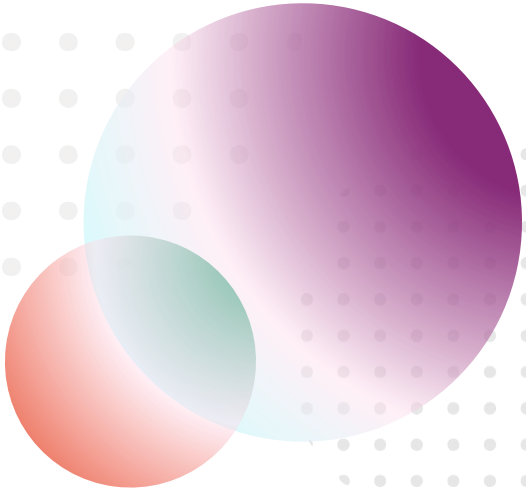


지역 광주

팀명 codeRunner

이름 김현지, 박가인, 장소명, 정동환

Smilegate®

CONTENTS

codeRunner

3개월동안 넘어지지 말고 끝까지 열심히 달려보자는 의미

FE 김현지

FE/BE 박가인

BE 장소명

FE/BE 정동환

팀 목표

팀 프로젝트를 통해서 달성하고 싶은 모두의 목표

1. 포트폴리오 정리를 하러 들어왔을 때에도 이해가 가능한 코드를 작성하자
2. 프로젝트 전체를 보는 시각을 기르기 위해, 서로의 분야를 먼저 이해하자
3. 개발과정을 문서화해 두자
4. 사용기술(socket통신, MSA 등)의 장단점을 파악하는 기회를 가지자

주제

CRUD 중심 서비스 뿐만 아니라
Socket 통신, MSA구조 등과 같은
다양한 경험이 가능한 플랫폼 인가?

⇩

복잡한 구조를 가진 플랫폼 인가?

⇩

팀 원 모두에게
익숙한 플랫폼 인가?

⇩

WorkFlow

Core time

월 ~ 금요일, 9:00 - 12:00

ZEP 회의실 1에 모여 본인이 맡은 부분을 집중해서 개발하는 시간

Weekly Meeting

월요일, 20:00 ~

ice breaking을 시작으로 회의 안건에 대해서 논의

Daily Scrum

월 ~ 금요일, 9:00 - 9:15

ZEP 에서 오늘 개발할 것, 어제 개발한 것, 이슈 혹은 장애 공유

WorkFlow

Ground Rule

1. 부득이한 사정으로 회의를 미뤄야 할 경우 사전에 미리 알려주기.
2. 업무와 관련된 연락은 최대한 빨리 답해주기.
3. 노션 또는 코드를 수정할 경우 어떠한 사유로 어떻게 수정했는지 알려주기
4. 각 분야의 고충을 토로할 시 명확한 근거를 바탕으로 communication하기

(ex.그렇게하면 안돼요. 그럼 저희 복잡해져요. 가 아닌 ~~해서 ~~한 이유로 다른 방법이 필요할 것 같아요!)

Coding Convention - JavaScript

Airbnb JavaScript Style Guide

- 연산자(= + - * /) 와 쉼표 뒤에 공백을 두기
- 코드 들여쓰기 : Tab 키 대신 spacebar키 2번으로 통일
- 파일 이름은 소문자로 통일
- css 속성 이름(글꼴 크기)에 하이픈을 사용
- 가독성을 위해서 한 라인은 80자를 넘지 않도록 하기(한줄에 다 들어가지 않을 때는 연산자 또는 쉼표 로 구분하기

Coding Convention - JavaScript

〈 객체 규칙 〉

- 객체 이름과 같은 줄에 여는 괄호 배치
- 속성과 값 사이에 콜론과 하나의 공백 두기
- 객체 정의는 세미콜론으로 끝내기
- 닫는 괄호 앞에는 공백 두지 않기
- 마지막 key-value 뒤에 쉼표 두지 않기

