

JIKJI팀 최종발표

김정현 박종하 이한솔 김영현

목차

팀 소개

주제

기술 스택

팀목표

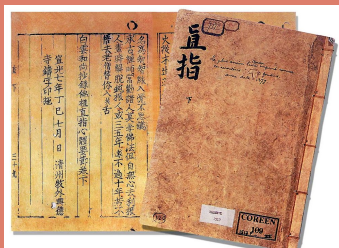
일정관리

기존 설계와 최종 구현

기능 구현

회고

QnA



직지

김정현



이한솔



박종하



김영현





Instagram

선정 이유

- 다양한 서비스들이 혼합되어 있어 MSA 구조로 개발하기 유리
- 심플한 UI => 백엔드 개발에 집중
- 참고 자료가 많아서 개발이 용이
- 대용량 트래픽을 가진 서비스의 서버 아키텍처를 구현해보면서 대형 서비스에서 서버 아키텍처를 선정, 구현하는 원리를 알아보고 싶음
- 여러 서비스들을 함께 개발하면서 협업에 대한 이해도 증진 및 소프트 스킬 습득 가능

Language



Framework



* react 17

Database



DevOps



전문적인 협업 개발 프로세스를 통해 완성도 높은 팀 프로젝트 경험

=> 지속 성장 가능한 개발자 되기

+

체계적인 협업 경험을 통해 소프트 스킬 습득

일일 스크럼 회의

시간 : 매일 아침 10시

장소 : 슬랙 허들

회의 안건 :

- 전날 스크럼 회의에서 세운 목표사항 완료 여부 체크
- 새로운 일일 목표 공유
- 문제 상황 공유



주간 스프린트 회의

시간 : 금요일 저녁 8시

장소 : 슬랙 허들

회의 안건 :

