

교육 과정 소개서.

초격차 패키지 : 백엔드 실전 역량 강화를 위한 4개 도메인
프로젝트 (Kotlin/Spring)



강의정보

강의장	온라인 강의 데스크탑, 노트북, 모바일 등
수강 기간	평생 소장
상세페이지	https://fastcampus.co.kr/dev_online_kotlinfinal
강의시간	약 60시간
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

- 다양한 도메인 주제를 통해 기능 개발 뿐만 아니라 프로젝트 설계부터 유지보수까지! 실전 프로젝트로 학습합니다.
- 평균 경력 11년! 시니어, 리드급 강사님의 도메인에 특화된 개발 노하우와 인사이트까지 프로젝트를 진행합니다.
- 대용량 트래픽 처리, 성능 테스트, 모니터링 등 언어와 무관하게 시니어 개발자라면 반드시 고려해야 할 백엔드 심화 기술을 익힙니다.

강의요약

- 각 백엔드 실전/심화 역량 강화를 위해 설계된 프로젝트 구성! 깊이 있는 4개 프로젝트를 통해 백엔드 기술력을 제대로 업그레이드 해 봅니다.
- 실무에서 빈번하게 사용하는 결제 대행사(PSP)를 통해 실제 작동하는 결제 시스템을 구축하고 비동기의 여러 개념을 적용합니다.
- 커머스 정산 전문가 강사님께 상황별 데이터 정산 방식과 성능 튜닝 미션까지 익히고 바로 실무에 적용해보세요!
- 친숙한 배달 서비스를 통해 실시간으로 이루어지는 양방향 통신과 대용량 데이터 처리까지 시니어 급의 실무 경험을 얻을 수 있습니다.
- IT 대기업의 전체 개발 프로세스와 각 단계에서 의사 결정 방법과 트렌디한 기술을 심도 깊게 배울 수 있습니다.



강사

정화수	과목	- Project 4. 코루틴 활용 비동기식 결제 서비스
	약력	- 현) 원스토어 - 전) SK planet LG CNS
김진실	과목	- Project 3. 배치 퍼포먼스를 고려한 커머스 정산 시스템
	약력	- 현) 29CM - 전) 브랜디 파트장 컬리
이광운	과목	- Project 2. 실시간 데이터 처리에 집중한 대규모 배달 서비스
	약력	- 현) 트렌비 Tech Leader - 전) 쿠팡 Tech Leader
이성훈	과목	- Project 1. 배포 자동화를 적용한 게시판 서비스
	약력	- 현) 카카오택시 - 전) 컬리페이 금융결제원



CURRICULUM

Project 01 .

배포 자동화를 적용한 게시판 서비스

파트별 수강시간 06:59:25

CH01. 강의개요
01. 프로젝트 소개
02. 왜 게시판인가
CH02. 기술 스택 소개
01. Kotlin (vs Java)
02. Kotest (vs JUnit)
03. Swagger (vs RestDocs)
04. ktlint
05. 브랜치 전략 (Github Flow)
CH03. 프로젝트 환경설정
01. 프로젝트 환경세팅 (git, kotest, ktlint, swagger, querydsl 연동)
02. mysql 설치 with docker
03. github actions를 이용한 aws cloud 배포
04. 간단한 동작 확인 (spring profile 설정)
CH04. 게시판 기본기능 개발
01. 요구 사항 분석 & 유스케이스 작성
02. 게시글 API, 도메인 설계
03. 게시글 생성, 수정, 삭제 테스트 작성 및 개발
04. 게시글 조회 & 목록조회 테스트 작성 및 개발
05. 댓글 요구 사항 분석 & 유스케이스 작성
06. 댓글 API, 도메인 설계
07. 댓글 생성, 수정, 삭제 테스트 작성 및 개발
08. 댓글 조회 테스트 작성 및 개발
CH05. 게시판 추가기능 개발
01. 태그 요구 사항 분석 & 유스케이스 작성 & API, 도메인 설계
02. 태그 생성, 삭제 테스트 작성 및 개발
03. 태그 조회 & 검색 개발
04. 좋아요 요구 사항 분석 & 유스케이스 작성 & API, 도메인 설계
05. 좋아요 기능 테스트 작성 및 개발
06. 좋아요 개수 조회 테스트 작성 및 개발
CH06. 게시판 최적화
01. 성능 최적화란 무엇인가
02. 프로그램 최적화 (N1 문제)
03. 데이터베이스 인덱스
04. 레디스 이용한 캐싱
05. 도메인 이벤트



CURRICULUM

Project 02.