Coding Convention - Java

〈 변수, 함수 클래스 명명 규칙 〉

- 변수는 명사형으로 작성
- 함수는 서술형으로 작성
- 클래스 명은 파스칼 케이스를 사용
- 기본적으로 카멜 케이스을 사용 (camelCase)
- 상수는 영문 대문자 스네이크 표기법(Snake case)를 사용
- 한 줄 당 한 선언문만 사용
- 메서드들을 구분하기 위해서 각 메서드들 사이에는 한 줄을 비우기

Coding Convention - Java

〈 객체지향 생활 체조 9가지 원칙 〉

- 규칙 1: 한 메서드에 오직 한 단계의 들여쓰기(indent)만 한다.
- 규칙 2: else 예약어를 쓰지 않는다.
- 규칙 3: 모든 원시값과 문자열을 포장한다.
- 규칙 4: 한 줄에 점을 하나만 찍는다.
- 규칙 5: 줄여쓰지 않는다(축약 금지).
- 규칙 6: 모든 엔티티를 작게 유지한다.
- 규칙 7: 3개 이상의 인스턴스 변수를 가진 클래스를 쓰지 않는다.
- 규칙 8: 일급 콜렉션을 쓴다.
- 규칙 9: 게터/세터/프로퍼티를 쓰지 않는다.

Commit Convention

〈 structure 〉

type : subject

body

footer (option)

〈 type 〉

- feat : 새로운 기능 추가
- fix : 버그 수정
- docs : 문서 수정
- style : 코드 포맷팅, 세미콜론 누락, 코드 변경이 없는 경우
- refactor : 코드 리팩토링
- test : 테스트 코드, 리팩토링 테스트 코드 추가 등
- chore : 빌드 업무 수정, 패키지 매니저 수정

개인 목표 : 김현지

현재 상태

1. 상태관리를 redux로만 구현해낼 수 있음.
2. creat-react-app 을 사용하지 않지 않으면 react를 사용할 수 없음.
3. git-hub에서 commit과 pull request만 경험해봄.

목표

1. redux, recoil, swr의 차이점을 파악하고, 이번에 사용할 기술인 recoil 공식문서 읽기.
2. creat-react-app 을 사용하지 않고 직접 프로젝트 환경 설정해보기.
3. branch 전략을 사용해서 기능별로 issue에 등록한 후 branch를 나누고 pull request를 날려 merge해보기.

개인 목표 : 박가인

현재 상태

1. router 파일에 쿼리문과 비지니스 로직을 다 넣어서 코드 가독성이 떨어짐
2. recoil 의 selector 함수의 용도를 파악하지 못함
3. api 명세서를 작성해본 경험이 없음

목표

1. 채널, 게시글 rest api 작성할 때 코드 파악이 쉽도록 controller 부분을 따로 만들고 db쿼리문은 config파일에 작성하기
2. 리액트에서 비공개 채널과 공개 채널 게시글을 분리하여 상태관리하는 로직은 jest로 테스트 코드 작성하여 code coverage 70% 달성하기
3. recoil 라이브러리의 selector 함수는 어떤 역할을 하는 함수인지 2줄로 설명할 수 있게 되기
4. 채널, 게시글의 api 명세서를 swagger를 이용하여 작성하기

개인 목표 : 장소명

현재 상태

1. MA 구조의 프로젝트만 해왔음
2. MySQL과 같은 관계형 DB만 사용
3. Socket 통신 구현 경험 없음

목표

1. MSA 구조 설계를 통해 기존 사용했던 MA 구조와 차이점 조사 및 기록을 통해 이해
2. NoSQL과 SQL 차이점 조사 및 기록을 통해 이해
3. Redis의 특징, 흐름 조사 및 기록을 통해 이해
4. Socket 통신의 특징, 흐름 조사 및 기록을 통해 이해

