



스詈 ~ 하게 주식과 친해져볼까?

J'STOCK

2팀 헬로드투헤븐

KAKAO TECH



# 목차

- 1 프로젝트 개요
- 2 프로젝트 팀 구성 및 역할
- 3 프로젝트 수행 절차 및 방법
- 4 프로젝트 수행 경과
- 5 자체 평가 의견



프로젝트 개요

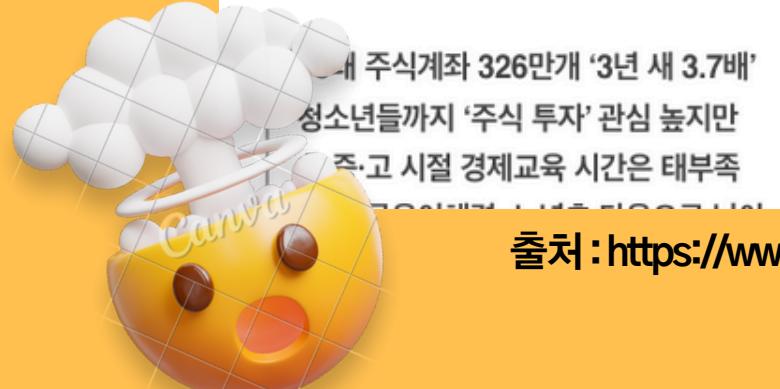
# 01 프로젝트 개요 프로젝트 기획 배경

## 청년들이 금융 지식을 갖추지 못한 채 빚까지 내서 투자하는, 이른바 ‘빚투’ 현상에 주목

### 투자하는 청년 늘어나는데 금융 이해력은 떨어진다

입력 : 2023.09.05 06:00 | 수정 : 2023.09.05 06:02 | 권정혁·유희곤 기자

#### ⑤ 교육은 없는데 일단 공격투자

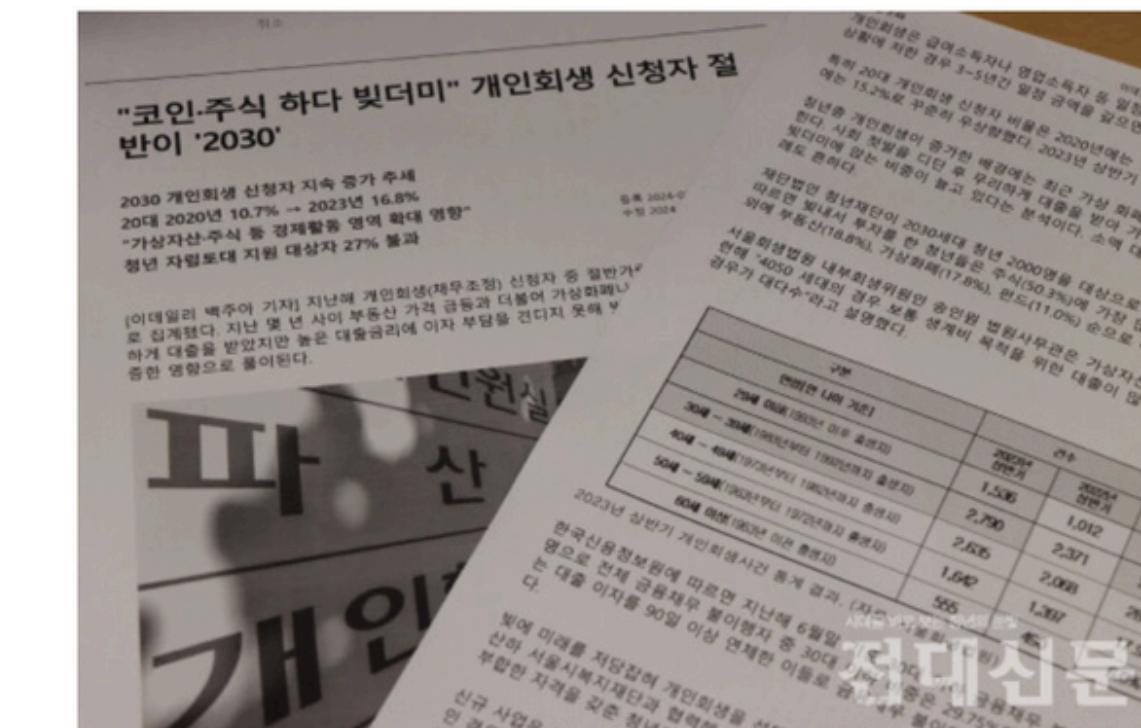


출처 : <https://www.khan.co.kr/economy/finance/article/202309050600035>

### 빚내서 투자하는 청년 증가…“청소년기부터 금융 교육 필요”

전대인(全大人) 주식 톡톡 ④ 주식 교육의 현주소

교육부재로 인한 청년 개인회생 증가  
유대인 교육법, 13살부터 투자 교육 시작



출처 : <https://press.cnunmedia.jnu.ac.kr/news/articleView.html?idxno=21638>

주요뉴스

제22대 총장

지역 여성주

“여수캠퍼스  
력 높이겠다”

하먼 우리 선  
안 해볼게?

취업 이상의

감수성 예민

온전한 민주

“제가 해내것”

## 투자하는 청년 늘어나는데 금융 이해력은 떨어진다

입력 : 2023.09.05 06:00 | 수정 : 2023.09.05 06:02 | 권정혁·유희곤 기자

#### ⑤ 교육은 없는데 일단 공격투자



10대 주식계좌 326만개 '3년 새 3.7배'  
청소년들까지 '주식 투자' 관심 높지만  
초·중·고 시절 경제교육 시간은 태부족

출처 : <https://www.khan.co.kr/economy/finance/article/20230905060003>

투자하는 청년 증가…“청소년기부터 금융 교육 필요”

총 10 주식 교육의 현주소

인회생 증가  
투자 교육 시즌

The image shows two newspaper clippings side-by-side. The left clipping is from 2003, titled '개인회생(재무조정) 신청자 증세 빙반가운데' (Individual bankruptcy (financial adjustment) applicants increase in number), and includes a table of statistics. The right clipping is from 2023, also titled '개인회생(재무조정) 신청자 증세 빙반가운데' (Individual bankruptcy (financial adjustment) applicants increase in number), and includes a larger, more detailed table comparing statistics between 2003 and 2023.

출처 : <https://press.cnumedia.jnu.ac.kr/news/articleView.html?idxno=21638>

# 01 프로젝트 개요

## 프로젝트 기획 배경

체결  
증거금  
등락률  
우선주  
호가

# 주식 접근성



# 01 프로젝트 개요

## 프로젝트 기획 배경

# 투자론

주변 조언

직관



# 01 프로젝트 개요 서비스 한줄 요약

“청년들이 최소한, 뉴스라는 정보에 크게 해서  
투자하는 습관을 만들 수 있도록 돋는 서비스”

# 01 프로젝트 개요 시연영상

The screenshot shows the U'STOCK mobile application interface. At the top is a search bar with the letter '카' (K) typed in. Below it is a list of stock entries, each with a small circular icon containing a letter (e.g., 카, 스, 커, 티, 밀, 인, 올, 삼, 미, 대), the company name, its stock number, and its current price and percentage change. The list includes: 카스 (1,204원, +0.18%), 카이노스或多 (3,980원, -0.2%), 카이바이오텍 (6,340원, 0%), 카카오 (37,250원, -0.2%), 카카오게임즈 (17,320원, -1.03%), 카카오뱅크 (22,250원, 1.14%), 스맥 (3,815원, +23.48%), 커뮤니케이 (6,130원, +20.91%), 디啵이홀딩스 (821원, +20.08%), 일리외서재 (17,940원, +18.72%), 인스코비 (2,105원, +17.07%), 올릭스 (25,100원, +16.47%), 삼성출판사 (17,150원, +16.27%), 미스터빌루 (1,634원, +15.31%), and 에이스티크 (0,699원, -1.6%). At the bottom are navigation icons for Home, Portfolio, and Game.



