

# 코드리뷰를 위한 README

☰ 태그	코드리뷰
📅 날짜	@2023년 2월 10일

## instagram-clone project

### 📁 develop 브랜치 기준 폴더 구조

|— discovery.  
|— docs  
|— feed-service  
|— follow-service  
|— gateway.  
|— membership-service  
|— post-service  
|— front (feature/front/all 브랜치에서 develop 브랜치로 추가 예정)  
|— chatting-service (test 브랜치에서 develop 브랜치로 추가 예정)  
|— search-service (개발 x)  
|— notification-service (개발 x)

README는 develop 브랜치에 있습니다.

### 🐱 코드 리뷰를 받을 브랜치

1. BE
  - a. feature/back/gateway
  - b. feature/back/feed-service
  - c. feature/back/membership-service
  - d. feature/back/post-service
  - e. feature/back/follow-service
  - f. test (chatting-service가 개발된 브랜치)
2. FE
  - a. feature/front/all

## 팀 공통 고민

1. 최근 팀원 한 분이 개인 사정으로 나갔습니다.
  - a. 채팅 서비스만 개발을 완료해주실 수 있는지 제안을 했지만, 안 된다고 하셔서
  - b. 채팅, 검색, 알림을 이제 저희 3명이 개발해야 합니다.
    - i. 채팅은 어느 정도 개발이 진행되어 있는 상태입니다. (파이썬, Django로 구현)
    - ii. 검색, 알림은 개발 진행 상황이 없습니다.
    - iii. 두 가지 방안
      1. 검색, 채팅, 알림을 한 명씩 맡고, 해결 못하는 분이 있으면 다 같이 돕기
      2. 현재 각자 맡고 있는 서비스들에 최대한 집중 후 채팅, 검색, 알림을 구현할 수 있다면 시도하기

- a. 이렇게 한다면 채팅, 검색, 알림 중에서 빼야 하는 기능이 생길 것 같습니다.
  - i. 채팅은 어느 정도 개발이 진행되어 있으니 3명이 같이 만들고, 검색과 알림은 포기
  - ii. 아니면, 채팅/알림/검색을 다 빼기 😊

## 개인별 작성

### ▼ 강민정

#### 1. Post-service 모델링

- a. `Post` 게시물
- b. `PostLike` 게시물 좋아요
- c. `User` 계정
- d. `Hashtag` 해시태그
- e. `CommentLike` 댓글 좋아요
- f. `Comment` 댓글

### 🤔 고민한 점

1. post 생성 및 삭제 시 hashtag를 반복문으로 생성/삭제합니다. ➡ 성능을 향상시키려면, hashtag를 도큐먼트에 생성/삭제하는 작업은 비동기로 해야할까요? ➡ 한 포스트에 적는 hashtag 개수를 제한하는 게 좋지 않을까 생각도 해봤습니다.

- a. [현재 사용하는 방법] 검색 서비스를 위해서 해시 태그를 document로 분리
  - i. [첫 번째 시도] Post에 Hashtag를 embed ➡ 검색 서비스에서 Hashtag로 검색할 때 불편합니다.
  - ii. [두 번째 시도] Hashtag와 Post는 상호 참조 ➡ 서로 식별자 id를 가져야 하는데 어떻게 저장할지부터 복잡했습니다.

#### 2. 이미지 업로드

- 이미지 업로드 기능은 MSA에서 하나의 서비스로 다루는 경우도 봤습니다.
  - `membership-service`, `post-service` 에서 이미지 업로드 기능을 사용하는데 지금처럼 분리 안 하고 진행할까요?
  - 현재 로컬에 저장하도록 구현했는데 좋은 방법이 아닌 것을 알지만, AWS를 사용해야하나 고민중입니다.
  - 이미지를 업로드를 비동기로 구현하는 게 일반적인 방법이나요?
    - 유저가 저장 버튼을 누르기 전까지 N번 이미지를 바꿀텐데 N-1번 이미지는 서버에서 삭제되어야 할 것 같습니다.
- 어떻게 자동으로 삭제할지 모르겠습니다.

#### 1. 트랜잭션

- a. `membership-service` 의 signup 메소드에서 Kafka 이벤트 생성 때문에 DB에 user가 들어오지 못하는 경우가 있었습니다.
  - i. 하지만, 비동기 작업은 끝나지 않아도 응답은 성공으로 옵니다.
    1. 비동기 작업이 있으면 트랜잭션 적용을 안 할까요?
- b. 트랜잭션을 어디에 적용할지 잘 모르고 대부분의 기능에 사용하는 경향이 있는 것 같습니다.