=> 모두 팀원에게 설명할 수 있는지 확인

개인 목표 : 정동환

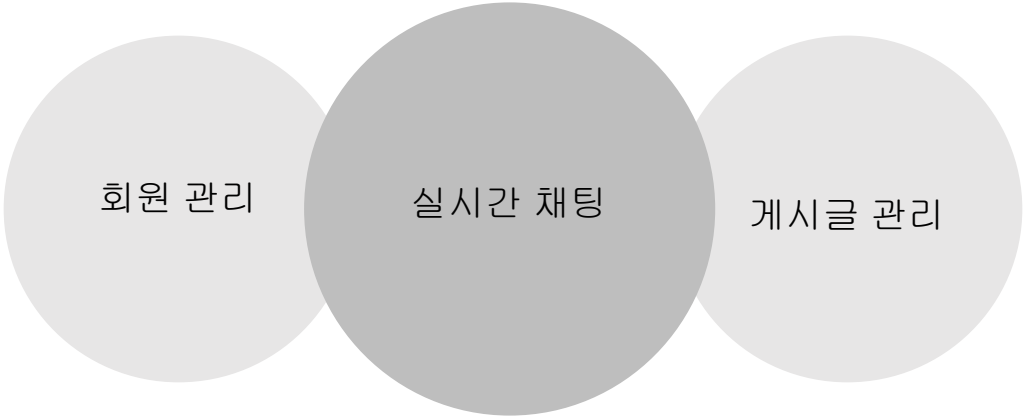
현재 상태

1. 두 명 이상이서 협업개발을 해본 적이 없음
2. 리액트를 자바스크립트 페이지와 큰 차이 없이 사용함
3. 개발에 필요한 문서를 만들어 본 경험이 부족함

목표

1. 1주일에 3회 이상 배운점 문서화하기
   - 팀 목표인 문서화와와도 직결됨
   - 그동안 학습한 것을 일회용으로 넘기지 않기 위함
2. git으로 하는 협업 코딩에서 중요한 것들에 대해 알아보기
   - 다른 사람이 알아보기 쉬운 코드
   - 팀원들과 합의한 방식으로 변경점 등 업로드하기
3. react와 recoil 에 대해 익히기
   - 처음 써보는 기술인만큼 향후 스펙으로 활용할 수 있도록 공부하기
4. vscode 자체 기능 및 확장프로그램에 대해 알아보기
   - 개발 시간 단축에 도움이 되는 여러 기법 익히기