# 01 프로젝트 개요 기획중점 요소



**주식 공부  
동기부여**

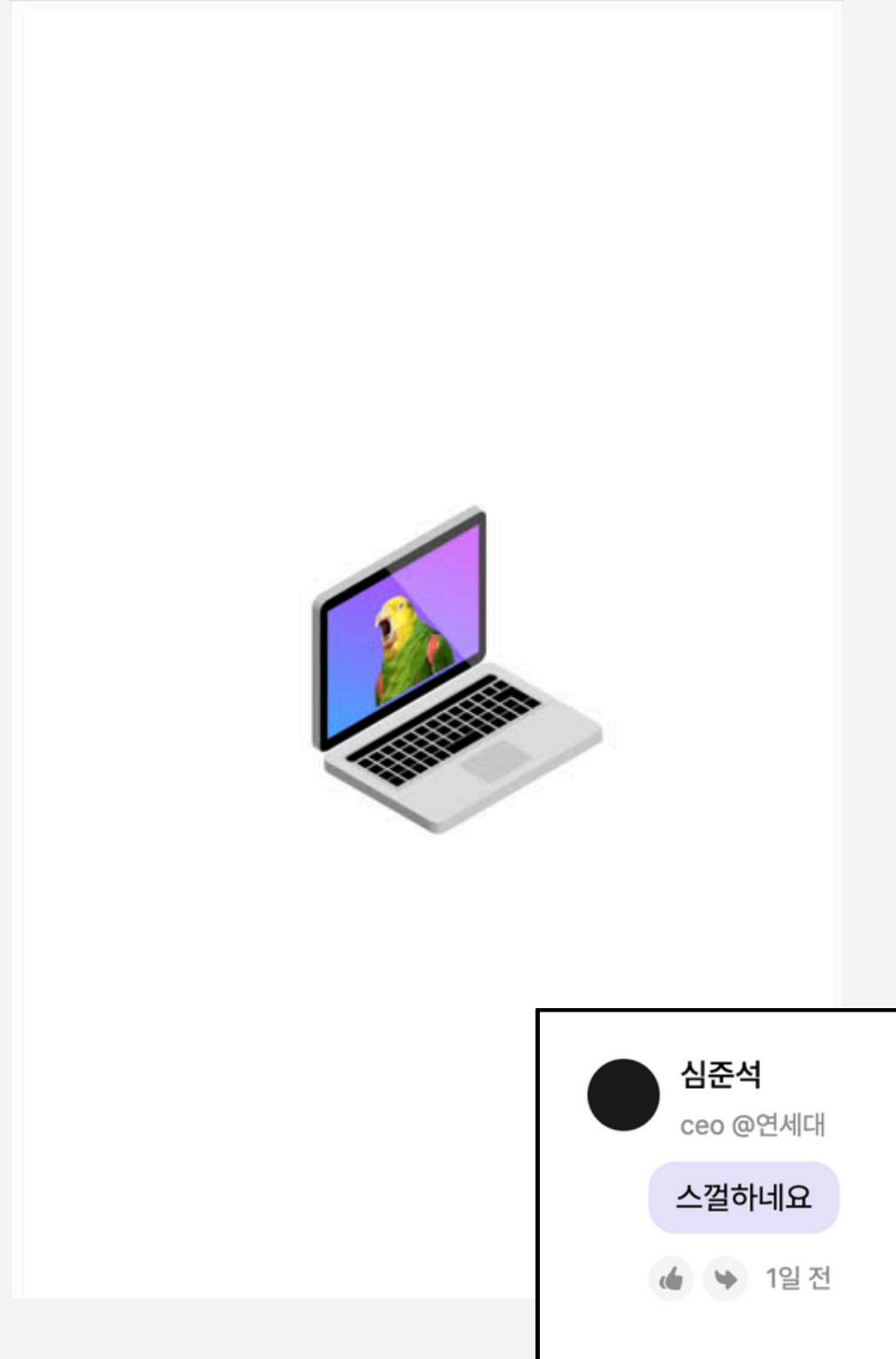
**크게 있는  
투자유도**

# 01 프로젝트 개요 동기부여



## 재미있는 포인트

### 컨셉



### 스껄 게임

veronica 님의 계좌잔고

거래가능금액 500,000 원

총 평가금액 500,000 원

작년 대비 0 원 (0%)

전체 대비 0 원 (0%)

보유주식      실시간 랭킹

아직 보유한 주식이 없습니다!

번호 종목 전년 올해 등락

종목 선택

A 바이오 8,000 원

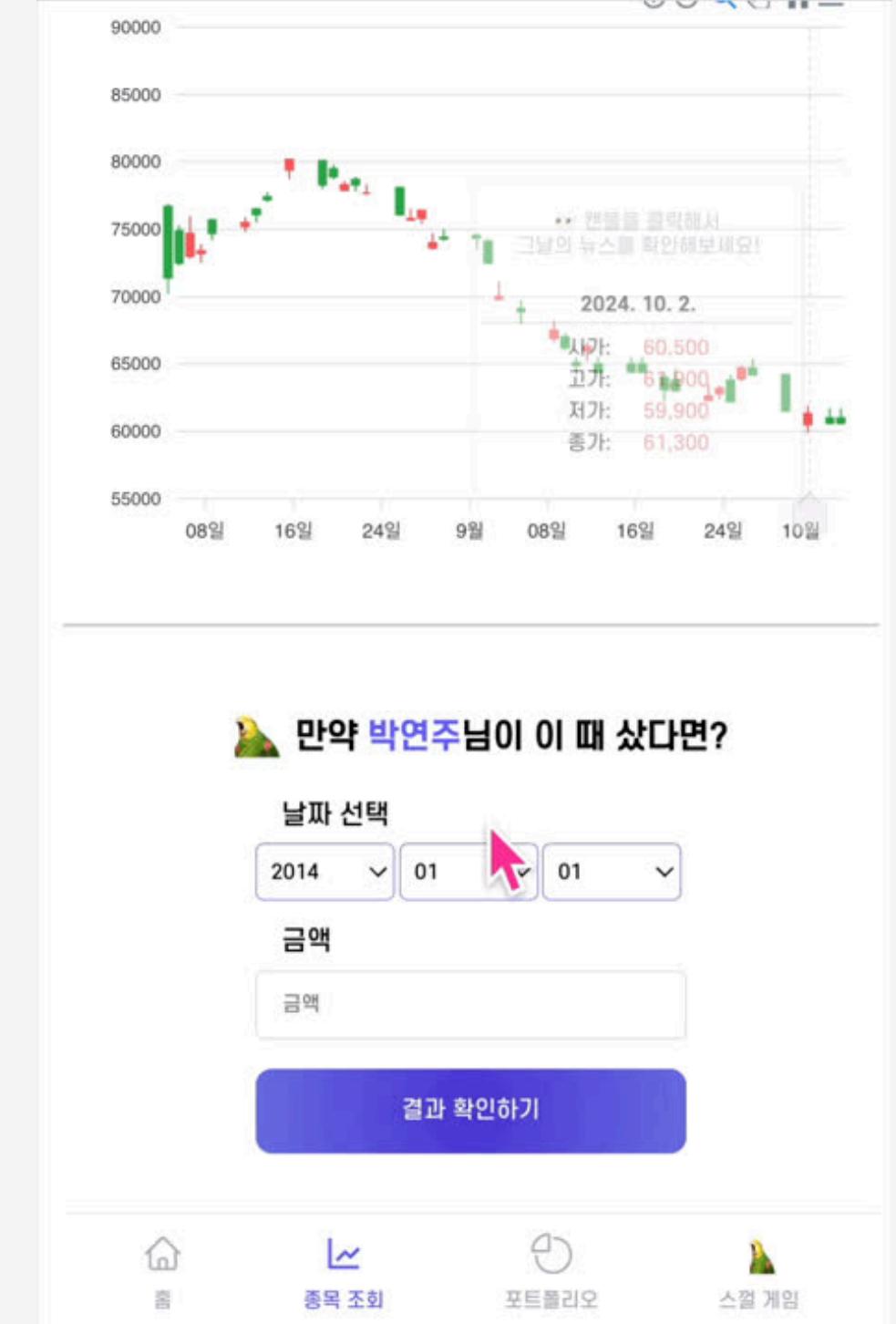
수량 선택      보유수량 : 0주

최대로 구매하기

거래가능금액 500,000 원

팔기      사기

### 스껄 계산기



# 01 프로젝트 개요 투자크기



## 크거있는 투자유도

### 나만의 뉴스

**U'STOCK**

연주 ☎

오늘의 종시

KOSPI	2594.36	▼ 16.02	-0.61%
KOSDAQ	778.24	▼ 2.77	-0.35%

오늘의 인기 종목

오늘 가장 거래가 많은 종목을 확인해보세요 !

III	피엘릭스 032580	1,603원	+8.30%
형	한성그룹 900270	427원	-6.77%
L	LK삼양 225190	2,610원	+19.18%
코	코풀스코리아 322780	1,185원	+7.53%
C	CNH 023460	346원	+14.57%

나만의 뉴스

내 포트폴리오에 등록된 종목들의 뉴스만 모아서 확인할 수 있어요 !

✗ 아직 보유하신 종목이 없어요 ✗  
내 포트폴리오를 생성하고, 종목을 추가해보세요 !

나만의 뉴스 만들러가기

홈 종목 조회 포트폴리오 스칼 게임

### 차트 일자별 뉴스

**U'STOCK**

연주 ☎

LG에너지솔루션 436,500원  
373220 ▲ 16,500원 (3.93%)

일 주 월

500000  
480000  
460000  
440000  
420000  
400000  
380000  
360000  
340000  
320000  
300000

2024. 10. 7.  
시가: 413,500  
고가: 426,000  
저가: 409,000  
종가: 420,000

06일 16일 24일 10월 03일

만약 박연주님이 이 때 샀다면?

날짜 선택  
2014 01 01

금액  
금액

결과 확인하기

홈 종목 조회 포트폴리오 스칼 게임

### 게임 내 정보거래소

정보거래소

현재 내 자산 : 500,000

종목 선택

추측 의심 확신

1단계 정보  
2014년 의 정보는 50,000원입니다.  
구매하시겠습니까?

돌아가기 구매



**프로젝트 탐구성 및 역할**

# 02 구성 및 역할

## 우리가 누구?



Uzu (송인아)

Mentor

#기술검토

#질의응답



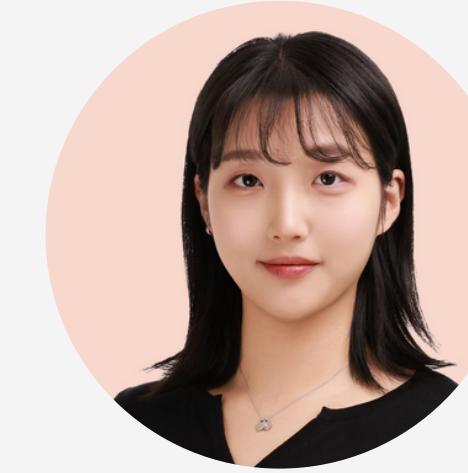
Lucy (전은주)

FrontEnd

#검색기능

#나만의뉴스

#게임결과창



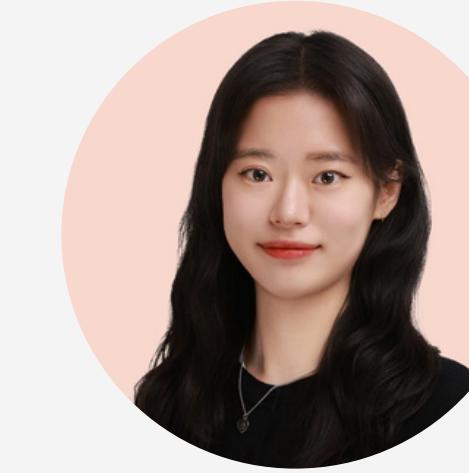
Elle (박연수)

