http 서버 구현
06. Reactor pattern 정리
Ch 09. [부록] A. Proactor pattern
01. Reactor 패턴과 비교하여 개념 설명

CURRICULUM

02.

Spring Webflux

파트별 수강시간 10:32:10

Ch 01. Spring Reactive Stack

- 01. 챕터 소개
- 02. spring servlet stack
- 03. spring reactive stack

Ch 02. Netty

- 01. eventLoop
- 02. channelHandlerContext
- 03. channelHandler
- 04. netty로 Echo server 구현 예제
- 05. bootstrap
- 06. netty 서버 구현
- 07. netty 정리

Ch 03. Reactor

- 01. reactor 복습
- 02. subscribe
- 03. sequence 생성
- 04. scheduler와 쓰레드
- 05. 에러 핸들링
- 06. 결합
- 07. 유용한 연산자
- 08. context
- 09. reactor로 비동기 로직 처리
- 10. reactor 정리

CURRICULUM

02.

Spring Webflux

파트별 수강시간 10:32:10

Ch 04. Spring Webflux

- 01. spring webflux 구조
- 02. httpHandler
- 03. webHandler
- 04. webHandler codec
- 05. webFilter
- 06. webExceptionHandler

Ch 05. Spring Webflux 심화

- 01. webHandler에 Spring 적용
- 02. dispatcherHandler
- 03. functional endpoints
- 04. annotated controllers
- 05. handler method
- 06. spring webflux로 서버 구현
- 07. webClient
- 08. spring security reactive
- 09. client, security 서버에 적용
- 10. spring webflux 심화 정리

Ch 06. Server Sent Event

- 01. event streaming
- 02. server sent event
- 03. sse로 알림 서버 구현
- 04. server sent event 정리

Ch 07. WebSocket

- 01. simpleUrlHandlerMapping
- 02. websocketHandler
- 03. websocket으로 채팅 서버 구현
- 04. websocket 정리



CURRICULUM

03.

Spring Data
Reactive

파트별 수강시간 08:24:01

Ch 01. 챗터 소개
01. 챗터 소개
Ch 02. Mysql, MongoDB, Redis
01. MySQL 소개
02. MongoDB 소개
03. Redis 소개
04. 데이터베이스 사용 사례
Ch 03. R2dbc mysql
01. R2dbc 소개
02. R2dbc MySQL
03. R2dbcEntityTemplate
04. Object mapping
05. Metadata mapping
06. Codec
07. R2dbcEntityOperations
08. R2dbcRepository
09. Query method
10. 내용 정리
11. 실습
12. 정리

CURRICULUM

03.

Spring Data
Reactive

파트별 수강시간 08:24:01

Ch 04. Reactive MongoDB

- | |
|-----------------------------|
| 01. Reactive mongoDB driver |
| 02. MongoDB Document |
| 03. Spring data mongoDB |
| 04. Object mapping |
| 05. MongoOperations |
| 06. ReactiveMongoRepository |
| 07. Query method |
| 08. 실습 |
| 09. 정리 |

Ch 05. Reactive Redis

- | |
|---------------------------|
| 01. Redis 소개 |
| 02. Lettuce |
| 03. ReactiveRedisTemplate |
| 04. ReactiveOperations |
| 05. 실습 |
| 06. 정리 |

Ch 06. Spring data reactive의 다른 라이브러리 소개

- | |
|--|
| 01. Spring data neo4j reactive |
| 02. Spring data elasticsearch reactive |



CURRICULUM

04.

Kotlin Coroutines

파트별 수강시간 05:20:23

Ch 01. 챗터 소개
01. 챗터 소개
Ch 02. Coroutine 기초
01. suspend 함수
02. Finite state machine
03. Conitnation passing style
04. CPS와 FSM 적용
05. 동기, 비동기, Coroutine 비교
06. 비동기에 FSM 적용
07. 비동기에 CPS 적용
08. 비동기를 Coroutine으로 전환
09. Coroutine 도입
Ch 03. CoroutineContext
01. CoroutineContext 이해
02. CoroutineContext 전파
03. CoroutineContext Element
04. CoroutineExceptionHandler
05. CoroutineDispatcher
Ch 04. CoroutineScope
01. CoroutineScope와 coroutine builder
02. Coroutine
03. Job cancellation
Ch 05. Coroutine 심화
01. Flow 소개
02. Flow 연산자
03. Channel 소개
Ch 06. 실습
01. Spring websocket, mongo에 coroutine 적용
02. Spring webflux, r2dbc에 coroutine 적용

CURRICULUM

05.

Spring Reactive Test

파트별 수강시간 08:04:30

Ch 01. 챗터 소개
01. 챗터 소개
Ch 02. 단위 테스트
01. 단위 테스트 소개
02. JUnit 소개
03. JUnit Test와 생명주기 메소드
04. JUnit Assertions - 비교
05. JUnit Assertions - 실행
06. JUnit Assumptions
07. JUnit 테스트 조건부 실행
08. JUnit 유틸리티
09. Mockito 소개
10. Mockito Mock과 Stubbing
11. Mockito ArgumentMatchers
12. Mockito verify
13. Mockito Spy
14. Mockito 어노테이션
15. Reactor test 소개
16. Reactor StepVerifier와 Step
17. Reactor StepVerifier와 LastStep
18. Reactor withVirtualTime
19. Reactor TestPublisher
20. private 메소드 테스트
21. 단위 테스트 실습
Ch 03. 슬라이스 테스트
01. 슬라이스 테스트 소개
02. Mockito와 MockBeans
03. WebTestClient 생성
04. WebTestClient 검증 (1)
05. WebTestClient 검증 (2)
06. WebFluxTest
07. DataR2dbcTest
08. DataR2dbcTest sql init
09. DataMongoTest
10. WebFluxTest 실습
11. DataR2dbcTest 실습

CURRICULUM

05.