실시간 데이터 처리에 집중한 대규모 배달 서비스

파트별 수강시간 15:02:51

CH01. 음식 배달 서비스와 필요 기술 소개
01. 프로젝트 소개
02. 기본적인 음식 배달서비스 흐름
03. 기본적인 음식 배달서비스 시스템 구성
04. 음식 배달 서비스에 필요한 기술들
05. WebSocket 기본
06. WebSocket 채팅 서비스 기본 구조
07. 스프링 부트로 채팅 서버 프로젝트 생성
08. 스프링 부트로 채팅 클라이언트 프로젝트 생성
CH02. 초간단 음식 배달 서비스 기본 구조
01. 고객용 배달앱 기본 구조 알아보기
02. 배달기사용 라이더앱 기본 구조 알아보기
03. 상점용 스토어앱 기본 구조 알아보기
CH03. 초간단 고객용 배달앱 만들어보기
01. 배달앱 기능과 구조 알아보기
02. 배달 API 서버 기능과 구조 알아보기
03. 배달 서비스 테이블 설계
04. 배달 서비스용 데이터베이스 설치
05. 배달 서비스 테이블 생성
06. 배달 API 설계
07. 스프링 부트로 배달 API 서버 생성하기
08. 배달 음식 목록 API 구현하기
09. 배달 음식 상세보기 API 구현하기
10. 배달 음식 장바구니와 체크아웃 API 구현하기
11. 배달 음식 결제하기 API 구현
12. 배달 음식 주문 상세 API 구현하기
13. 배달 음식 주문 목록 API 구현하기
14. 스프링 부트로 초간단 고객용 배달앱 프로젝트 생성



CURRICULUM

Project 02.

실시간 데이터 처리에 집중한 대규모 배달 서비스

파트별 수강시간 15:02:51

CH04. 초간단 라이더용 앱 만들어보기

01. 라이더 서비스 앱 기능과 구조 알아보기

02. 라이더 API 서버 기능과 구조 알아보기

03. 라이더 서비스 테이블 설계

04. 라이더 서비스 데이터베이스 설치

05. 라이더 서비스 테이블 생성

06. 라이더 API 설계하기

07. 스프링 부트로 라이더 API 서버 생성하기

08. 배달 주문 콜 목록 보기 API 구현하기

09. 배달 주문 콜 수락 API 구현하기

10. 배달 완료 API 구현하기

CURRICULUM

Project 02.

실시간 데이터 처리에 집중한 대규모 배달 서비스

파트별 수강시간 15:02:51

CH05. 초간단 배달 상점 서비스 만들어보기

01. 배달 상점 서비스 기능과 구조 알아보기
02. 배달 상점 API 서버 기능과 구조 알아보기
03. 배달 상점 서비스 테이블 설계
04. 배달 상점 서비스 데이터베이스 설치
05. 배달 상점 서비스 테이블 생성
06. 배달 상점 API 설계하기
07. 스프링 부트로 배달 상점 API 서버 생성하기
08. 코틀린으로 배달 주문 목록 API 구현하기
09. 카프카 결제 메시지를 통한 주문 처리
10. 배달 주문 콜 수락 API 구현하기
11. 배달 완료 API 구현하기
12. 스프링부트로 배달 상점 서비스 프로젝트 생성
13. 코틀린으로 배달 주문 목록 구현하기
14. 배달 주문 수락 구현하기
15. 배달 주문 거절 구현하기

CH06. 추가적으로 필요한 기능들

01. 상점 계약하기 및 온보딩
02. 상점 배달음식 등록
03. 상점 정산하기
04. 성능 테스트와 모니터링
05. 대규모 배달 주문 처리를 위한 시스템 구조
06. 전체 배달 서비스 프로젝트 정리



CURRICULUM

Project 03.