FrontEnd

#모달

#차트뉴스

#성능개선



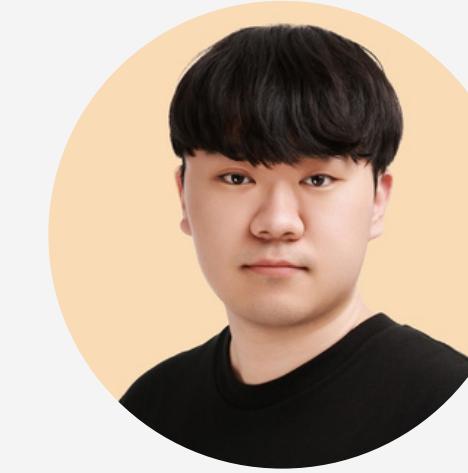
Veronica (박연주)

FrontEnd

#포트폴리오

#게임API연동

#Zustand



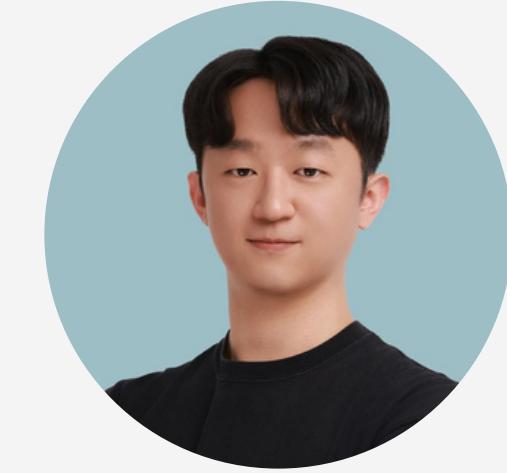
Jade (최진호)

BackEnd

#REST API

#OAuth2

#스케줄링



Kevin (유승범)

DevOps

#CI/CD

#모니터링

#게임AI

**프로젝트 수행 절차 및 방법**

# 03 수행절차 및 방법 스프린트

번호	구분	기간	활동	비고
스프린트 1	사전 기획	7/22(월) ~ 8/11(금)	<ul style="list-style-type: none"><li>프로젝트 주제 선정 및 기획</li><li>화면설계서 및 API 설계서 작성</li></ul>	아이디어 선정
스프린트 2	■MVP 기능개발	8/12(월) ~ 8/25(금)	<ul style="list-style-type: none"><li>한국투자증권 API를 통한 주식 목록 및 차트 상세보기 기능</li><li>주식 차트 캔들 클릭 시 해당 일자의 뉴스 조회 기능</li><li>주식 포트폴리오 생성 기능</li></ul>	
스프린트 3	1차 배포 및 유지보수	8/26(월) ~ 9/8(금)	<ul style="list-style-type: none"><li>TC 작성, QA 진행 및 Issue 처리</li><li>서비스홍보</li></ul>	9/3 디스콰이엇을 통한 서비스 1차 홍보
스프린트 4	게임 기능개발	9/9(월) ~ 9/20(금)	<ul style="list-style-type: none"><li>게임 시나리오 기획</li><li>게임 기능 화면설계서 및 API 명세서 작성</li><li>게임 기능 개발</li><li>게임 관련 데이터 수집 및 정규화</li></ul>	
스프린트 5	2차 배포 및 유지보수	9/23(월) ~ 10/11(금)	<ul style="list-style-type: none"><li>게임 기능 TC 작성 및 QA진행 및 ISSUE 처리</li><li>서비스2차 홍보</li></ul>	10/9 디스콰이엇을 통한 서비스 2차 홍보

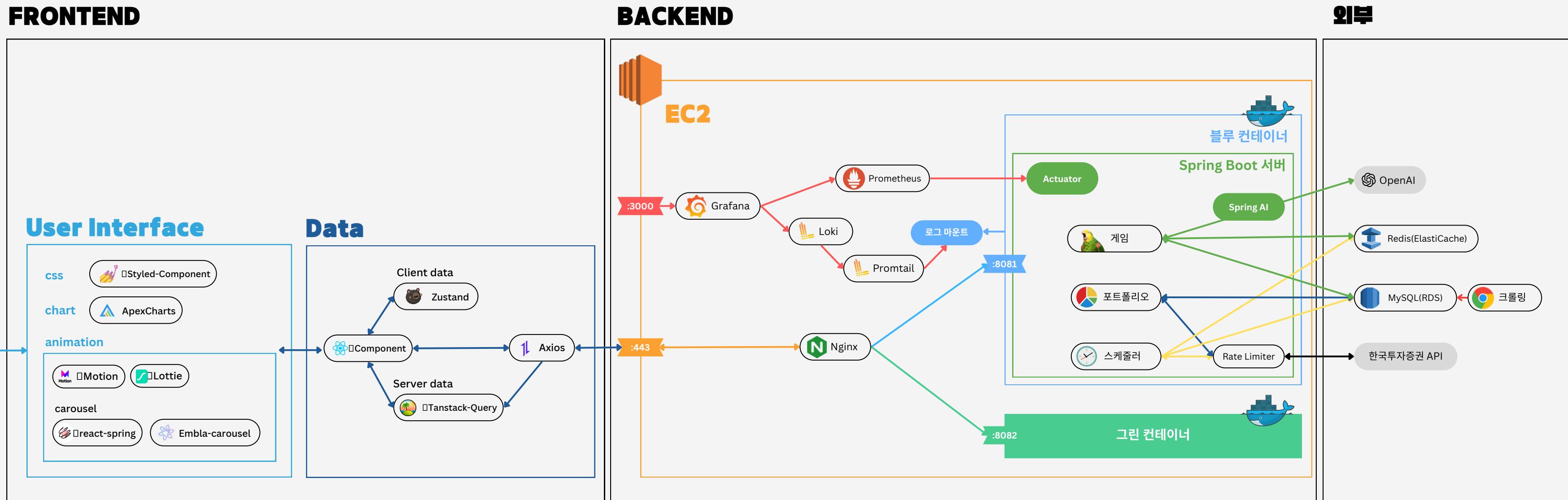
총 개발기간

7/22(월) ~ 10/11(금) 총 12주

**프로젝트 수행 결과**

# 04 수행결과

## 전체 프로젝트 구조



# 04 수행결과

## 프로젝트 구현 내용 : Backend

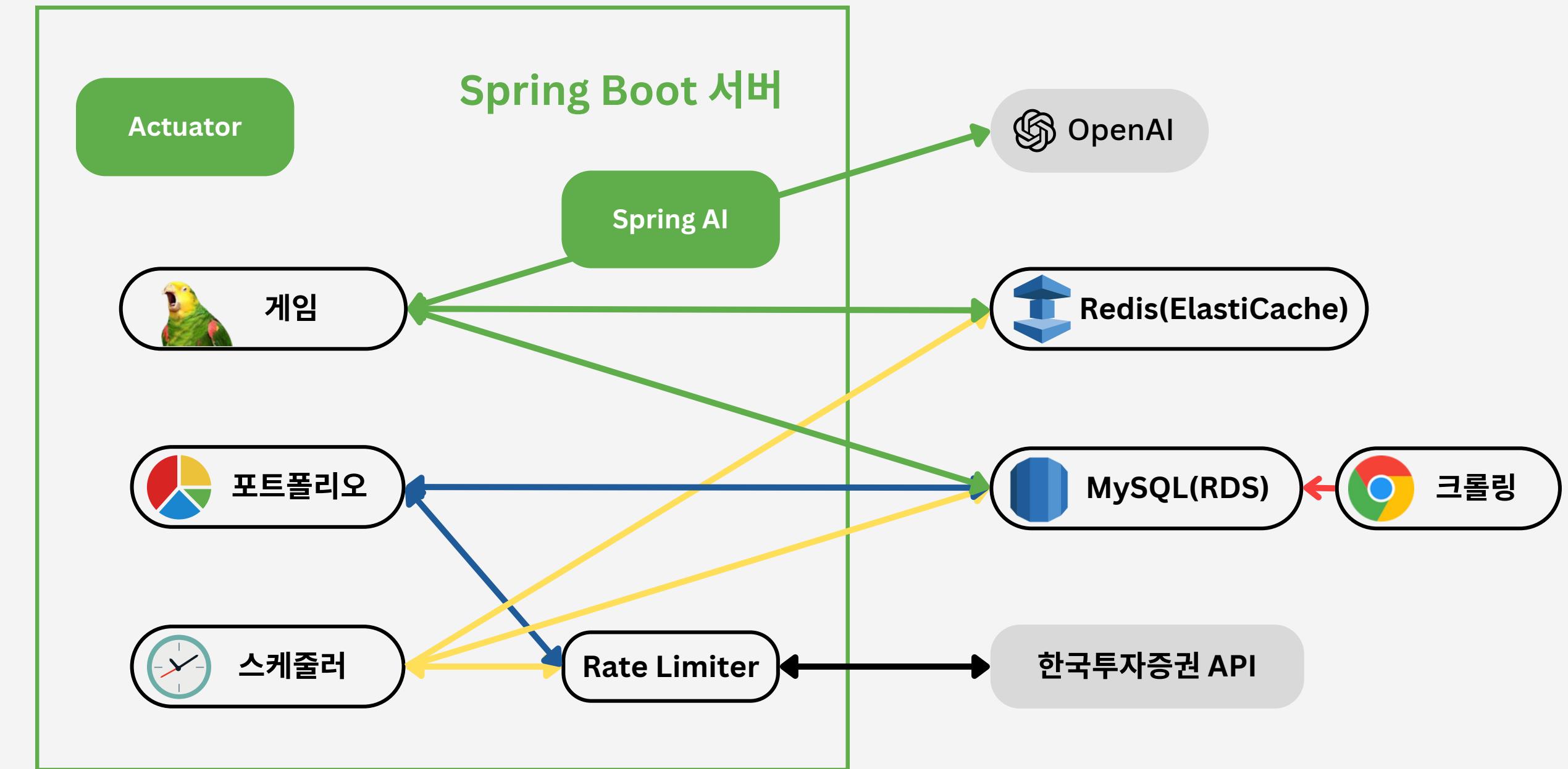


✓ OAuth2 인증/인가

✓ 외부 API 연동

✓ 데이터 UPDATE 스케줄링

✓ 게임 기능 구현



# 04 수행결과 키 포인트 1. Redis 저장

☒ AS-IS

## 1. 실시간 데이터 캡신 제한

WEBSOCKET : 1세션  
실시간체결가 + 호가 + 예상체결 + 체결통보 등 실시간 데이터 합산 **41건까지 등록 가능**  
※ 국내주식/해외주식/국내파생/해외파생 모든 상품 실시간 합산 41건  
※ 체결통보의 경우 HTS ID 단위로 등록하여, ID에 연결된 모든 계좌의 체결 통보 수신  
※ 계좌(앱키) 단위로 유량 제한(1개의 계좌(앱키)당 1세션)  
※ 1개의 PC에서 여러 개의 계좌(앱키)로 세션 연결 가능