- 스프린트 진행 상황 점검
- 마일스톤 환기
- 문제 상황 공유

팀빌딩

주기적으로 팀의 목표와 개인의 목표를 되돌아보는 시간 가지기

협업을 위해 필요할 경우 회의를 주최하고 되도록 참석하기

개발 상황에 대해 솔직하게 이야기하고 서로 신뢰하기



그라운드룰

회의-슬랙, 노션-자료공유

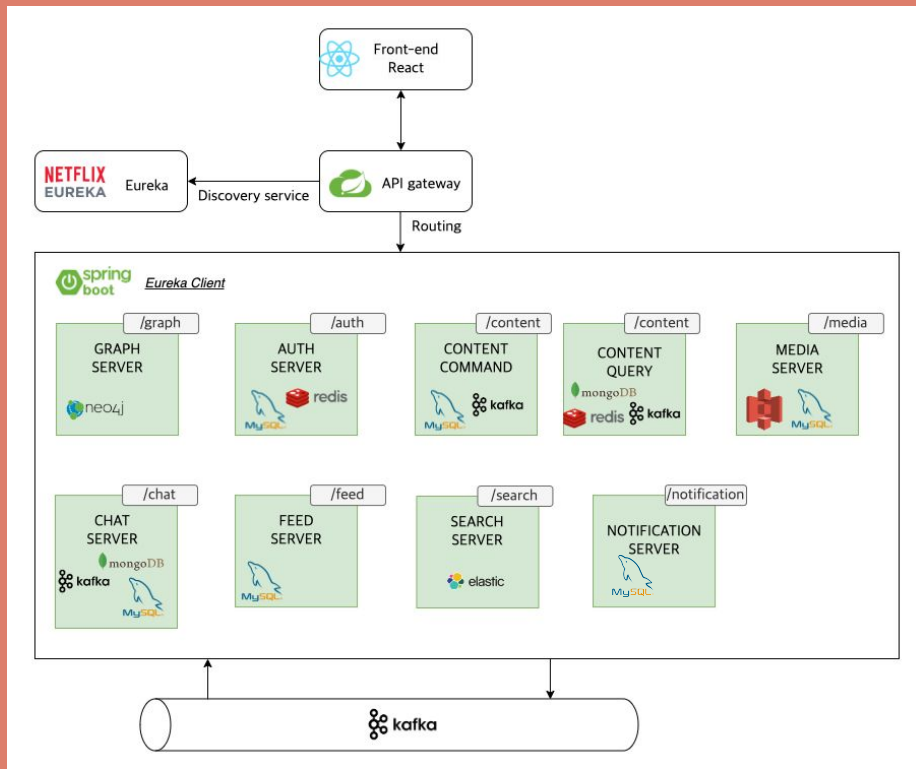
만장일치로 의사 결정하기

네이버 자바 코딩 컨벤션 활용

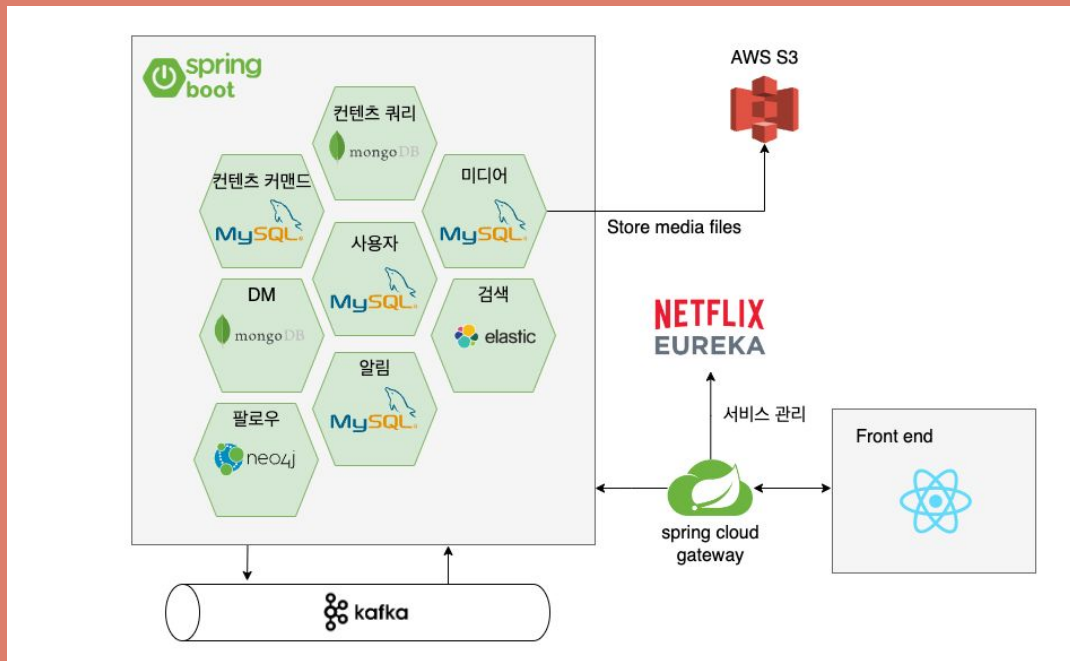
결석 많이 하지 않기

리뷰시 서로 존중하기

기존 설계




최종 구현




로그인 및 회원 가입 - 사용자 서버

로그인 기능



[비밀번호를 잊으셨나요?](#)

[계정이 없으신가요? 가입하기](#)




로그인에 문제가 있나요?

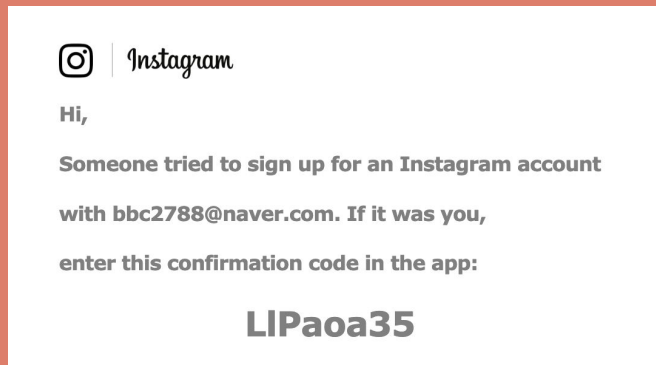
이메일 주소, 전화번호, 사용자 이름을 입력하시면 가입하신 이메일에 임시 비밀번호를 보내드립니다.

로그인 및 회원 가입 - 사용자 서버

메일 인증을 통한 회원 가입



친구들의 사진과 동영상을 보려면
가입하세요.



accessToken	eyJhbGciOiJIUzI1NiJ9.eyJzdWiiOiJpYmMyNzg4QG5hdmVyL...
refreshToken	eyJhbGciOiJIUzI1NiJ9.eyJzdWiiOiJpYmMyNzg4QG5hdmVyL...

