

교육 과정 소개서.

마이크로 프론트엔드(MFA:Micro Frontend Architecture)
: 대규모 서비스 설계부터 운영까지 (feat.
모노레포&모듈페더레이션)



강의정보

강의장	온라인 강의 데스크탑, 노트북, 모바일 등
수강 기간	평생 소장
상세페이지	https://fastcampus.co.kr/dev_online_mfa
강의시간	39시간
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

- MFA 개념부터 구현까지 한 번에 정리하고 프로젝트로 직접 구현하며 완벽 이해하는 커리큘럼
- 설계의 시작, 적합한 레포지토리 형식 결정! 모노레포로 설계하고, 최적의 도구까지 한 번에 학습합니다.
- 도메인을 기준으로 분리하여 마이크로 앱을 구현하고 디자인 시스템과 인프라 레벨의 코드를 공통 모듈로 작성합니다.
- 개별 마이크로 앱들을 수많은 회사들이 선택한 Webpack의 Module Federation으로 런타임에 통합합니다

강의요약

- 실무에서 MFA를 도입하는 프로세스를 담고 있습니다.
- MFA, 모노레포, 모듈 번들러, 모듈 페더레이션, 패키지 매니저 등의 실무 초밀착 사용방법을 포함하고 있습니다.
- MFA를 실무에서 도입한 경험이 있는 N사의 강사님이 메인 강의를 진행합니다.
- 카카오의 MFA 도입 케이스 스터디까지 스페셜 강의로 제공합니다.

강사

이웅재

과목

- 마이크로 프론트엔드(Micro Frontend)부터 모노레포까지: 대규모 프로젝트 적합 아키텍처

약력

- 현) N사 프론트엔드 개발팀 리드
- 프론트엔드 아키텍처 설계 및 개발 업무
- Monorepo를 이용하여 대규모 서비스 개발과 운영 중
- Micro Frontend 도입/구축 경험
- 2017 ~ 2023 Microsoft MVP (TypeScript)

CURRICULUM

01.

마이크로 프론트엔드 개념 기초 학습

파트별 수강 시간 03:09:32

Ch 01. 마이크로 프론트엔드 아키텍처 등장

00. 강의개요

01. 웹 개발 발전 과정과 아키텍처의 중요성

02. 기존 아키텍처들과 마이크로 프론트엔드 아키텍처 비교

Ch 02. 마이크로 프론트엔드 아키텍처 필요 상황 진단하기

01. 마이크로 프론트엔드 도입을 위한 문제 식별 및 패턴 소개

02. 마이크로 프론트엔드 도입의 장점

03. 마이크로 프론트엔드 도입의 단점

04. 마이크로 프론트엔드 도입이 필요한 경우

Ch 03. 마이크로 프론트엔드 아키텍처 설계 핵심 원칙과 기술 소개

01. 마이크로 프론트엔드를 규정하는 가장 중요한 요소

02. 서비스를 어떻게 나눠야 할까

03. 마이크로 프론트엔드와 모노레포

04. 모노레포 도입 여부 결정 및 구현 방식 결정

05. 통합 방법 설계 및 적절 기술 스택과 도구 선택

06. 공통 모듈 설계 및 인터페이스 결정

07. 마이크로 프론트엔드와 디자인 시스템

CURRICULUM

02.

기술 이해를 통한 마이크로 프론트엔드 개념 심화 학습 1 : 설계를 위한 모노레포

파트별 수강 시간 08:11:48

Ch 01. 마이크로 프론트엔드 설계의 시작, 모노레포 이해하기

01. 싱글 레포지토리와 멀티 레포지토리, 그리고 모노 레포지토리

02. 모노레포의 특징과 장점 및 단점 살펴보기

Ch 02. 모노레포로 구성된 프론트엔드 프로젝트를 위한 도구 학습하기 1 : 패키지 매니저

01. 프론트엔드 프로젝트와 패키지 매니저 살펴보기

02. npm workspaces

03. yarn workspaces

04. pnpm workspaces

Ch 03. 모노레포로 구성된 프론트엔드 프로젝트를 위한 도구 학습하기 2 : 빌드 시스템 도구

01. 모노레포 빌드시스템 도구와 그 필요성

02. lerna

03. nx

04. rush

05. turborepo

Ch 04. 모노레포로 구성된 프론트엔드 프로젝트를 위한 도구 학습하기 3 : 트랜스파일러와 번들링 도구

01. 프론트엔드 프로젝트 배포를 위한 트랜스파일러와 번들러 이해하기

02. babel

03. rollup 과 terser

04. esbuild

05. swc

06. webpack

07. vite

Ch 05. 모노레포로 구성된 프론트엔드 프로젝트를 위한 적합 도구 최종 선택

01. 패키지 매니저와 모노레포 빌드시스템 도구 및 개발 도구 선택하기

CURRICULUM

03.