## 2. 캡신이 필요한 데이터 많음

```
select count(*) "종목 수" from stocks;
```

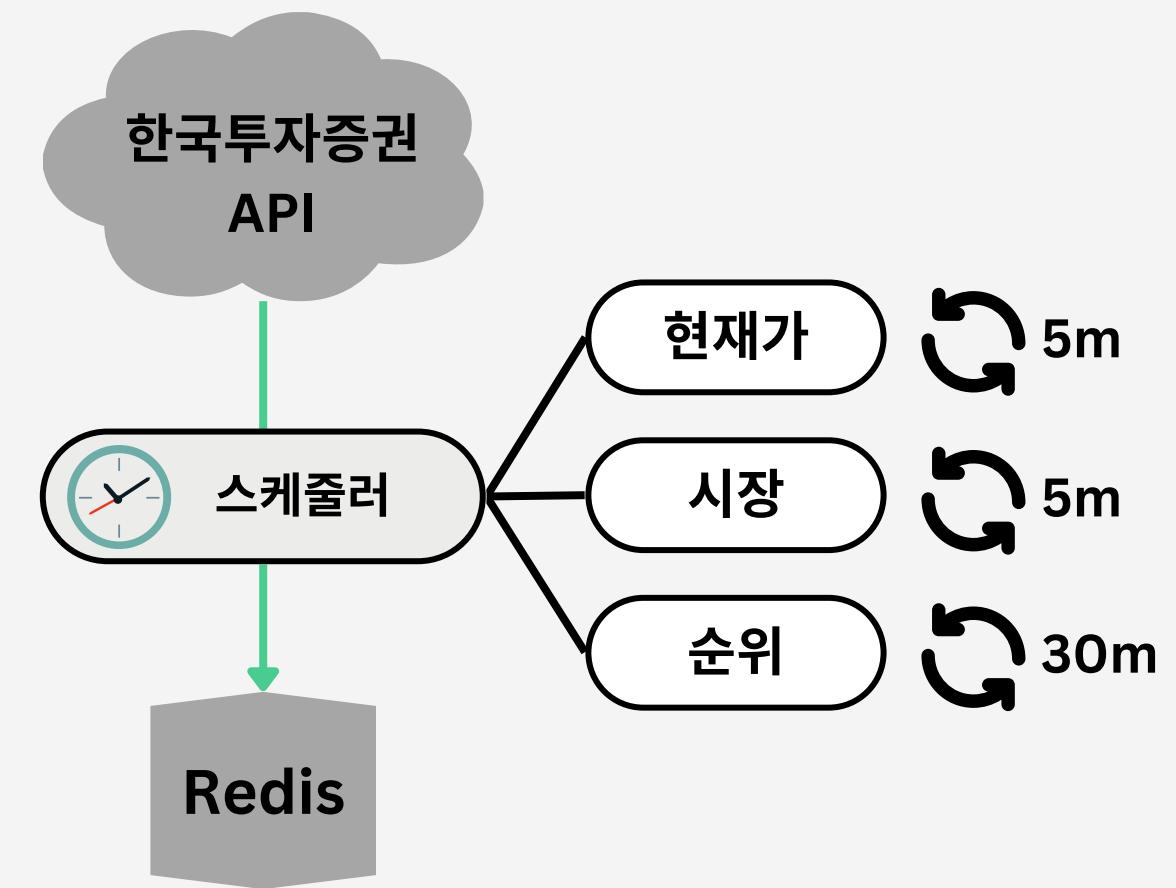
目 '종목 수' ▾  
1 2836

## 3. 빠른 응답속도 요구

종목을 사용하는 모든 API에서 현재가, 등락률 요구

카카오			
kakao	카카오 035720	37,400원	0.13%
kakao games	카카오게임즈 293490	17,010원	-2.8%
kakao bank	카카오뱅크 323410	22,250원	1.14%