주요 기능



기술 stack

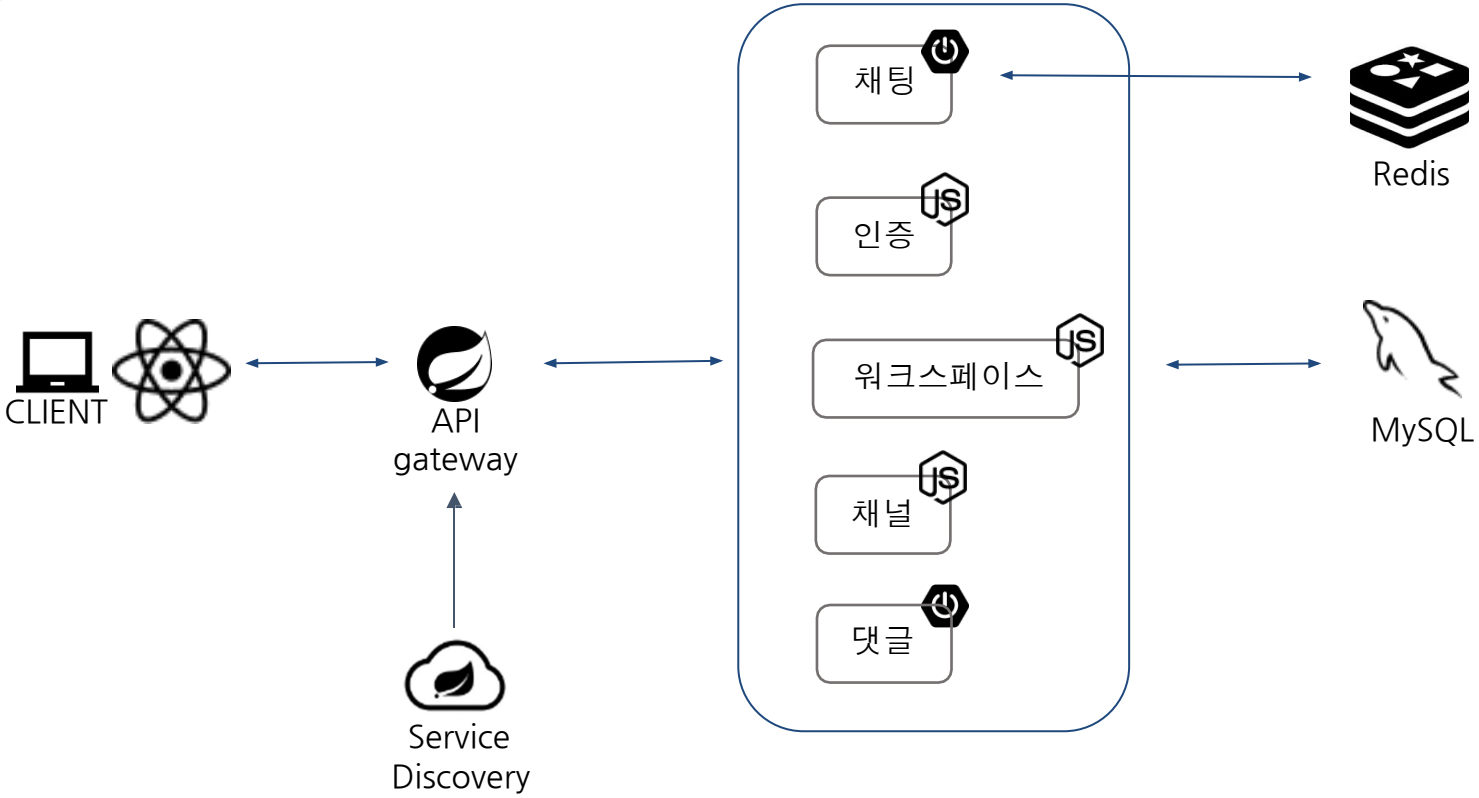
Front - End

- React
- React icons
- React Router
- Recoil
- Node.js
- Styled-components
- SockJS