기술 이해를 통한 마이크로 프론트엔드 개념 심화 학습 2: 분해와 통합을 위한 여러 기술 비교

파트별 수강 시간 02:42:53

Ch 01. 마이크로 프론트엔드를 구현하는 기술 등장 배경 이해하기

01. 마이크로 프론트엔드 기술 발전 과정

Ch 02. 마이크로 프론트엔드를 구현하는 기술1 : 클라이언트에서 ajax 를 이용한 프래그먼트 통합

01. Ajax 를 이용한 템플릿 통합 예제

02. Ajax 를 이용한 템플릿 통합 결론 및 장단점

Ch 03. 마이크로 프론트엔드를 구현하는 기술2 : 서버에서 proxy 를 이용한 여러 페이지 통합

01. Nginx 와 proxy 를 이용한 페이지 분리 및 통합 예제

02. Nginx 와 proxy 를 이용한 페이지 분리 및 통합 결론 및 장단점

Ch 04. 마이크로 프론트엔드를 구현하는 기술3 : 서버에서 SSI 를 이용한 프래그먼트 통합

01. Nginx 와 SSI (Server-Side Includes) 를 이용한 서버에서의 통합 예제

02. Nginx 와 SSI (Server-Side Includes) 를 이용한 서버에서의 통합 예제 결론 및 장단점

Ch 05. 마이크로 프론트엔드를 구현하는 기술4 : 클라이언트에서 Web Components 를 이용한 프래그먼트 통합

01. Web Components를 이용한 클라이언트에서의 통합 예제

02. Web Components를 이용한 클라이언트에서의 통합 결론 및 장단점

Ch 06. 마이크로 프론트엔드를 구현하는 기술5 : 클라이언트에서 iframe 을 이용한 통합

01. iframe 을 이용한 클라이언트에서의 통합 예제

02. iframe 을 이용한 클라이언트에서의 통합 결론 및 장단점

CURRICULUM

04.

마이크로 앱 통합의 핵심 기술 Module Federation 예제로 익히기

파트별 수강 시간 04:49:38

Ch 01. Module Federation 기술 이해하기

- 01. Webpack5 Module Federation 원리 이해하기
- 02. Vite 의 Module Federation 플러그인 이해하기

Ch 02. 예제로 익히는 Module Federation 기술1 : 기본 설정

- 01. 나의 React 앱에서 다른 서버의 React 컴포넌트를 런타임에 가져와서 사용하기
- 02. 여러 마이크로 앱에서 런타임에 공통적으로 사용하는 라이브러리를 한번만 로드하기
- 03. 두 앱에서 Context를 공유해서 사용하기
- 04. Lazy, Suspense, ErrorBoundary를 통한 런타임 통합

Ch 03. 예제로 익히는 Module Federation 기술2 : 프로덕션을 위한 필수 설정

- 01. 다른 서버의 React 컴포넌트에 TypeScript 의 타입 설정하기
- 02. 다른 서버의 리액트 앱을 가져와서 엘리먼트에 렌더하기 (격리)
- 03. 여러 페이지가 있는 두개의 앱을 하나로 통합하기 (feat. react-router-dom)
- 04. 전역에서의 Redux를 사용할 때,
다른 서버의 React 컴포넌트를 가져오면서 리듀서를 추가하기

Ch 04. 예제로 익히는 Module Federation 기술3 : 심화 과정

- 01. 나의 React 앱에서 다른 서버의 React 컴포넌트를 가져와서 동적으로 사용하기
- 03. 다른 서버의 React 컴포넌트를 가져올 때, 다른 서버를 동적으로 지정하기

CURRICULUM

05.

[프로젝트]
 마이크로
 프론트엔드
 아키텍처를
 이용한 커리어
 플랫폼 서비스
 만들기 : 설계

파트별 수강 시간 00:53:23

Ch 01. 마이크로 프론트엔드 아키텍처 적합성 판단

- 01. 커리어 플랫폼 서비스 프로젝트 소개 및 시나리오
- 02. 마이크로 프론트엔드 아키텍처에 적합한 프로젝트인지 파악하기
- 03. 마이크로프론트엔드 설계를 위해 풀어야 할 문제 식별 및 패턴 확인하기
- 04. 서비스를 나누는 기준 수립하기

Ch 02. 프로젝트 모노레포 설계

- 01. 모노레포 도입 여부 결정 및 구현 방식 결정
- 02. 모노레포 환경 구축 (turborepo, pnpm)

Ch 03. 패키지 설계

- 01. 통합 방법 설계 및 적절 기술 스택과 도구 선택하기
- 02. 공통 모듈 설계 및 인터페이스 결정하기

CURRICULUM

06.