TO-BE



일정 시간마다 스케줄러 실행

서비스 안정성과 데이터 신선도 조율

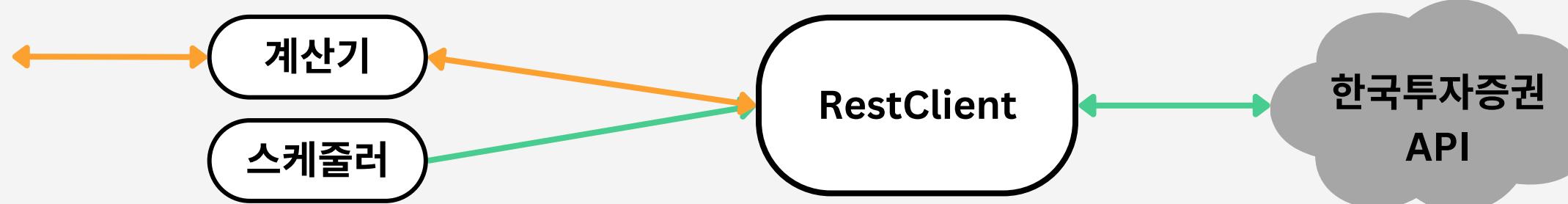
요청 응답 속도 ↑

# 04 수행결과 키 포인트 2. Rate Limiter 도입

BEFORE



외부 API 요청 20회 제한



예외 발생율 ↑

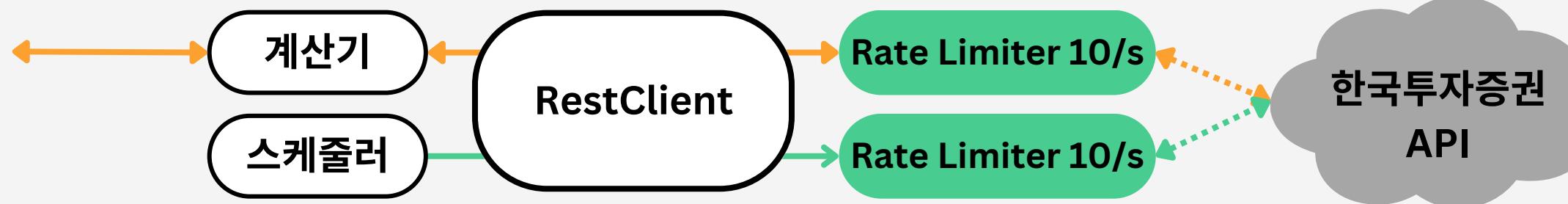
스케줄링 처리시간 ↑

요청 처리속도 ↓

AFTER



Rate Limiter 도입



예외 발생 X

스케줄링 처리 효율 향상

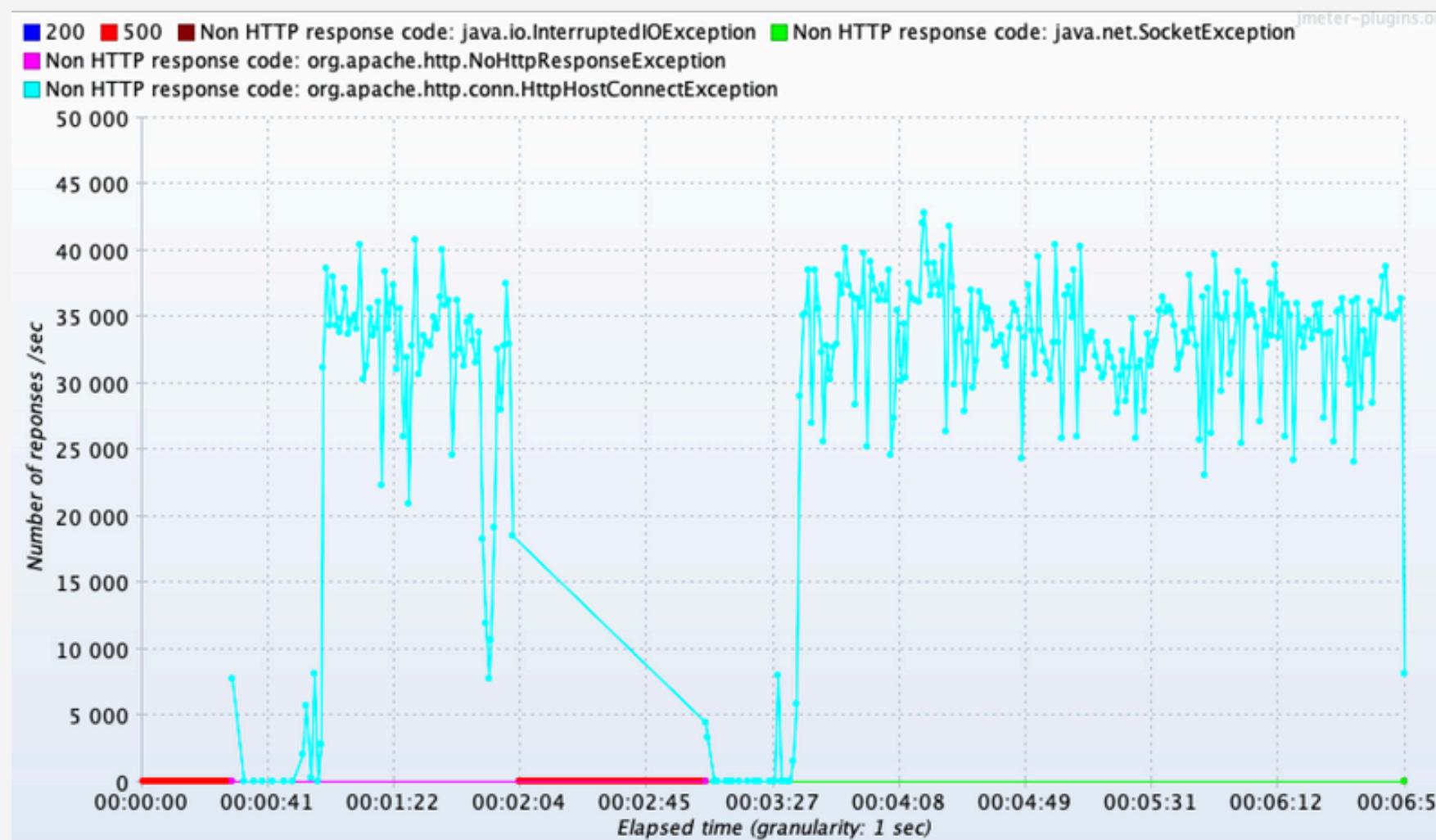
안정적인 요청 처리

Server Flow

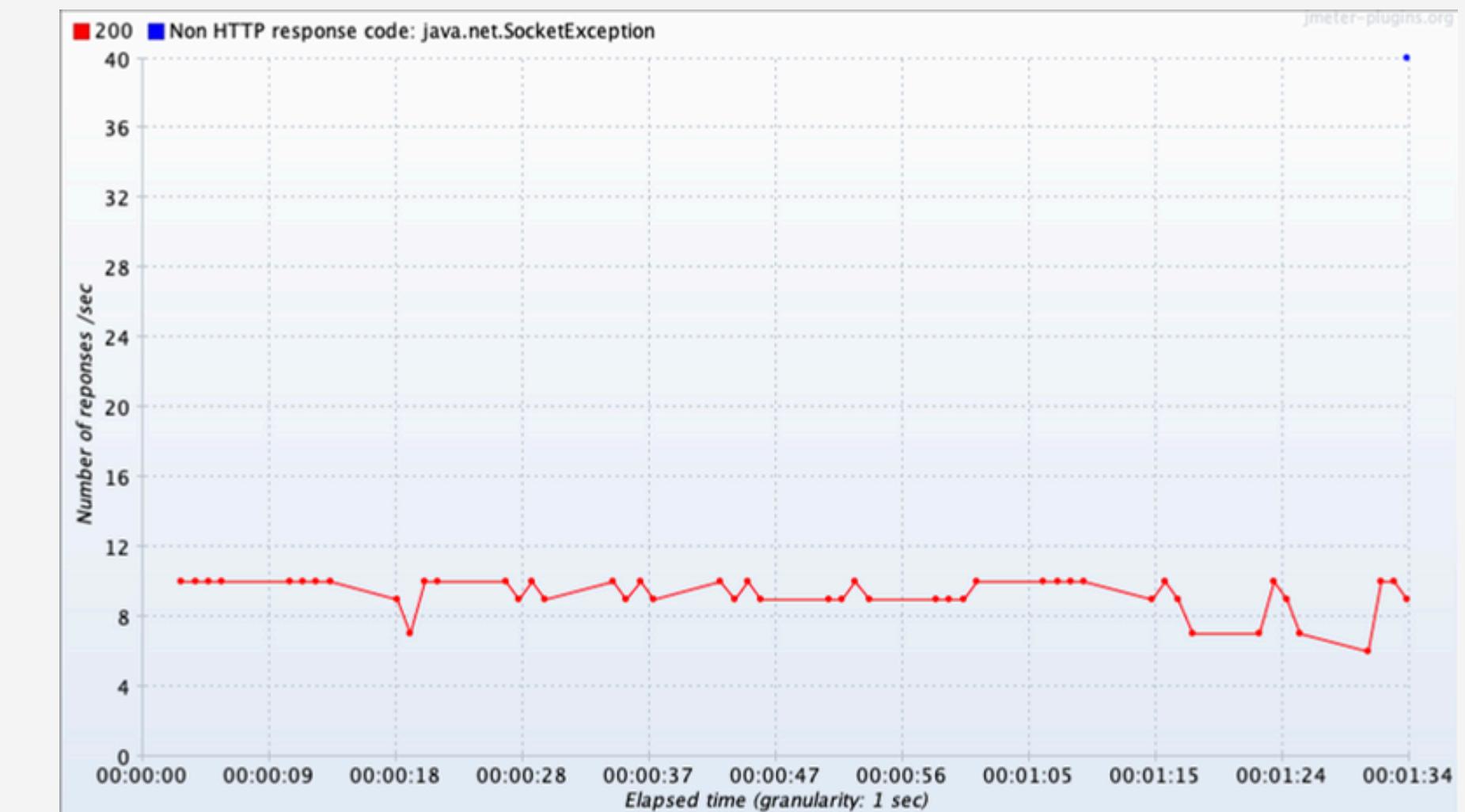
User Flow

# 04 수행결과 키 포인트 2. Rate Limiter 도입

BEFORE



AFTER



Rate Limiter 도입 이후, API 연결과 관련된 오류가 발생하지 않음

# 04 수행결과 키 포인트 3. 생성형 게임 AI 구현

## 1 OpenAI로 게임AI 구현

응답 타입은 반드시 주어진 JSON 형태의 문자열으로만 반환(JSON 객체 외의 모든 문자는 전부 빼)해야 하고 닉네임을 키로 반환해.

id는 요청 정보에 넣어서 보낸 종목 정보 안에 있는 id와 정확하게 매칭해서 반환해.

판매할 종목의 아이디, 이름, 수량을 객체로 묶은 후, 리스트로 반환해. 그리고 왜 그런 선택을 했는지 이유도 같이 반환해.

아래의 예시를 참고해.

```
{"COM1": {"list": [{"id": 1, "name": "A 전자", "quantity": 3}, {"id": 2, "name": "C 게임", "quantity": 5}], "reason": "종목을 선택한 이유"}, "COM2": {...}, ...}
```

```
buyResponse: {  
    COM3={  
        list=[{id=13, name=F IT, quantity=33},  
        {id=1, name=C 전자, quantity=6}],  
        reason=F IT는 세계 생활가전 시장의 성수기 돌입으로 수요가 급증할 것으로 예상되어  
        매력적인 투자처입니다. C 전자는 자회사 분리로 인해 구매 규모가 대폭 확대되어 성장  
        가능성이 높습니다.}, ...  
}
```

```
public class GameAiSelectDto {  
    private List<GameAiStockDto> sell;  
    private List<GameAiStockDto> buy;  
    private String reason;  
}
```

```
public class GameAiStockDto {  
    private long id;  
    private String name;  
    private int quantity;  
}
```

## 2 AI의 비정상적인 응답 예외처리

### Case 1. 코드 블럭을 반환하는 예외

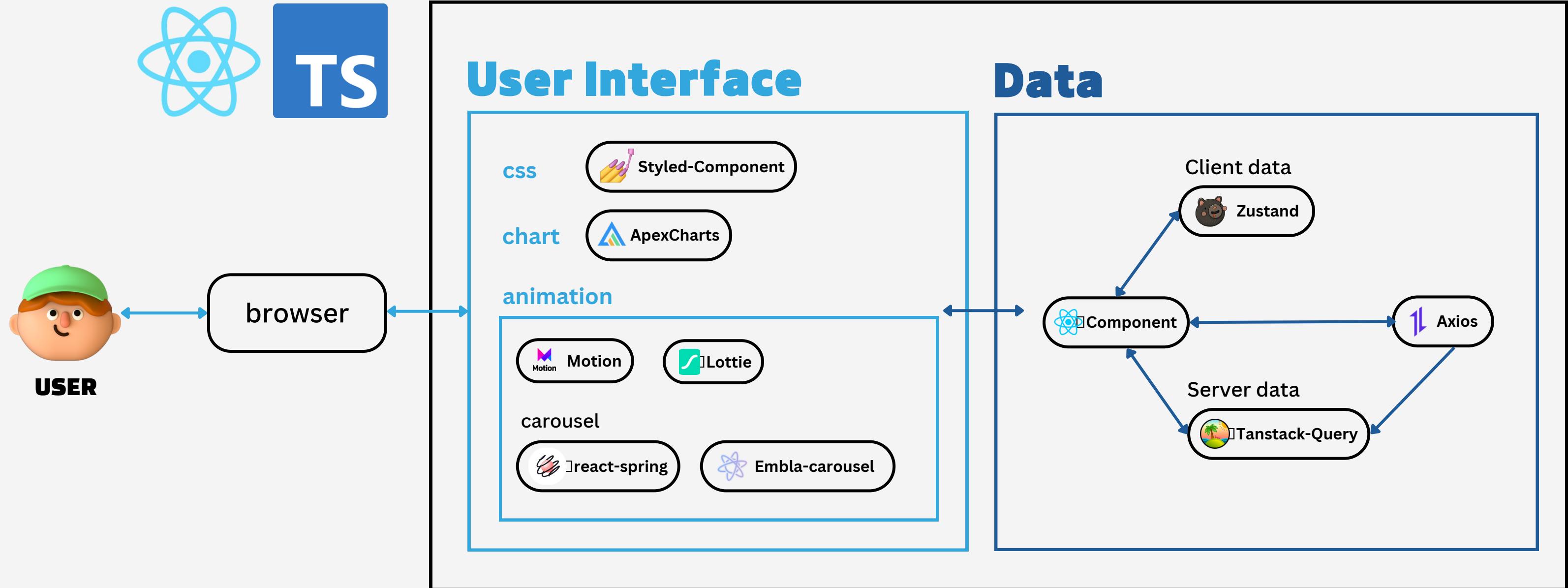
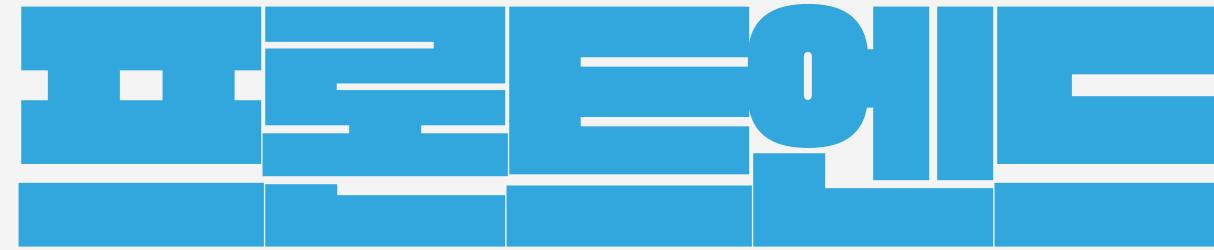
```
if (responseJson.startsWith("```")) {  
    responseJson = responseJson.replace( target: "```", replacement: "" )  
    | .replace( target: "json", replacement: "" );  
}
```

### Case 2. 빈 JSON을 반환하는 예외

```
if (sellResponse.get(aiNickname).toString().equals("{}")) {  
    sellResponse.put(aiNickname, new HashMap<>());  
}
```

# 04 수행결과

## 프로젝트 구현 내용 : FrontEnd



# 04 수행결과

## Frontend 표준 개발 환경

```
>   public  
  < src  
    > apis  
    > assets  
    > components  
    > constants  
    > hooks  
    > Pages  
    > services  
    > store  
    > Styles  
    > types  
    > util  
  App.css  
  App.tsx  
  index.css  
  index.tsx  
  react-app-env.d.ts  
  reportWebVitals.ts  
  Router.tsx  
  routes.ts
```

Page Layer

Product (App)

페이지 컴포넌트

Component Layer

컴포넌트

Business Layer

hooks

훅 A

훅 B

훅 C

...

services

서비스 A

서비스 B

...