피드, 포스트 - 콘텐츠 쿼리 서버

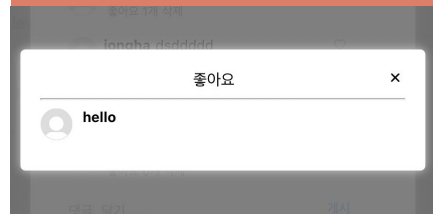
피드 (infinite scroll view)- 포스트 slice
조회



댓글 조회



댓글 좋아요 리스트 조회



피드, 포스트 - 콘텐츠 커맨드 서버

포스트 생성, 이미지 업로드

파일 선택

선택된 파일 없음

media upload

본문 입력...

댓글 활성화 ☐

좋아요 숨기기 ☐

추가

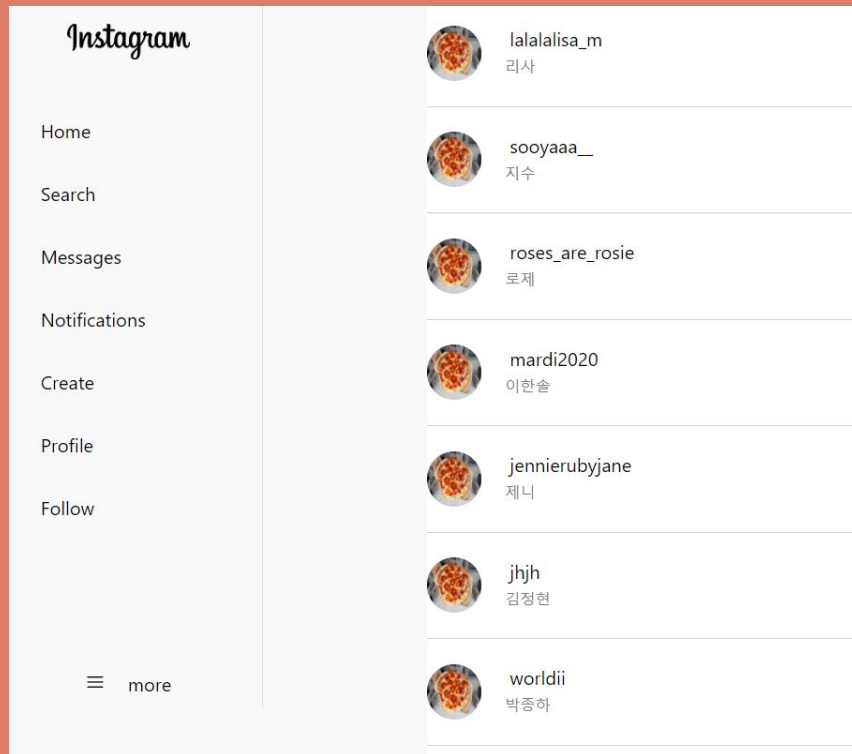
포스트 삭제, 수정, 좋아요,
북마크



댓글 생성, 좋아요, 삭제




팔로우 서비스



