

교육 과정 소개서.

[오프라인] 고성능 서버 개발을 위한 Async Non-blocking in
Practice



코스요약

| | |
|---------|--|
| 코스명 | [오프라인] 고성능 서버 개발을 위한 Async Non-blocking in Practice |
| 기간 | 2024. 9. 21 (토) ~ 2024. 10. 19 (토) |
| 일정 | 1~4회차 14:00 - 18:00 (4시간) 5회차(mini workshop) 10:00 - 17:00 (6시간) 매주 토요일, 5주간 5회 총 22시간 |
| 장소 | 패스트캠퍼스 강남 강의장 (강남역 4번 출구, 미왕빌딩) |
| 준비물 | 개별 노트북 (사양 무관) |
| 수강료(정가) | 1,200,000원 |
| 문의 | 고객센터 |

수강신청안내

결제가 완료된 뒤, 패스트캠퍼스 홈페이지 내 마이페이지 > 내 강의 보기를 통해 강의의 등록현황을 확인하실 수 있습니다.

결제를 완료하신 수강생 분들께는 강의에 대한 자세한 사항을 개강 일주일전, 3일전, 1일전 각각 메일과 문자를 통해 별도로 안내드립니다.



코스목표

백엔드 성능 개발의 정점 '비동기 논블로킹' 18년차 베테랑 개발자가 이해될때까지 오프라인으로 알려드립니다!

코스정보

본 강의는 오직 실무자를 위해 이론 10% 실습 90% 로 구성된 강의입니다.
그저 '순차 실행'과 '작업 흐름' 같은 개념으로만 이해하는 것이 아니라, 실습을 통해 강의 시간 내에 내용을 온전히 이해하고 적용까지 해볼 수 있습니다.

강사소개



정화수

[이력

현) 네카라쿠배 중 한 곳

전) 윈스토어 | SK Planet | LG CNS

- 패스트캠퍼스 '백엔드 실전 역량 강화를 위한 5개 Kotlin 프로젝트' 강의의 Project 5. 담당 강사

“안녕하세요. 저는 LG CNS, SK Planet, 윈스토어 그리고 지금은 커머스 플랫폼 회사에서 백엔드 개발, 데이터 파이프라인 구축 및 시스템 디자인 작업 등을 담당해 왔습니다. 저는 운이 참 좋았습니다. 훌륭한 선배들과 자랑스러운 동료들을 참 많이 만났고, 차근차근 단계를 밟아나가면서 부족한 기술을 채워넣어 한 걸음씩 착실하게 성장할 수 있었습니다. 하지만 누구나 그런 기회를 얻는 것은 아니란 것도 잘 알고 있습니다. 준비가 부족하다고 스스로를 자책하다 보면, 어느 순간 몸과 마음이 지쳐 고슴도치처럼 변해버리는 자신을 발견할 수도 있습니다.

어떤 분야든 손쉬운 지름길은 없지만, 그것이 달성할 수 없는 불가능한 장벽은 아닙니다. 어제보다 조금씩만 더 배우고 실력을 쌓아간다면, 나중에 되돌아보았을 때 어느새 꽤 높은 곳에 올라와 있는 자신을 발견할 수 있을 것입니다. 그동안 선배들과 동료들에게 도움을 받았던 것처럼 이제는 제가 여러분께 도움을 드릴 수 있다고 생각하니 벌써부터 가슴이 뿌듯하네요. 강의실에서 뵙겠습니다. 감사합니다!

”



커리큘럼

1주차 • 비동기 프로그래밍 이론 (4H)

✔ 학습 내용

비동기 NIO(Non-blocking IO) 어플리케이션의 기본 개념을 학습하고, 이론을 검증해보기 위한 pure kotlin 프로그램 구현합니다.