Store Layer

queries

쿼리 A

쿼리 B

...

stores

스토어 A

스토어 B

...

Utility Layer

apis, utils 등 각종 유ти리티 함수 존재

# 04 수행결과 키 포인트 1. Zustand 도입

AS-IS



비효율적인 state 관리

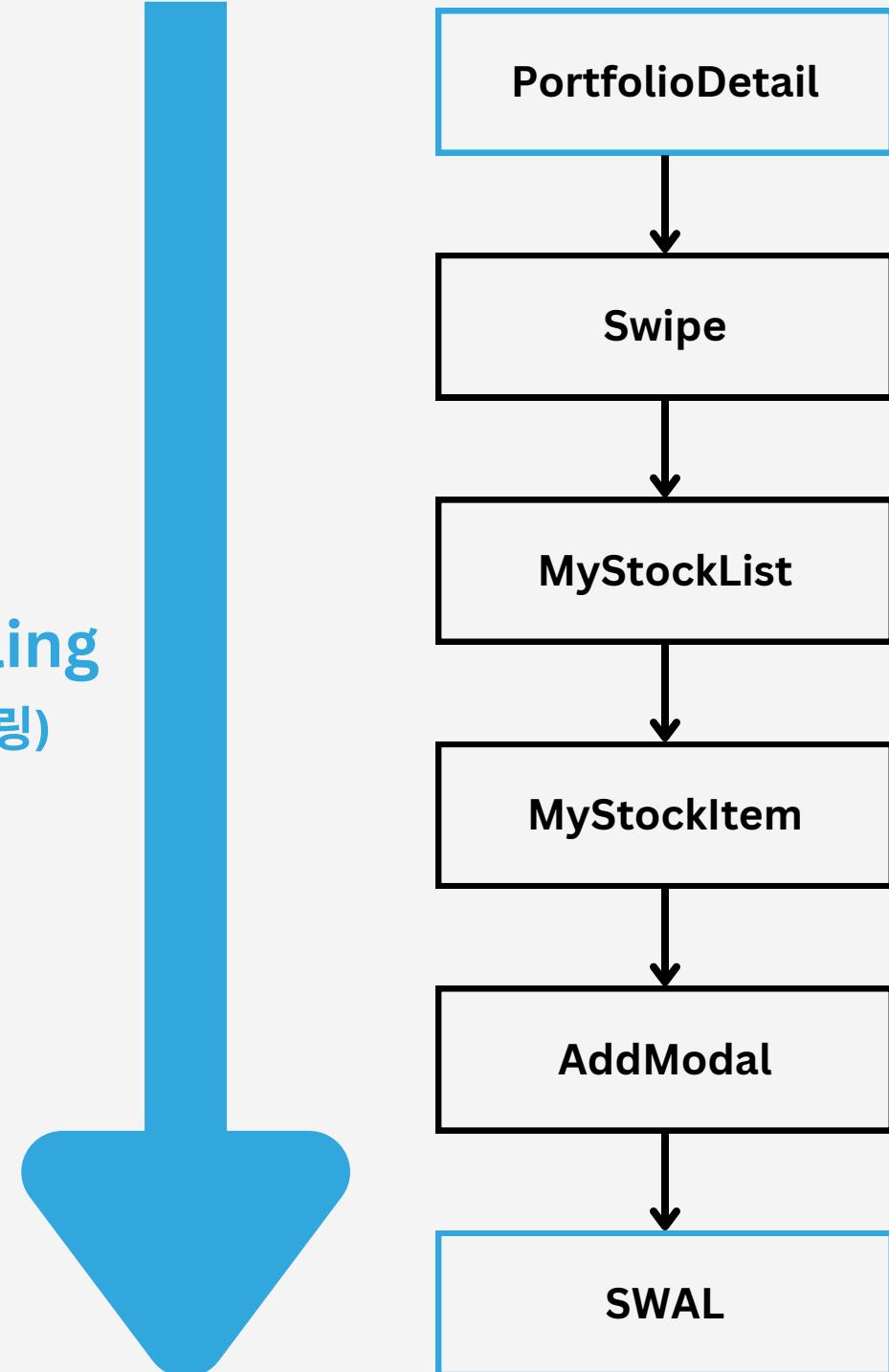
컴포넌트, 모달 수

props drilling

코드 복잡, 유지보수 어려움



PropsDrilling  
(프롭스 드릴링)



# 04 수행결과 키 포인트 1. Zustand 도입

TO-BE

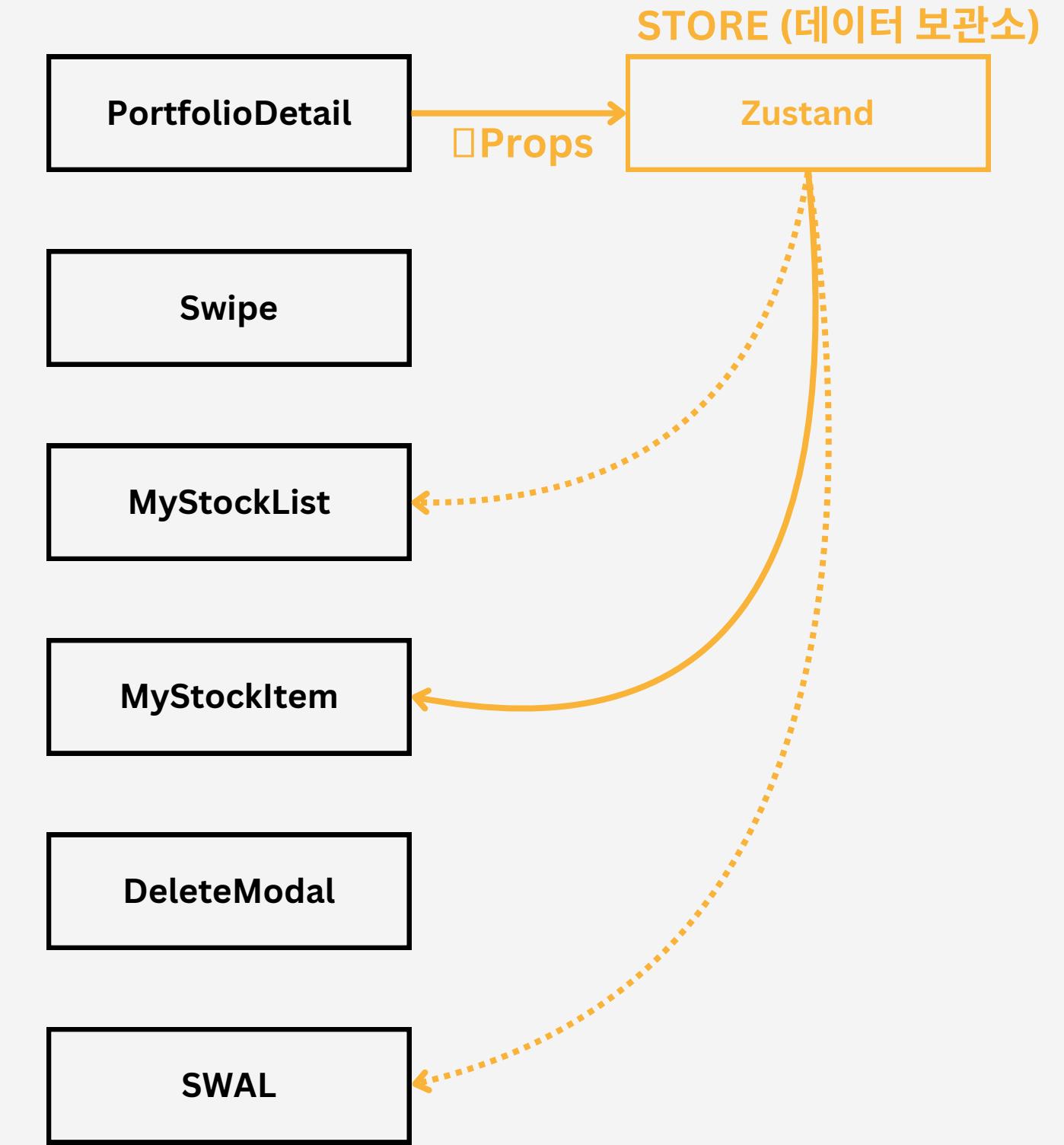


☒ **zustand** 도입

state를 전역으로 관리

불필요한 리랜더링 x

사용 용이, 코드수



# 04 수행결과

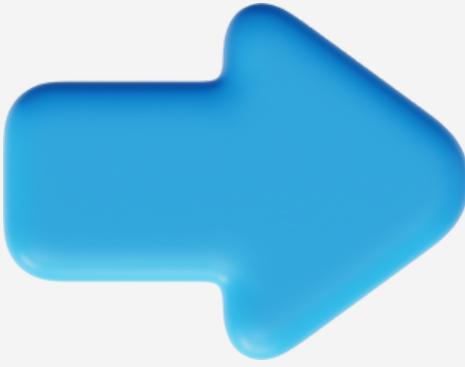
## 키 포인트 2. Tanstack-query 도입

AS-IS

```
const Home: React.FC = () => {
  const nav = useNavigate();
  const [market, setMarket] = useState<MarketDataProps | null>(null);
  const [list, setList] = useState([]);

  // 오늘의 종시 데이터
  useEffect(() => {
    axios
      .get(`${process.env.REACT_APP_API_URL}/v1/stocks/market`, {
        withCredentials: true,
        headers: {
          "Content-Type": "application/json",
        },
      })
      .then((res) => {
        if (res.status === 200) {
          const marketData = res.data;
          setMarket(marketData);
        } else if (res.status === 401) {
          nav("/login");
        }
      })
      .catch((e) => {
        nav("/error");
      });
  }, []);

  // 인기 종목 리스트 데이터
  useEffect(() => {
    axios
      .get(`${process.env.REACT_APP_API_URL}/v1/stocks?order=top`, {
        withCredentials: true,
        headers: {
          "Content-Type": "application/json",
        },
      })
      .then((res) => {
        if (res.status === 200) {
          const stockData = res.data;
          setList(stockData);
        } else if (res.status === 401) {
          nav("/login");
        }
      })
      .catch((e) => {
        nav("/error");
      });
  }, []);
}
```