디엠 서비스

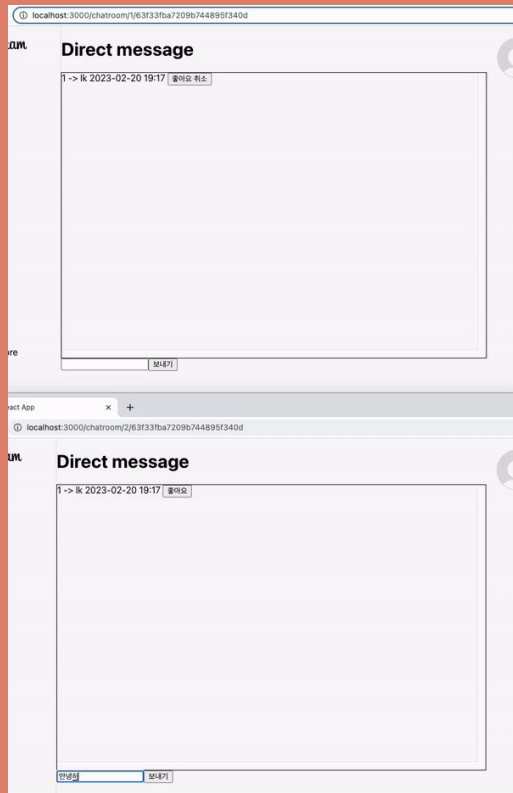
채팅 목록

 안녕하세요 | Ik | 2023-02-20 19:17 들어가기

채팅방 생성

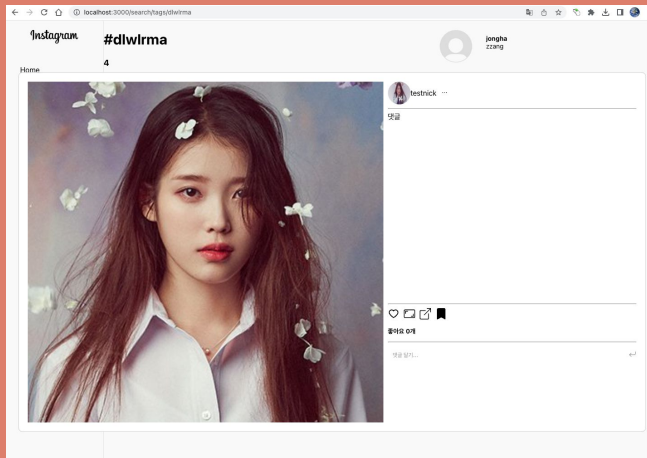
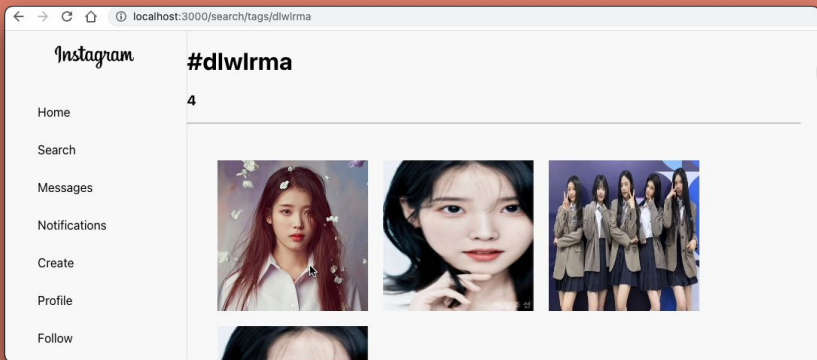
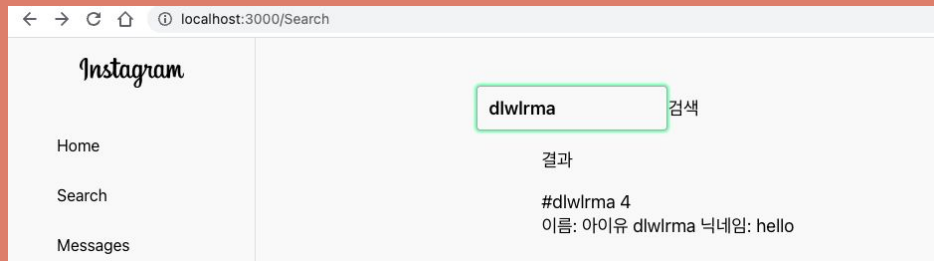
DM 목록

DM 전송

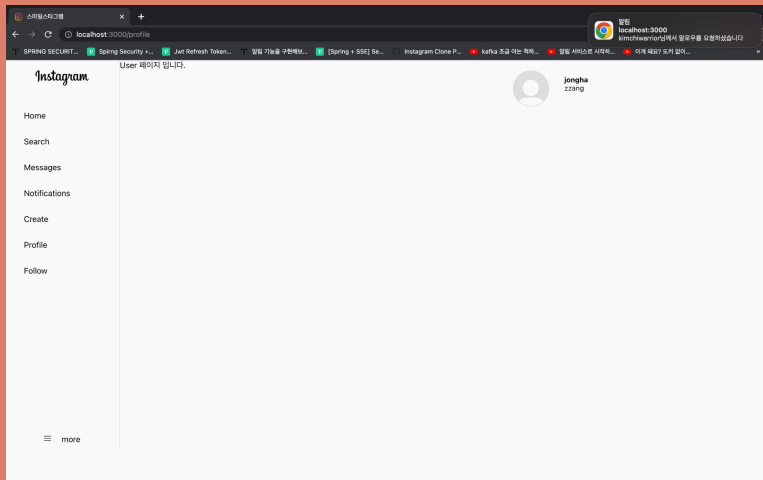
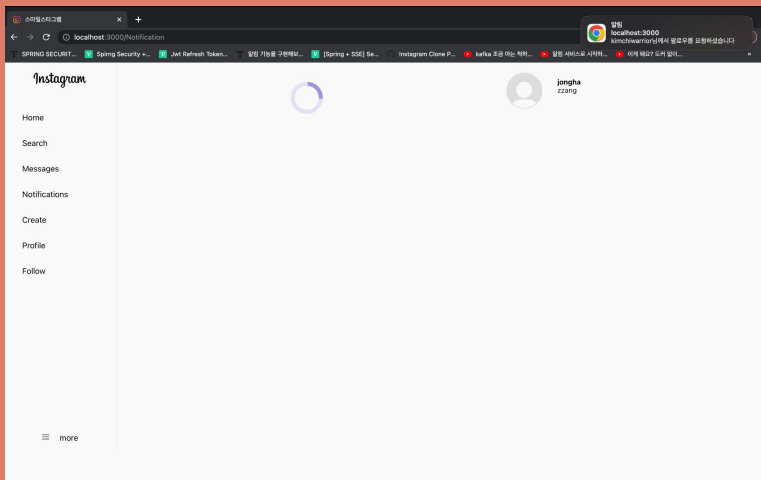