[프로젝트]
마이크로
프론트엔드
아키텍처를
이용한 커리어
플랫폼 서비스
만들기 : 공통
모듈 및 마이크로
앱 구현하기

파트별 수강 시간 09:08:17

Ch 01. 공통 모듈 개발하기1 : UI 라이브러리 패키지 만들기

01. UI 라이브러리 패키지 생성 및 설정

02. 컴포넌트 제작 및 빌드

Ch 02. 공통 모듈 개발하기2 : App Shell 만들기

01. App Shell 의 역할 및 패키지 생성 및 설정

02. react-router-dom 의 lazy 기능 학습 및 내비게이션 구현

03. 공통 레이아웃 설정 및 스타일 충돌을 위한 설정

04. App Shell 에서 필요한 공통 기능 구축레이아웃 및 스타일 충돌을 해결

Ch 03. 마이크로 앱 만들기1 : 포스팅

01. 기술 스택 선정 및 패키지 생성 및 설정

02. 포스팅 서비스 개발하기

Ch 04. 마이크로 앱 만들기2 : 교육 콘텐츠

01. 기술 스택 선정 및 패키지 생성 및 설정

02. 교육 콘텐츠 서비스 개발하기

Ch 05. 마이크로 앱 만들기3 : 인맥 관리

01. 기술 스택 선정 및 패키지 생성 및 설정

02. 인맥 관리 서비스 개발하기

Ch 06. 마이크로 앱 만들기4 : 채용 공고

01. 기술 스택 선정 및 패키지 생성 및 설정

02. 채용 공고 서비스 개발하기

Ch 07. 프래그먼트 만들기

01. 프래그먼트 공유를 위한 준비

02. 일촌 맺기 프래그먼트 컴포넌트 작성

03. 추천 채용 공고 프래그먼트 컴포넌트 작성

04. 프래그먼트 사용하기

CURRICULUM

07.

[프로젝트] 마이크로 프론트엔드 아키텍처를 이용한 커리어 플랫폼 서비스 만들기 : 통합 및 빌드 배포하기

파트별 수강 시간 01:36:11

Ch 01. Module Federation으로 제작된 마이크로 앱 빌드 및 배포하기
01. 마이크로 앱과 프러그먼트 공유를 위한 모듈 페더레이션 설정
02. CI/CD 파이프라인 구축 및 자동화 설정
Ch 02. 개발 후 배포 과정 시뮬레이션
01. 마이크로 앱 추가 개발 후 배포 범위 설정 및 배포
02. 프러그먼트 추가 개발 후 배포 범위 설정 및 배포
03. 마무리

CURRICULUM

08.

[프로젝트]
마이크로
프론트엔드
아키텍처를
이용한 커리어
플랫폼 서비스
만들기 :
운영하기

파트별 수강 시간 01:45:08

Ch 01. 성능 최적화 고려하여 운영하기

01. 프론트엔드 성능 측정 개요와 그 중요성
02. Core Web Vitals 알아보기 (LCP, FID, CLS 등)
03. Lighthouse 도구 사용하기
04. Chrome Performance Tab 활용하기
05. React Developer Tools 활용하기
06. 사례로 알아보는 성능 개선 팁 (초기 로딩 다이어트, 라이브러리 재사용 등)

Ch 02. 보안 이슈

01. 마이크로 프론트엔드의 보안 이슈 해결

Ch 03. 팀 운영하기

01. 팀을 어떻게 나누어야 할까? (마이크로 프론트엔드와 팀의 운영 구조)
02. 팀의 기술적 다양성과 한계

Ch 04. 배포 전략 설정

01. 브랜치 전략과 배포 방식
02. 로컬 개발 환경과 프로덕션 환경

Ch 05. 장애 대응

01. 단일 장애 지점을 피했다면, 장애의 범위를 줄여나가기 위한 전략과 한계

CURRICULUM

09.

부록1. 사례로 배우는 모놀리식에서 마이크로 프론트엔드로의 점진적 전환

파트별 수강 시간 00:51:20

Ch 01. 사례로 배우는 모놀리식에서 마이크로 프론트엔드로의 점진적 전환

- | |
|--|
| 01. 시나리오 소개 (FE 모놀리식 SPA, BE 마이크로 서비스) |
| 02. 모노레포 안의 패키지 분리 (feat. TypeScript) |
| 03. 단순 코드 분할에서 의존성 정리 및 패키지 계층 분리 (feat. TypeScript) |
| 04. 빌드 타임 코드 공유에서 패키지 빌드 후 공유로 점진적 전환 |
| 05. 런타임 공유 패키지와 마이크로 앱 분할 및 독립 개발 환경 구축 |
| 06. 마무리 |

CURRICULUM

10.