Spring Reactive Test

파트별 수강시간 08:04:30

Ch 04. 통합 테스트
01. 통합 테스트 소개
02. MockWebServer
03. Testcontainers 소개
04. MySQLContainer
05. MongoDBContainer
06. SpringBootTest
07. 통합 테스트 실습



CURRICULUM

06.

Reactive
Microservice

파트별 수강시간 07:45:08

Ch 01. 챗터 소개
01. 챗터 소개
Ch 02. Reactive microservice 구성 소개
01. Microservice 구성 소개
02. Reactive 반영하기
Ch 03. Spring cloud circuit breaker
01. Circuit breaker 소개
02. Circuit breaker closed
03. Circuit breaker closed 전환
04. Circuit breaker open
05. Circuit breaker half open
06. Circuit breaker 설정
Ch 04. Spring cloud gateway
01. gateway 소개
02. Spring cloud gateway 소개
03. Custom route predicate
04. 기본 route predicate (1)
05. 기본 route predicate (2)
06. Custom gatewayFilter
07. 기본 gatewayFilter (1)
08. 기본 gatewayFilter (2)
09. Global filter
Ch 05. Spring cloud stream과 Kafka
01. Kafka 소개
02. Spring cloud stream 소개
03. Spring cloud stream kafka binder 소개
Ch 06. Reactive microservice 만들기
01. Reactive application 서버 구현
02. Corouine 기반의 reactive application 서버 구현
03. Kafka와 Spring cloud function으로 큐 구현
04. Spring cloud gateway로 api gateway 구현
05. Sprigin cloud circuit breaker 적용
06. 통합테스트 만들고 실행하기

CURRICULUM

[Bonus].**코틀린 문법**

파트별 수강시간 06:02:17

Ch 01. 코틀린 문법 기초

- | |
|----------------|
| 01. 개발 환경 구성하기 |
| 02. 변수 |
| 03. 함수 |
| 04. 흐름제어 |
| 05. 널 안정성 |
| 06. 예외처리 |
| 07. 클래스와 프로퍼티 |
| 08. 상속 |
| 09. 인터페이스 |
| 10. 열거형 |

Ch 02. 코틀린 문법 심화

- | |
|-----------------|
| 01. 컬렉션 |
| 02. 데이터 클래스 |
| 03. 싱글톤과 동반객체 |
| 04. 실드 클래스 |
| 05. 확장 함수 |
| 06. 제네릭 |
| 07. 지연초기화 |
| 08. 페어와 구조분해할당 |
| 09. 스코프 함수 |
| 10. 고급 예외처리 |
| 11. 람다로 프로그래밍하기 |

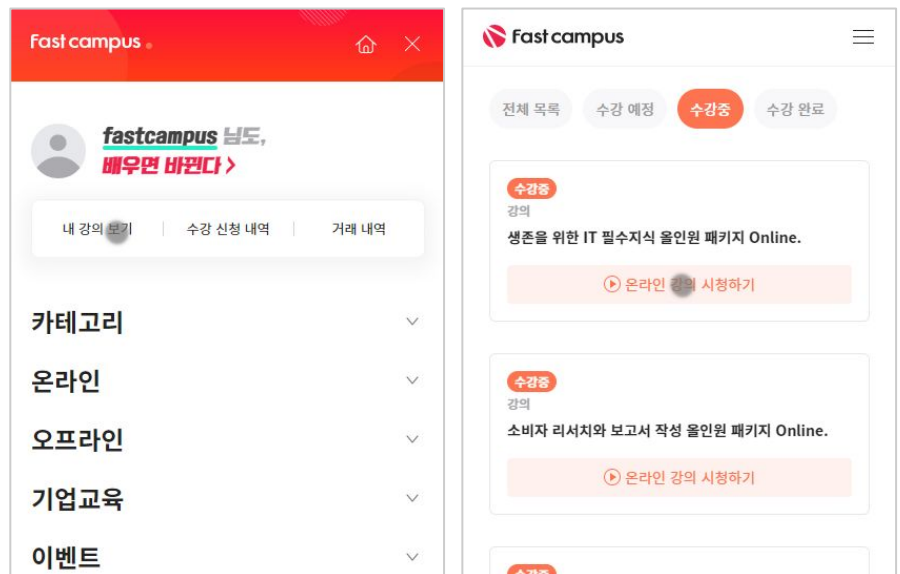


주의 사항

- 상황에 따라 사전 공지 없이 할인이 조기 마감되거나 연장될 수 있습니다.
- 패스트캠퍼스의 모든 온라인 강의는 아이디 공유를 금지하고 있으며 1개의 아이디로 여러 명이 수강하실 수 없습니다.
- 별도의 주의사항은 각 강의 상세페이지에서 확인하실 수 있습니다.

수강 방법

- 패스트캠퍼스는 크롬 브라우저에 최적화 되어있습니다.
- 사전 예약 판매 중인 강의의 경우 1차 공개일정에 맞춰 '온라인 강의 시청하기'가 활성화됩니다.



환불 규정

- 온라인 강의는 각 과정 별 '정상 수강기간(유료수강기간)'과 정상 수강기간 이후의 '복습 수강기간(무료수강기간)'으로 구성됩니다.
- 환불금액은 실제 결제금액을 기준으로 계산됩니다.

수강 시작 후 7일 이내	100% 환불 가능 (단, 수강하셨다면 수강 분량만큼 차감)
수강 시작 후 7일 경과	정상(유료) 수강기간 대비 잔여일에 대해 환불규정에 따라 환불 가능

※ 강의별 환불규정이 상이할 수 있으므로 각 강의 상세페이지를 확인해 주세요.