<https://drive.google.com/file/d/1tUIViOiyQyn7njingTRxzSrCwMCAIFPb/view>

검색 서비스

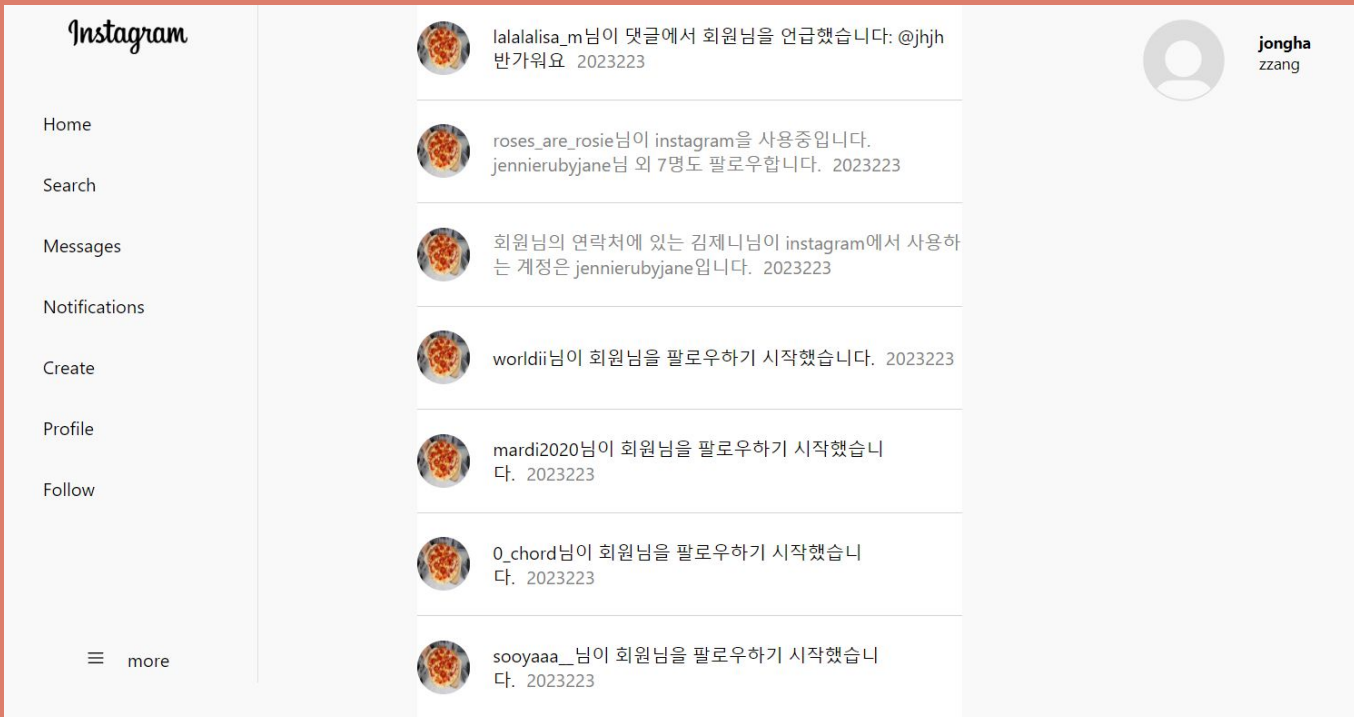


dlwlrma으로 검색 및 결과, 상세
페이지



Server-Sent-Event를 통한 실시간 알림 서비스

알림 서비스



박종하

개인 목표

- 스프링을 활용한 백엔드 구현 능력 향상, 백엔드 개발자로서의 도약.
- 개발 이슈 및 해결 과정 문서화
- MSA 아키텍처를 이해하고 구현

박종하

시행 착오 및 어려움

- 백엔드 개발에 미숙
 - DTO 에 불필요한 요소를 넣거나 msa 방식으로 설계할 것을 모놀리식한 방식으로 설계. 감이 잘 잡히지 않음.
 - 리팩토링으로 많은 불필요한 부분 수정
- 기능 개발에 치중
 - 미뤘던 테스트 코드 작성을 하지 못함. 앞으로 개발을 할때 작은 기능을 구현한 후 TDD 방식을 사용 결심
 - 필요한 부분만 최소한의 기능을 갖게끔 구현해야함을 깨달음
- 개발 투자 시간이 부족하여 기능 개발을 완성도 있게 못했던 것에 대한 아쉬움
- 백엔드 기초 부터가 아닌, 구현을 위해 찾아보는 방식으로 공부를 하여서, 깊게 공부하지 못했다는 아쉬움

박종하

회고 및 성장

- MSA 아키텍처 이해
아키텍처에 대해
이해하고, 구현하는
경험
- 협업을 통한 성장
팀원들의 좋은 코드를
보며, CQRS 패턴,
카프카, 스프링
클라우드 등 새로운
개념들을 접함.

-> 백엔드 개발자로서
앞으로의 방향성 정립.
- 백엔드 프론트엔드 둘다
경험

구현한 API 를 react를
통해 호출하여 동작
확인

-> API 호출할 때
underfetching,
overfetching을 고려하게
됨.
API 명세화 필요성 느낌

김정현

개인목표

- 개발자의 공부 방법, 문제 해결 방식을 경험해 나만의 문제 해결 루틴을 찾고 레퍼런스 프로젝트로 활용할 수 있도록 만들기
- 버전 관리 및 문서 관리를 체계적으로 경험하고 팀을 위해 기록하는 습관 만들기

김정현

성공한 부분

- 레퍼런스 프로젝트로 활용할 수 있도록 만들기

실패한 부분

- 나만의 문제 해결 루틴 찾기

애매한 부분

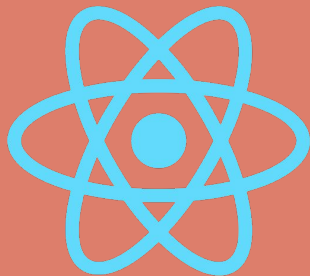
- 버전 관리 및 문서 관리를 체계적으로 경험
- 팀을 위해 기록하는 습관 만들기

김정현

고민했던 것들

1. 학습에 대한 고민 - 맨 땅에 헤딩 vs 충분한 학습
2. 협업에 대한 고민 - 예측 불가능한 기능 개발 소요 기간, 연동을 가정한 개발의 어려움
3. 내가 구현한 방법이 좋은 방법인지에 대한 의구심