배치 퍼포먼스를 고려한 커머스 정산 시스템

파트별 수강시간 08:10:35

CH01. 테이블 설계 - 설계 의도, 필요 구성 요소 등
01. 프로젝트 소개
02. 시작하기 전에, 강의에서 사용할 용어를 정리하기 (정산용어)
03. 정산을 위한 비즈니스 정책을 정하기
04. Seller 테이블 설계, 상품 정보 테이블 설계
05. 주문테이블 설계
06. 클레임 테이블 설계
07. 정산 테이블 설계
CH02. 베이스 데이터를 만들어 보자 - 구매 확정 데이터
01. 목표 설명 : 주문 확정 처리를 위한 데이터를 만들자 왜 구매확정이란 상태를 만들어 주는가
02. 배송완료 후 특정일이 지난 상태의 주문목록 조회를 위한 Reader 만들기
03. 배송완료 후 특정일이 지난 상태의 주문목록 처리를 위한 Processor, Writer 만들기
04. 기능 확장 구매확정 배치의 완성도를 높여보자
CH03. 베이스 데이터를 만들어 보자 - 클레임 데이터
01. 목표 설명 : 취소, 반품, 교환이 완료상태가 되면 정산에 필요한 데이터를 만들자
02. 먼저 각 클레임이 완료된 상태에 대한 정의를 먼저 하자
03. 진행 중인 클레임이 있는 경우, 구매확정 대상 Reader에서 해당 주문건은 제외 되도록 Reader를 수정하자
04. 클레임이 완료되었을 때, 클레임 완료시간이 업데이트 되도록 하자
CH04. 배치 job을 통해 정산 데이터를 만들자. - daily 정산 배치
01. 목표설명: 각 Seller별, 구매확정, 클레임 완료 건을 가져와 일일 정산 데이터를 만들자
02. 구매확정 데이터를 조회하는 Reader만들기
03. 구매확정 데이터를 계산을 위한 Processor만들기
04. 상품타입에 따른 세금계산기만들기
05. 위탁/매입에 따라 구매확정 주문에 대한 Seller 정산 (+)금액 계산하기
06. 처리된 데이터 저장을 위한 Writer 만들기
07. 클레임 완료된 목록을 조회하는 Reader만들기
08. 각 클레임에서 발생한 Seller 정산 (-)금액 계산을 위한 Processor만들기
09. 처리된 데이터 저장을 위한 Writer 만들기



CURRICULUM

Project 03.

배치 퍼포먼스를 고려한 커머스 정산 시스템

파트별 수강시간 08:10:35

CH05. 배치 job을 통해 정산 데이터를 만들자. - monthly 정산 배치
01. 목표설명 : 각 daily 정산 데이터를 기준으로 Seller에게 정산할 한달 정산금 만들기
02. daily 정산 (+,-)금액을 합산하는 Monthly 데이터 만들기
03. 데일리 정산 목록을 조회하는 Reader 만들기
04. 데일리 정산 목록을 합산하는 Processor 만들기
CH06. [고도화]정산 배치의 성능과 안정성을 위한 처리를 해보자
01. 실패한 메시지 처리를 슬랙으로 알람 보내기
02. 금액 정합성을 위한, 금액 계산 TestCase 작성
03. daily 정산 (,-)금액을 만들기 위한 벌크 데이터 생성
04. 배치 성능을 위한 리팩토링을 진행하자
05. 배치 성능을 확인 하자
06. Reader를 더 빠르게 읽을 수 있는지 확인해 보자
07. 저장할 때는 어떤 방식이 더 좋은가를 비교해보고 변경하자
CH07. [고도화] 카프카를 사용해 실시간으로 정산 데이터를 만들어보자
01. 목표 설명 : 구매확정, 클레임 완료상태를 Kafka를 통해 Producing하고, consuming을 통해 정산에 필요한 데이터를 만들자
02. 설계 의도 : 왜 kafka를 사용하는 가?
03. 고객이 구매확정을 먼저 하는 경우, Kafka Topic에 해당 상태를 Producing하자
04. 취소, 반품이 완료 될 때, Kafka Topic에 해당 상태를 Producing하자
05. Kafka Topic에 쌓인 구매확정 완료건을 Consuming하자
06. Kafka Topic에 쌓인 클레임 완료건을 Consuming하자

CURRICULUM

Project 04.