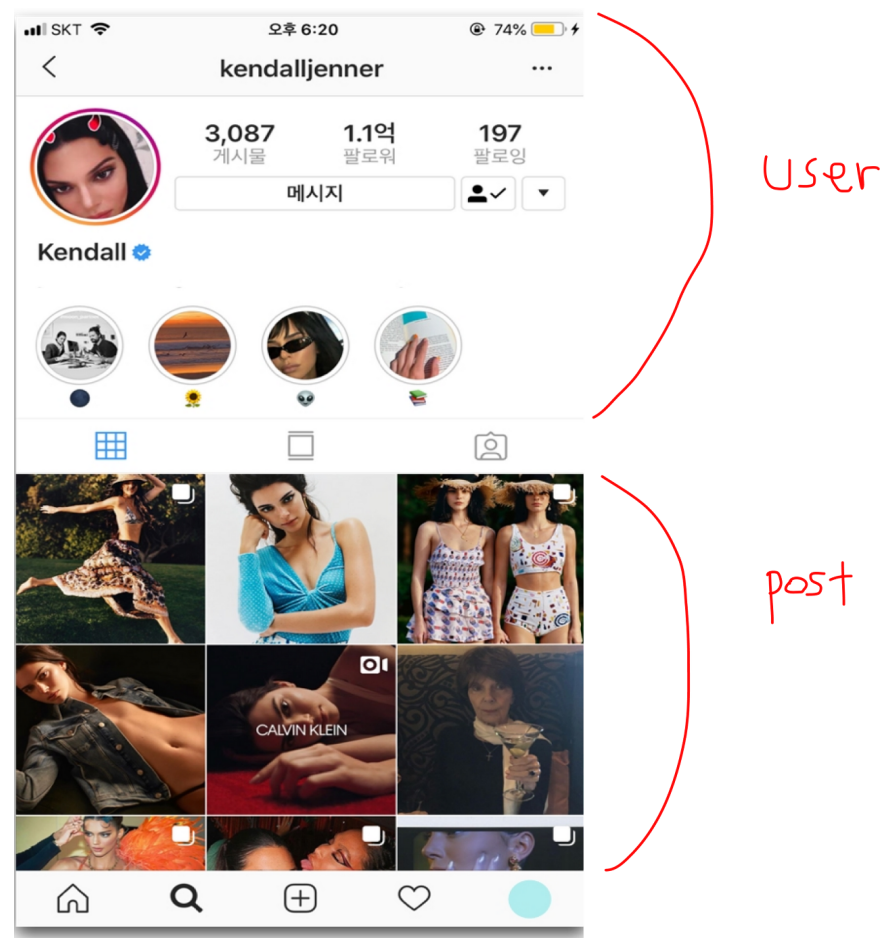
#### 4. 피드

- a. **pull** 방식으로 구현했습니다.
  - i. FeignClient 사용
    1. `POST /membership-server/login` AccessToken 받아오기
    2. `POST /post-server/posts/in` 포스트를 가져오기
    3. `POST /membership-server/auth/summary/in` 유저 정보(프로필) 가져오기
    4. `GET /follow-server/users/paginated/{username}/followers` 유저의 팔로워 목록 가져오기
  - ii. 페이징에 대해 좀더 이해가 필요할 것 같아요.

- b. 원래는 **Push** 방식을 사용하는 것을 계획으로 세웠습니다.
  - i. SSE로 피드를 프론트에 보내주는 방법을 생각하고 있습니다.
  - ii. 중간 발표 때 Push 방식을 사용하면 넘치는 데이터는 어떻게 처리할 것인지, Pull로 당겨올 것이냐는 질문을 받았는데 어떻게 해결해야 하는건가요?
- c. 테스트를 안 해봐서 지금 작성한 코드가 잘 동작할지는 모르겠어요..

## 5. 개인 피드 조회

- a. `user` 도큐먼트를 조회 후 `post` 도큐먼트를 조회합니다.



- post를 조회할 때 한번에 모든 post를 프론트엔드에 다 주고, 프론트엔드에서 보여주나요?
- 아니면, 스크롤 내릴 때마다 페이징네이션을 써서 가져오나요?

## 6. 테스트 코드 작성

- a. WebMvcTest를 작성하려고 했었습니다.
  - i. service 계층을 MockBean으로 주입하면 가짜 객체니까 return이 null이네요.
  - ii. WebMvcTest는 해당 API 요청이 잘 되는지 가볍게 확인하는 용도로만 쓰는건가요?
  - iii. DB까지 데이터를 넣으면서 테스트 코드를 작성하면, 테스트가 좀 무거워지도 하고 테스트를 작성하기 어려운 부분도 가끔 있어서 고민입니다.

## 7. Security

- a. 이메일 인증을 받은 사용자는 `ROLE_USER`, 받지 않은 사용자는 `ROLE_NOT_PERMITTED`입니다.
  - i. `ROLE_NOT_PERMITTED` 사용자는 대부분의 서비스를 사용할 수 없으니까 모든 서비스마다 권한 설정을 해줘야 할까요?
    - 1. 현재는 인증 받은 사용자이지만, 체크하고 있어요.
- b. Gateway의 필터에서 JWT 토큰을 검증하는데 그 뒤에 있는 서비스 단에서도 인증을 확인해야 하나요?
  - i. `anyRequest.permitAll()` 해도 되는지?
  - ii. membership-service에서는 `antMatchers().permitAll()`에 허용할 URL을 명시하고, 나머지는 `anyRequest.authenticated()` 를 사용해 인증을 받도록 했습니다.

## 8. 성능 테스트

- a. 성능 테스트는 어느 부분을 하나요?(로그인, 회원 가입 같은 부분은 안 해도 될 것 같은데)
- b. 지금 상태로 성능 테스트까지 해볼 수 있을까요?

9. membership-service ERD