김정현



맨 땅에 헤딩



충분한 학습

김정현

```
{
  "-userId": 1,
  "-text": "짧은 내용",
  "-imageUrl": "-https://img.youtube.com/vi/N7iIVIV-ZCs/mqdefault.jpg"
}
```

예측

```
{
  "content": [
    {
      "id": "63f4e5433ec4976d65374af1",
      "contentId": 29,
      "userId": 1,
      "text": "짧은 내용",
      "likes": 0,
      "imageUrl": [
        {
          "url": "naver.com",
          "order": 1,
          "user_id": null
        }
      ],
      "hashtags": [],
      "visibleLikes": false,
      "visibleComments": false,
      "createdAt": "2023-02-22 00:37",
      "modifiedAt": "2023-02-22 00:37"
    }
  ],
}
```

실제

김정현

1. Entity와 DTO 간 변환하는 메소드를 만들 때, 서비스나 컨트롤러에 선언하는 것과 DTO 클래스에 선언하는 것 중 어느쪽이 좋은 선택인지 궁금합니다.

DTO 클래스에 만드는 것이 여러 서비스와 컨트롤러에서 사용이 가능해서 좋아보이는데, DTO 객체를 생성할 때마다 메소드를 함께 생성하는 게 리소스 낭비인 것 같다는 생각도 듭니다.

```
@Service
public class RelationshipService {
    final RelationshipRepository relationshipRepository;

    private User userDtoToUser(UserDto userDto) {}
    private UserDto userToUserDto(User user) {}
}
```

```
public class NotificationDto {

    @Id @GeneratedValue
    private Long id;
    private String content;

    public Notification toEntity() {}
}
```

내가 잘 하고 있는 것인가

김정현

캠프를 통해 느낀 점

1. 협업이 곧 나의 성장이다.
2. 성공했을 때의 기쁨을 오래 가져가야 한다.
3. 도움을 청하는 것도 능력이다.
4. 성장은 결국 내가 하는 것이다.