✎ 이론 학습 내용

1. Async NIO 개념
2. Spring Webflux
3. Coroutine
4. Performance comparison

📄 실습 내용

1. Coffee Order
 - Java Thread Block I/O
 - Kotlin Coroutines NonBlock I/O
2. Context switching cost 비교 (Java Thread vs Kotlin coroutine)

2주차 • Spring MVC (4H)

✔ 학습 내용

기본기를 복습하고 추후 대조에 필요한 비교군 준비를 위해, Spring MVC 기반의 동기식 API 서버를 구축합니다.

📄 실습 내용

1. 구현내용 설명
 - 간단한 게시판 서비스
 - API, DATA
2. 환경구성
 - Spring MVC
 - Docker > MySQL
3. 구현
4. 테스트
 - Postman
 - IntelliJ dot http
 - Junit
 - Mockito



커리큘럼

3주차 ● Spring Webflux (by Coroutine) (4H)

✔ 학습 내용

본격적인 비동기 서비스 구현을 위해, Spring Webflux 기반의 비동기 API 서버를 구축합니다.

■ 실습 내용

1. Corutine 실습
 - ANIO programing in Coroutine
2. Spring Webflux 설명
3. 환경구성
 - Spring Webflux
 - Kotlin Coroutine
4. 구현
5. 테스트
 - Kotest

4주차 ● 성능비교 (4H)

✔ 학습 내용

2인 1조로 부하를 쓰는 쪽과 받는 쪽이 되어 성능 테스트를 수행합니다. 눈으로 직접 비동기 NIO 서버의 성능을 확인하며, 발생하는 에러 상황을 해결합니다.

■ 실습 내용

1. 성능 테스트 도구 셋업
 - Lucust 셋업
 - 성능테스트 스크립트 구현
2. 성능비교
 - 기 구현해놓은 Spring MVC/Webflux를 이용해 성능 비교
 - 매우 많은 돌발상황이 발생시키고 해결하는 과정 진행 (ex. 개발시 자주 하는 쿼리 로깅만 on 시켜도 webflux 성능은 mvc 보다 높지 않음)
3. Docker 를 이용한 성능비교
 - netty의 성능이슈로 윈도우 OS 를 사용하는 수강생은 성능비교시 차이점을 확인하지 못함
 - 때문에 2,3주차에서 실습한 프로그램을 docker로 만든 후, 이를 파라미터를 이용해 만져봄



커리큘럼

5주차 ● 실무형 심화 개발 #1 (3H)

: 오전

✔ 학습 내용

서비스 운영에 꼭 필요한 요구사항을 이해하고, Webflux 환경에서 이를 해결하기 위한 방법을 학습합니다. 총 6가지의 각 요구사항을 해결하기 위한 고도화 작업을 진행합니다.

📖 실습 내용

1. Trace ID logging

- 요구사항 : 병렬처리 로그더미에서 특정 request 로그만 필터링해서 확인

2. Error Handling

- 요구사항 : 모든 오류를 공통 모듈에서 로깅

3. Request payload logging

- 요구사항 : 사용자 요청 파라미터를 로깅

4. Circuit Breaker

- 요구사항 : MSA 에서 서버간 오류전파를 막기 위한 호출 차단기 구현

※ 중간에 1시간의 점심 시간이 있습니다.

5주차 ● 실무형 심화 개발 #2 (3H)

: 오후

✔ 학습 내용

서비스 운영에 꼭 필요한 요구사항을 이해하고, Webflux 환경에서 이를 해결하기 위한 방법을 학습합니다. 총 6가지의 각 요구사항을 해결하기 위한 고도화 작업을 진행합니다.

📖 실습 내용

5. Redis & Cache

- 요구사항 : API 조회속도 향상

(Redis 연동 실습 2시간)

6. Lock

- 요구사항 : 분산환경에서 효과적으로 데이터 transaction 을 처리하고 싶음

7. 과정 정리



수강환경

패스트캠퍼스 강남강의장

서울시 강남구 강남대로 364 미왕빌딩 11층 패스트캠퍼스



❖ 강의에 따라 강의장이 변경될 수 있습니다.