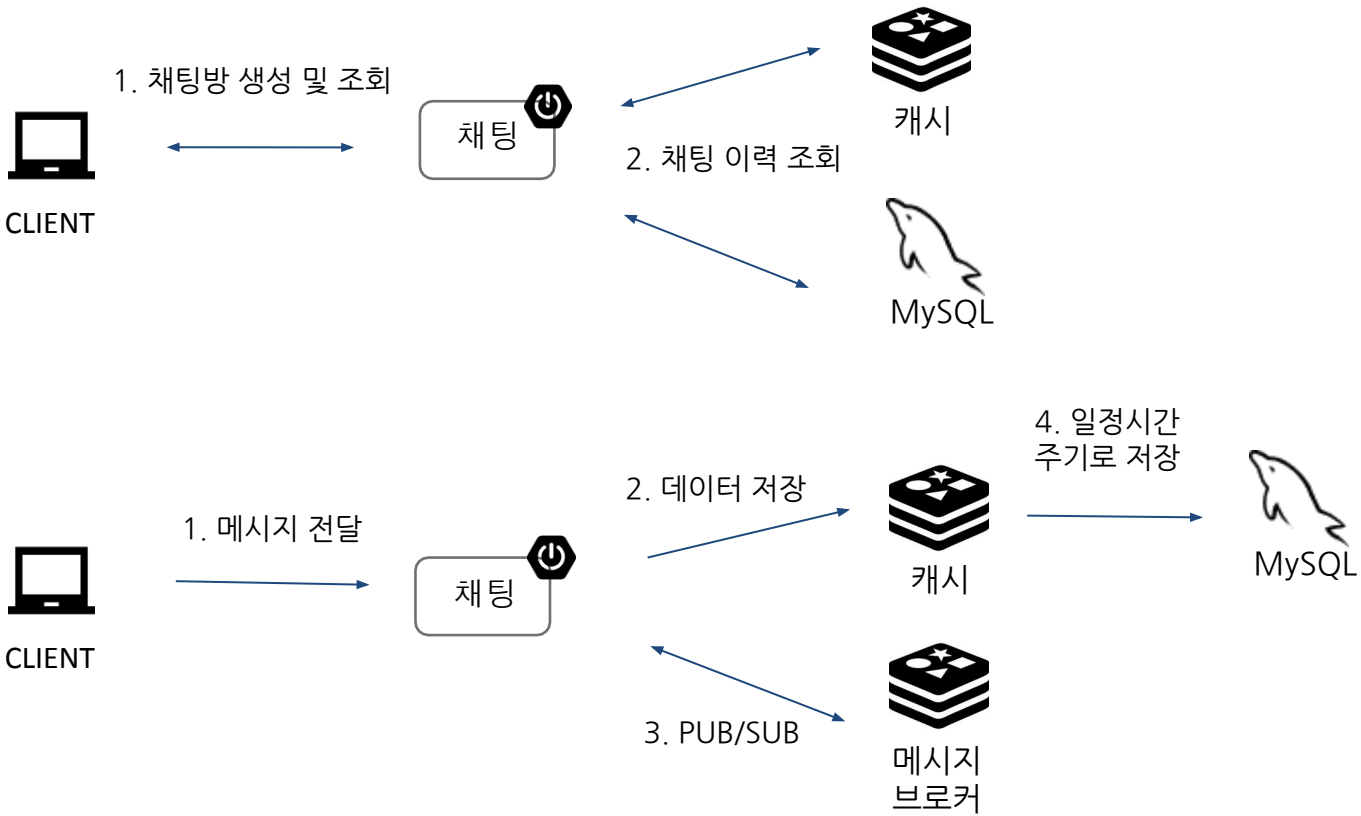
Back - End

- Spring Cloud
- My SQL
- Spring Boot
- Spring Data JPA
- Socket
- Redis

Architecture



주요 Server Architecture - 채팅



역할 소개 및 개발 계획

김현지

FE - 워크스페이스, 댓글

1월 4주차 : 개념 보충 및 개발 환경 설정
- recoil 예제 연습
- socket 통신 과정 공부
- prettier, eslint 등 환경 설정

2월 1주차 : 워크스페이스 기능 구현

2월 2주차 : 댓글 기능 구현
- 스레드 창 구현
- 댓글 crud 기능 구현

2월 3주차 : 성능테스트 및 개선
- 성능 개선을 위한 상태관리 및 컴포넌트 코드 부분 리팩토링

박가인

FE - 채널, 채팅 BE - 채널

1월 4주차 : 채널 서버 rest API 개발
-API 명세서 기반으로 rest API 개발

2월 1주차 : 채널 프론트부분 React로 개발
-sockJS 오픈소스 예시코드 읽기

2월 2주차 :채팅 프론트 부분 개발
-sockJS로 채팅관련 로직 작성
-채팅 UI 만들기

2월 3주차 : 성능테스트 및 개선
-코드 리팩토링

역할 소개 및 개발 계획

장소명

BE - API 게이트웨이, 채팅

1월 4주차 : MSA 통신 환경 구성
- MSA 예제 연습
- Eureka Server 구축, Client 등록
- API Gateway 구축

2월 1주차 ~ 2주차 : 채팅 서버 구축
- Redis를 사용한 채팅 예제 연습
- 채팅 관련 entity 구성
- Redis 연동 및 설정
- Websocket 서버 구축
- 채팅 service, controller 구현
- Websocket Handler 구축
- Websocket을 이용한 채팅 구현

2월 3주차 : 성능테스트 및 개선

정동환

FE - 회원 BE - 회원, 댓글

1월 4주차 : 인증 서버 구축
- 인증 api 명세서 작성
- 댓글 api 명세서 작성
- Node.js를 이용한 인증 BE 작성

2월 1주차 : 댓글 서버 구축
- Node.js를 이용한 댓글 BE 작성
- FE 사양과 상황에 맞는 지속적인 수정

2월 2주차 : 인증 서버 연동
- JWT와 메일링 서비스를 이용한 인증 서비스 구현

2월 3주차 : 성능테스트 및 개선

감사합니다.