이한솔

개인 목표

- MSA를 구현하고 통신하는 방법 이해
- 개발일지 작성 기록하는 습관 형성

결과


- 왜 필요한지 그 이유에 대한 충분한 고민을 하게 됨
- 이슈나 고민있을 때마다 기록으로 생각 정리

이한솔

Content 서버 구조에 대한 고민

- CQRS 패턴 도입
 - “Eventual consistency”
 - 변경과 조회의 분리
 - 지속 성장 가능한 구조

테스트 코드의 중요성

- API 인수 테스트, 기능 테스트 작성
- 번거로움 , 예외 케이스 및 검증 가능

불필요한 로직을 포함한 Transaction 개선

- 데이터를 변경할 때
- DB 커넥션을 사용하지 않는 시간을 줄여 병목 현상을 줄임

배운점

- 일정 산정, 업무 배정 및 시간 분배 중요성
 - front end 개발 팀원의 부재
 - 구현해야할 기능이 많아 많은 기능에 대해 깊은 고민을 하기 어려움
- dev 브랜치의 중요성
- 직접 front 개발을 해보니 시나리오, API 구성에 대해 느낄 수 있었음

김영현

개인목표

1. git 버전관리를 학습하여 팀 프로젝트에서 git 관리 실패하지 않기
2. MSA 직접 설계와 구현을 하며 MSA의 철학 이해하기
3. 기능 명세와 개발 과정, 문제해결 내용을 정리하여 명세서 작성하기
4. 프론트엔드를 React로 설계하여 백엔드와 프론트엔드의 협업 방식 이해하기

김영현

개인목표

1. git 버전관리를 학습하여 팀 프로젝트에서 git 관리 실패하지 않기



김영현

개인목표

2. MSA 직접 설계와 구현을 하며 MSA의 철학 이해하기



김영현

개인목표

3. 기능 명세와 개발 과정, 문제해결 내용을 정리하여 명세서 작성하기



김영현

개인목표

4. 프론트엔드를 React로 설계하여 백엔드와 프론트엔드의 협업 방식 이해



김영현

시행착오

영현

JIKJI 김영현 오후 9:19

안녕하세요 😊 늦은시간에 죄송하지만 eventsource에 관련하여 질문드립니다.
현재 sse를 연결하기 위해 프론트단에 eventsource API를 사용하고 있습니다.

useEffect를 통해 eventsource를 등록하다보니 다른 웹 페이지에서는 eventSource를 사용하지 못하는 문제가 발생하고 있습니다.

저는 로그인 시 eventSource를 등록하고 그 eventSource를 모든 경로에서 사용하고 싶은데 어떻게 해야 하나요?
모든 리액트 파일에서 eventSource를 생성하고 /sub를 호출하여 이벤트 등록을 해야 하나요?

image.png ▼

```
function Test() {  
  const location = useLocation();  
  let eventSource;  
  useEffect(() => {  
    let subscribeUrl = "http://localhost:8080/sub";  
    const permission = Notification.requestPermission();  
    const accessToken = localStorage.getItem("accessToken");  
    eventSource = new EventSource(subscribeUrl + "?token=" + accessToken);  
  }, [location]);  
}
```