부록2. 카카오 MFA 도입 사례 소개: 런타임 인젝션 통합방식을 활용한 카카오의 MFA

파트별 수강 시간 05:04:27

Ch 01. 모두의 MFA, Legacy에서도 적용하는 마이크로 프론트엔드

01. 모두의 MFA, Legacy에서도 적용하는 마이크로 프론트엔드

Ch 02. 실습 환경 및 초기셋팅

01. 실습 환경 및 초기셋팅

Ch 03. 스타일 격리를 위한 실습

01. ShadowDOM에 대한 이해와 직접 구현하는 ShadowDOM

02. react-shadow를 이용한 shadowDOM

Ch 04. runtime injection을 위한 webpack설정

01. react-app-rewired를 통한 webpack 설정 및 umd export 실습 (1)

02. react-app-rewired를 통한 webpack 설정 및 umd export 실습 (2)

03. react-app-rewired를 통한 webpack 설정 및 umd export 실습 (3)

04. react-app-rewired를 통한 webpack 설정 및 umd export 실습 (4)

05. docker와 nginx를 이용한 webpack-dev-server와 production 모드에서의 어플리케이션 차이 확인

06. CSS isolation을 위한 loader 설정

Ch 05. antd, redux 를 활용한 MailList 컴포넌트 작성

01. redux를 이용한 mailList store 작성

02. mail List Component 작성

Ch 06. tailwind, zustand를 활용한 ShoppingList 컴포넌트 작성

01. zustand 및 tailwind 설치 및 설정

02. zustand 및 Shopping store 작성

03. Button 공통 컴포넌트 작성

04. Table 공통 컴포넌트 작성

05. Modal 공통 컴포넌트 작성

CURRICULUM

10.

부록2. 카카오 MFA 도입 사례 소개: 런타임 인젝션 통합방식을 활용한 카카오의 MFA

파트별 수강 시간 05:04:27

Ch 07. Rounting 을 통한 서로 다른 프로젝트에서의 MFA component 소비

- 01. Rounting 을 통한 서로 다른 프로젝트에서의 MFA component 소비
- 02. Header 컴포넌트 작성 (1)
- 03. Header 컴포넌트 작성 (2)
- 04. Header 컴포넌트 작성 (3)
- 05. MFA 컴포넌트 소비를 위한 공통 컴포넌트 작성

Ch 08. legacy 환경 프로젝트 구축

- 01. legacy 프로젝트의 생성 및 초기 셋팅
- 02. Button 공통 컴포넌트 작성
- 03. Table 공통 컴포넌트 작성
- 04. Modal 공통 컴포넌트 작성
- 05. SNS Component 작성
- 06. Routing 및 Header 컴포넌트 작성

Ch 09. Legacy 에서의 MFA

- 01. legacy의 micro component를 소비하는 방법
- 02. ShadowDOM과 asset-manifest.json을 이용한 css isolation 구현
- 03. custom element를 활용한 다양한 인터페이스의 제공

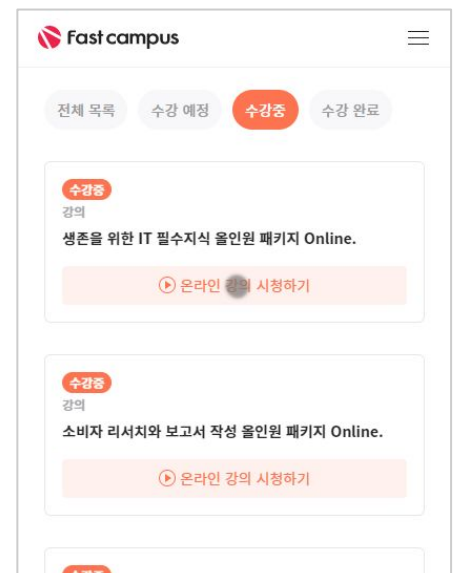
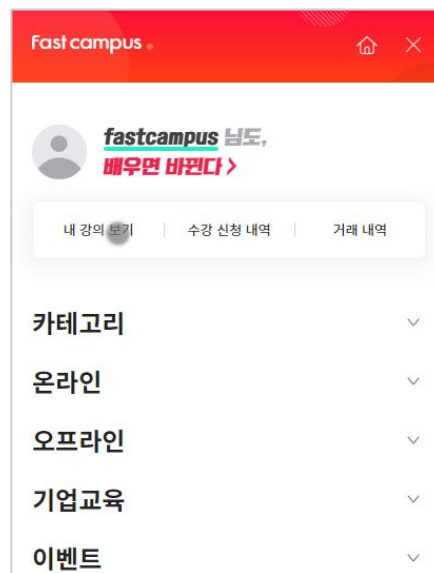


주의 사항

- 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

- 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.