

Smilegate®

지역 **광주**

팀명 **codeRunner**

이름 **김현지, 박가인, 장소명, 정동환**

CONTENTS



01. 팀 소개

02. 주제 설명

03. 팀 목표

04. 개인 목표

05. 기능 명세서

06. Architecture

07. DataBase

08. Ground Rules

09. Convention



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 이해해보자.
3. 개발과정을 문서화해 두자.
4. 사용기술(socket통신, MSA 등)의 장단점을 파악하는 기회를 가지자

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?





개인 목표

달성하고 싶은 개인 목표를 위한 학습 계획
김현지

1. redux, recoil, swr의 차이점을 파악하고, 이번에 사용할 기술인 recoil ? 익숙해지기??.
2. react에 대한 이해도 높이기
 - 내부 구조에 대해서 깊게 생각하며 코드 짜기
 - creat-react-app 을 사용하지 않고 직접 프로젝트 환경 설정해보기
3. git과 친해지기.
 - 기능별로 branch와 issue들을 활용해보기

개인 목표

달성하고 싶은 개인 목표를 위한 학습 계획

박가인

현재

1. router 파일에 쿼리문과 비즈니스 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 함수 인지 2줄로 설명할 수 있게 되기
4. 채널, 게시글의 api 명세서를 swagger를 이용하여 작성하기



개인 목표

달성하고 싶은 개인 목표를 위한 학습 계획

장소명

이전 나의 상태

- MA 구조의 프로젝트만 해왔음
- MySQL과 같은 관계형 DB만 사용
- Socket 통신 구현 경험 없음
- 이슈 해결 과정의 기록없이 진행함

이후 목표

- MSA 구조 설계를 통해 기존 사용했던 MA 구조와 차이점 조사 및 기록
- NoSQL과 SQL 차이점 조사 및 기록
- Redis의 특징, 흐름 조사 및 기록
- Socket 통신의 특징, 흐름 조사 및 기록
- 이슈 해결 과정 기록 후 다시 해당 이슈를 만났을 때 대처할 수 있는지 확인

개인 목표

달성하고 싶은 개인 목표를 위한 학습 계획

정동환

현재 나의 모습

- 두 명 이상에서 협업개발을 해본 적이 없음
- 리액트를 자바스크립트 페이지와 큰 차이 없이 사용함
- 개발에 필요한 문서를 만들어 본 경험이 부족함
-

목표

1. 1주일에 3회 이상 그동안 배웠고 깨달은 것들을 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. react는 물론 상태관리 라이브러리인 recoil 에 대해서 익히기
 - 처음 써보는 기술인만큼 향후 스펙으로 활용할 수 있도록 공부하기
4. vscode 자체 기능 및 유용한 확장프로그램에 대해 알아보기
 - 개발 시간 단축에 도움이 되는 여러 기법 익히기