영현

3개의 댓글 11일 전 마지막 댓글



김영현

시행착오



회고

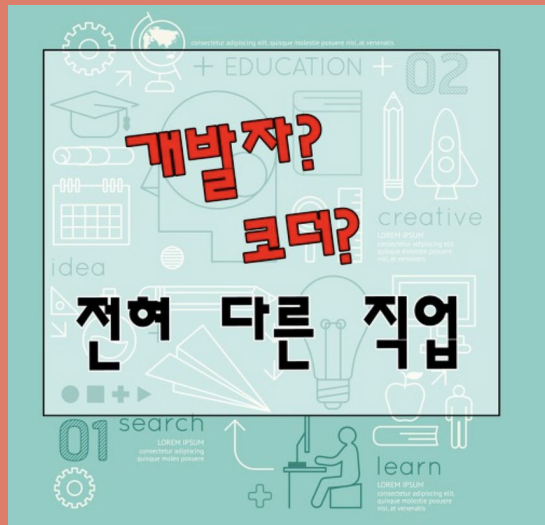
김영현

성장



김영현

성장



내 학습 >



스프링 핵심 원리 - 기본편

Spring Core Basic
OOP, Spring Container & Spring Bean

기한: 무제한 | 진도율: 100.00%

바로 학습 ▶



자바 ORM 표준 JPA 프로그래밍 - 기본편

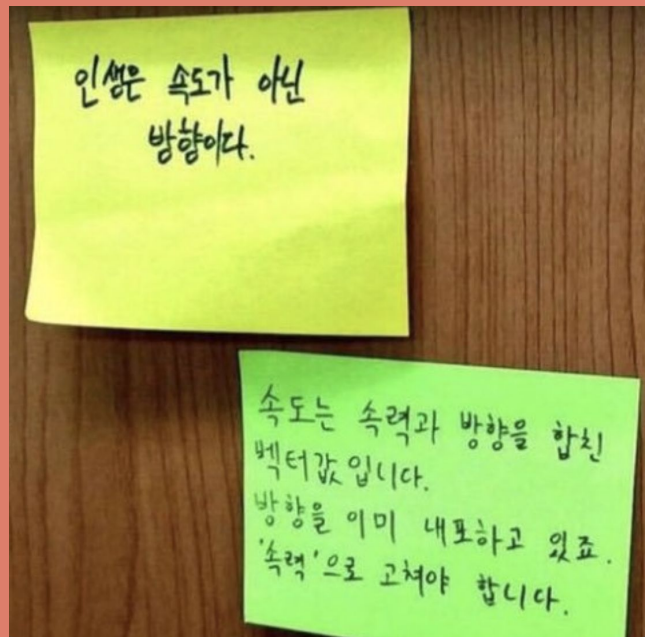
JPA
Java Persistence API

기한: 무제한 | 진도율: 100.00%

바로 학습 ▶

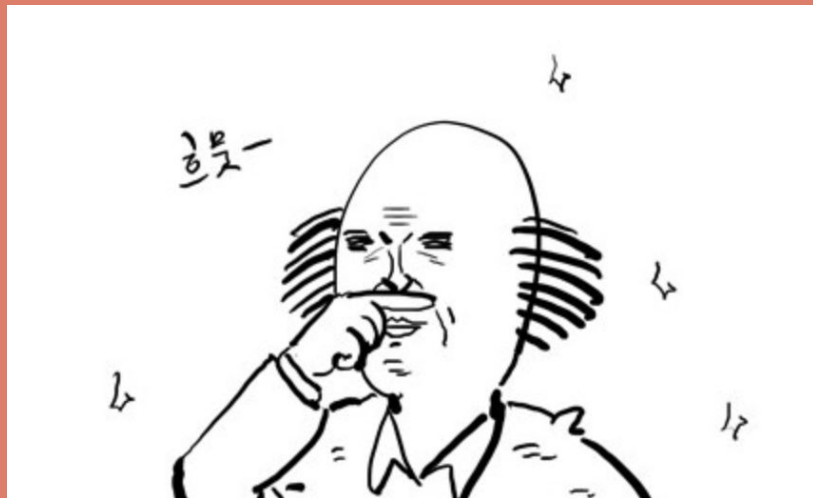
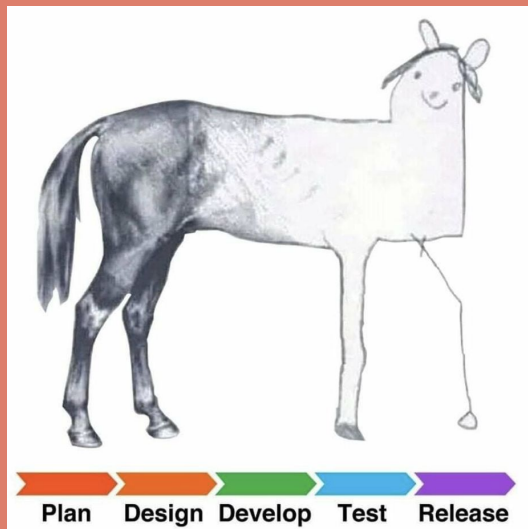
김영현

성장



김영현

프로젝트 회고



김영현

프로젝트 회고



QnA