코루틴을 활용한 비동기식 간편결제 서비스

파트별 수강시간 29:38:50

CH01. 비동기 서비스 소개: 오리엔테이션

01. 강의개요
02. 비동기 호출이란 ?
03. Servlet / Reactive

CH02. 동기 서비스 구현 (Spring MVC)

01. 구현내용 설명 및 개발환경 구성
02. Hello world API 구현
03. CRUD 서비스 구현 (MVC)
04. Unit 테스트 (JUnit)

CH03. 비동기 서비스 구현 (Spring Webflux)

01. 구현내용 설명 및 개발환경 구성
02. Hello world API 구현
03. CRUD 서비스 구현 (Webflux)
04. Unit 테스트 (JUnit)

CH04. 비동기 서비스 구현 (Spring Coroutine)

01. Reactor 구현의 난점
02. Coroutine 실습
03. Coroutine 이란
04. 구현내용 설명 및 개발환경 구성
05. Hello world API 구현
06. CRUD 서비스 구현 (Coroutine)
07. Unit 테스트 (Kotest)

CH05. 비동기 서비스 부하 테스트

01. Locust 셋업
02. Spring MVC / Webflux 성능비교 #1
03. Spring MVC / Webflux 성능비교 #2



CURRICULUM

Project 04.

코루틴을 활용 비동기식 간편결제 서비스

파트별 수강시간 29:38:50

CH06. 실무형 심화개발
01. Request 단위로 로그를 묶어서 볼 수 있는 trace id 구현
02. 에러응답 추가 등을 위한 Error Handler 구현
03. Request 파라미터 로깅
04. Circuit breaker & Rate Limiter 구현
05. R2DBC DbClient 를 통한 native query 호출
06. Reactive redis 연동
07. Redis 연산
08. Cache 구현
09. Optimistic Lock & Pessimistic Lock
10. Distributed Lock 구현
CH07. 결제 서비스 구현
01. Payment Gateway 소개
02. Auth-Caputre 패턴
03. 결제 시스템 아키텍처 설계
04. 환경 구성
05. 결제 서비스 구현 #1 - 모델 구현
06. 결제 서비스 구현 #2 - 주문 서비스 구현
07. 결제 서비스 구현 #3 - 결제 연동
08. 이력 조회 API 구현
CH08. 결제 서비스 기능 확장
01. 시나리오 소개
02. 시나리오 1. 결제 재처리
03. 시나리오 2. 주문이력 성능개선 (분석 및 설계)
04. Kafka
05. Elasticsearch
06. 시나리오 2. 주문이력 성능개선 (구현)
CH09. 과정 마무리: 회고
01. 회고

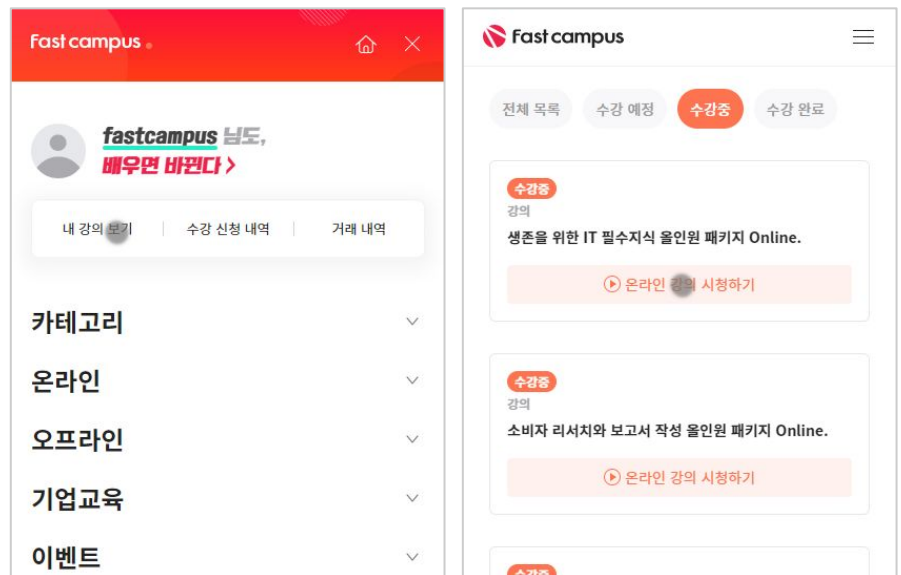


주의 사항

- 상황에 따라 사전 공지 없이 할인이 조기 마감되거나 연장될 수 있습니다.
- 패스트캠퍼스의 모든 온라인 강의는 **아이디 공유를 금지**하고 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

- 강의 시수, 형태에 따라 환불 규정이 상이 하므로 반드시 각 강의 상세페이지 하단 [환불 규정] 과 [패스트캠퍼스 취소/환불 정책] 내용을 확인해 주시기 바랍니다.
- 패스트캠퍼스 취소/환불 정책 : <https://fastcampus.co.kr/info/policies/refund>