기능 명세서

사용자의 관점에서 최종 결과물이 어떤 모습이며 어떻게 동작할 것인지를
기술한 문서

Front - End



React



React icons



React Router



Recoil



Node.js



Styled-components

Back - End



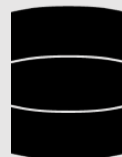
Spring Cloud



My SQL



Spring Boot



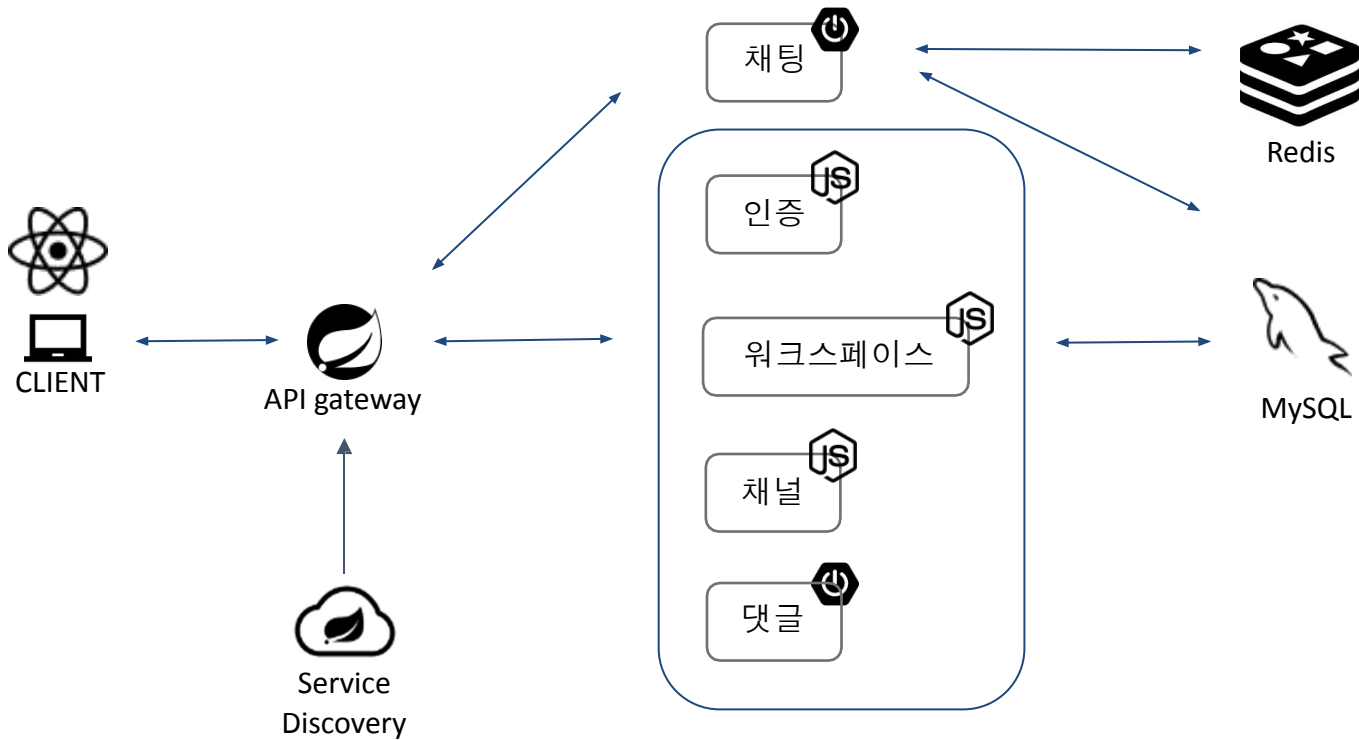
Spring Data JPA



Socket



Redis





달성하고 싶은 개인 목표를 위한 학습 계획



Ground Rules



core time - 9:00 - 12:00
daily scrum - 11:45 - 12:00

codeRunner

core time - 9:00 - 12:00
daily scrum - 11:45 - 12:00

Ground Rules

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.



Coding Convention - JavaScript



- 변수 이름

-camelCase를 사용

-모든 이름은 문자로 시작

- 연산자(= + - * /) 와 쉼표 뒤에 공백을 두기
- 코드 들여쓰기 : **Tab** 키 대신 **spacebar**키 2번으로 통일
- 파일 이름은 소문자로 통일
- **css** 속성 이름(글꼴 크기)에 하이픈을 사용
- 가독성을 위해서 한 라인은 **80**자를 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



Java

변수, 함수, 클래스 명명 규칙

- 변수는 명사형으로 작성
- 함수는 서술형으로 작성
- 클래스 명은 파스칼 케이스를 사용 (PascalCase)
- 기본적으로 카멜 케이스를 사용 (camelCase)
- 상수는 영문 대문자 스네이크 표기법 (Snake case)를 사용
- 한 줄 당 한 선언문만 사용
- 중괄호(일급함수 제외)
 - 중괄호는 하나의 구문만을 포함하고 있더라도 모든 제어문에 사용
 - if문의 경우, 하나의 구문만 포함할 경우, 같은 라인에 작성하고 중괄호를 생략
- 메서드 이름과 그 메서드의 파라미터 리스트의 시작인 괄호 (사이에는 빈 공간이 없어야 함)
 - `public void method(int a)`
- 여는 중괄호 { 는 클래스/인터페이스/메서드 선언과 동일한 줄의 끝에 찍어쓰기 포함하여 사용
 - `public void method(int a) {`
- 메서드들을 구분하기 위해서 각 메서드들 사이에는 한 줄을 비우기
- 한 줄이 100자를 넘기지 않도록 함



Commit Convention





Pull Request Convention





감사합니다.