123줄의 컴포넌트 코드

비즈니스 로직이 컴포넌트에 집중

중복된 API 호출 관리

TO-BE

```
const Home: React.FC = () => [
  // 오늘의 종시 데이터
  const { data: market, isLoading, isError } = useMarket();
  // 인기 종목 데이터
  const { data: list } = usePopularStock();

  if (isLoading) return <div>Loading...</div>;
  if (isError) return <Link to="/error" />;
]
```

83줄의 컴포넌트 코드

비즈니스 로직이 역할에 맞게 분산

러닝 커브

# 04 수행결과 키 포인트 3. 성능 최적화

1

코드 스플리팅 및  
동적 import 적용

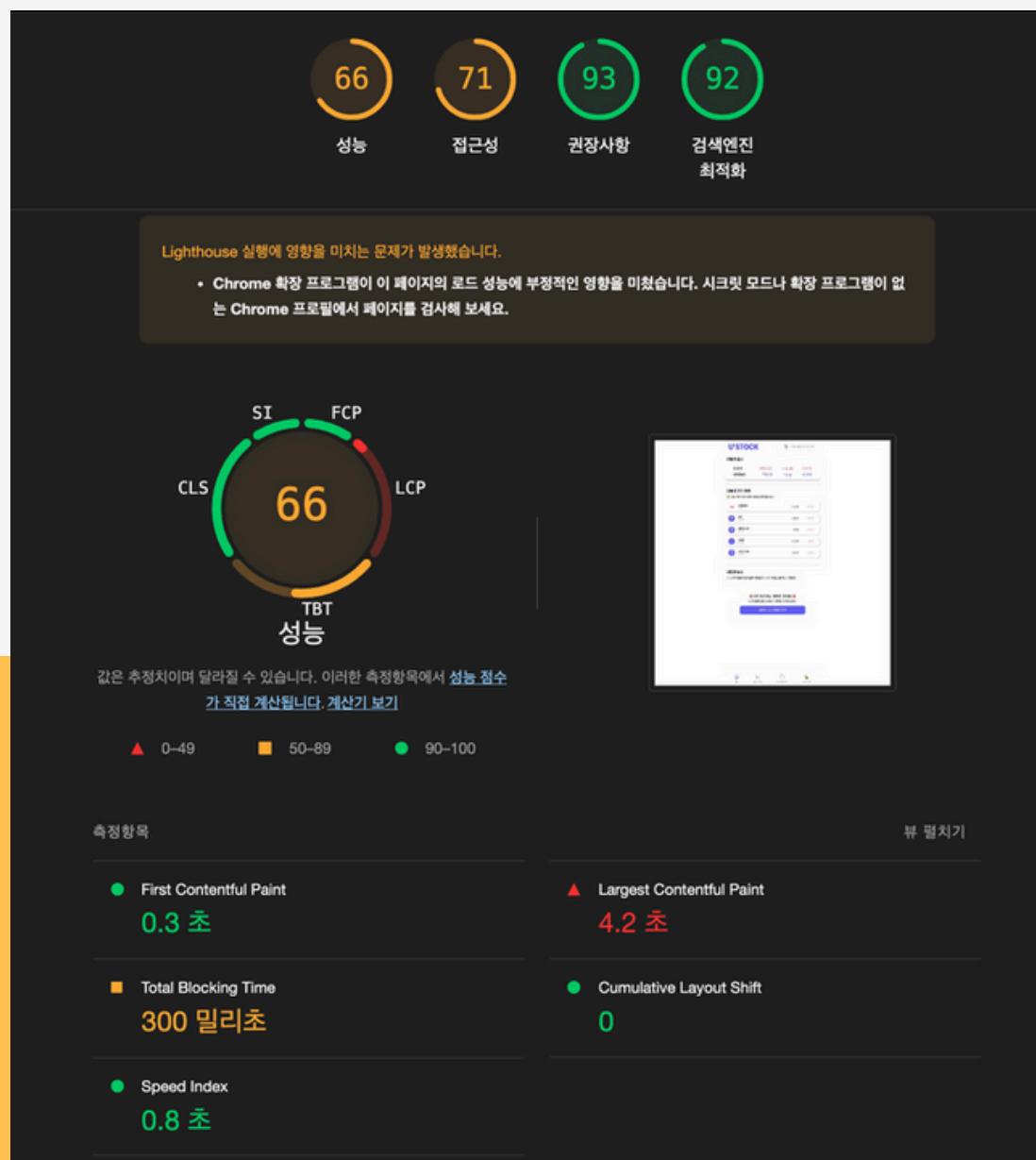
2

lazy-loading을 적용해  
이미지 최적화

3

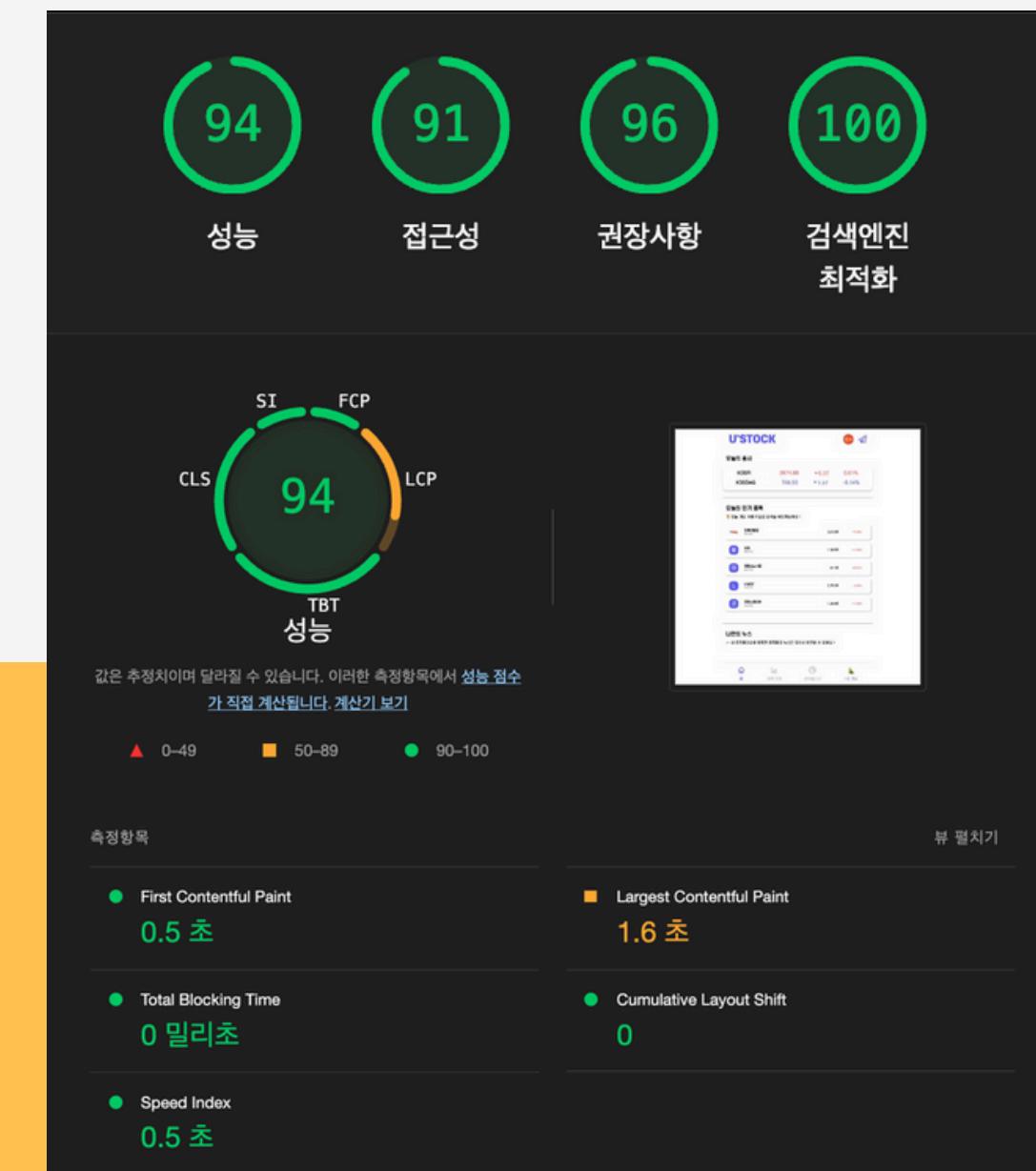
아이콘 개별 import로  
번들 크기 최적화

## AS-IS



## TO-BE

성능 42.4% 개선

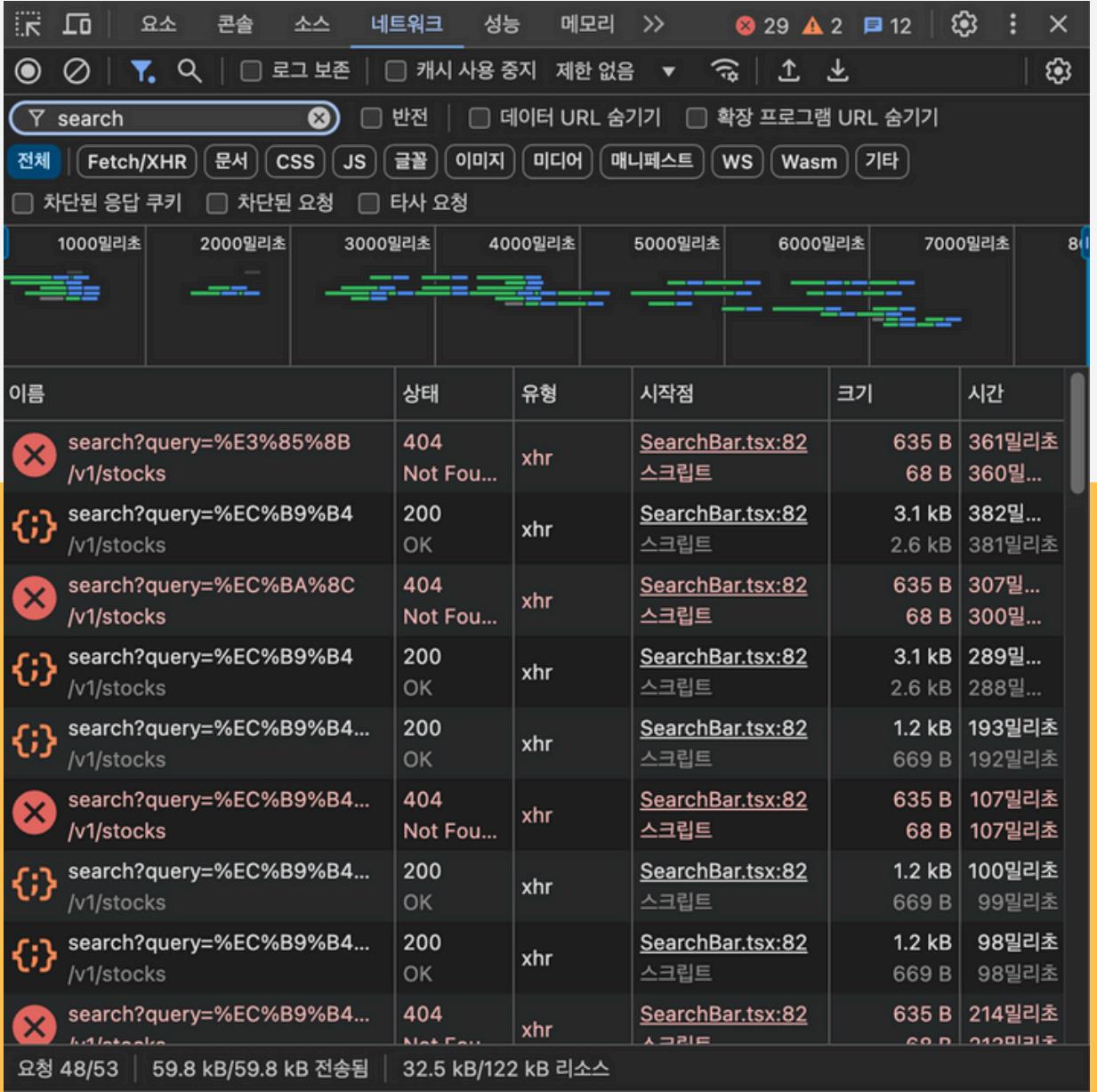


# 04 수행결과 키 포인트 4. Debounce 함수 적용

AS-IS

총 48번의 API 호출

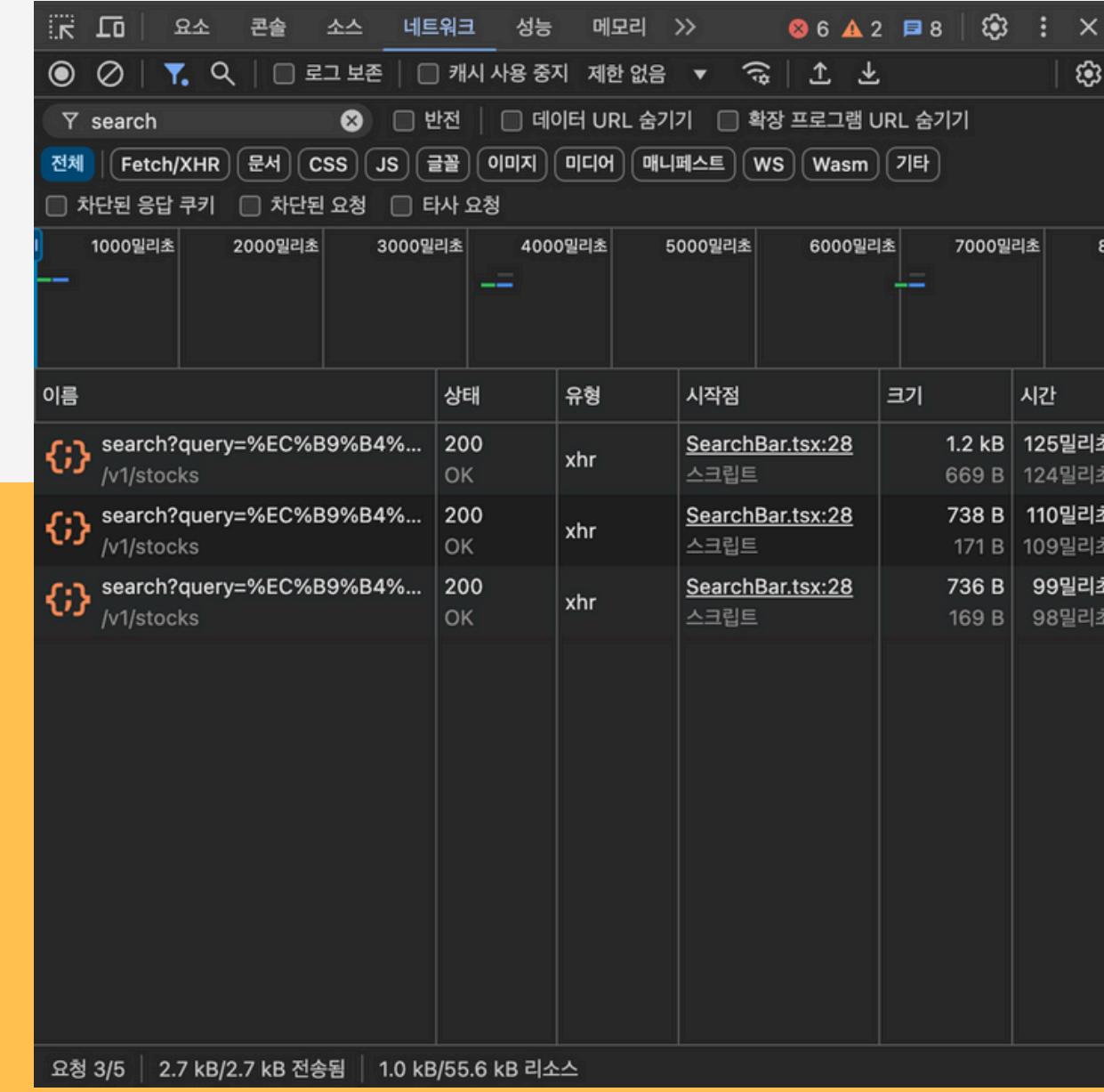
유효하지 않은 API 호출 19건



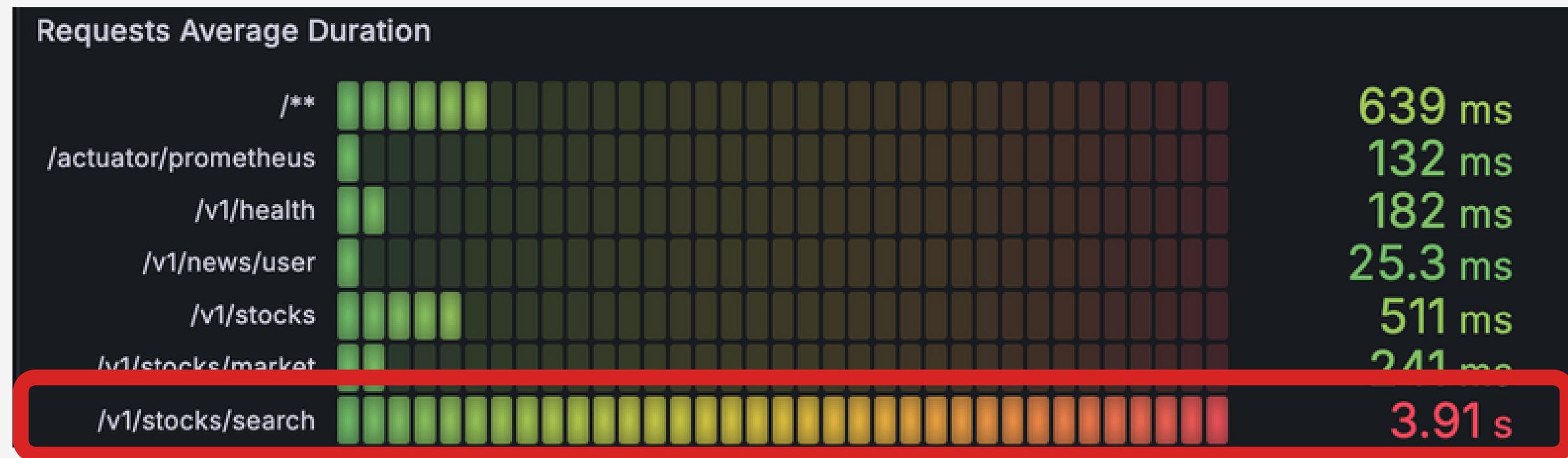
TO-BE

총 3번의 API 호출

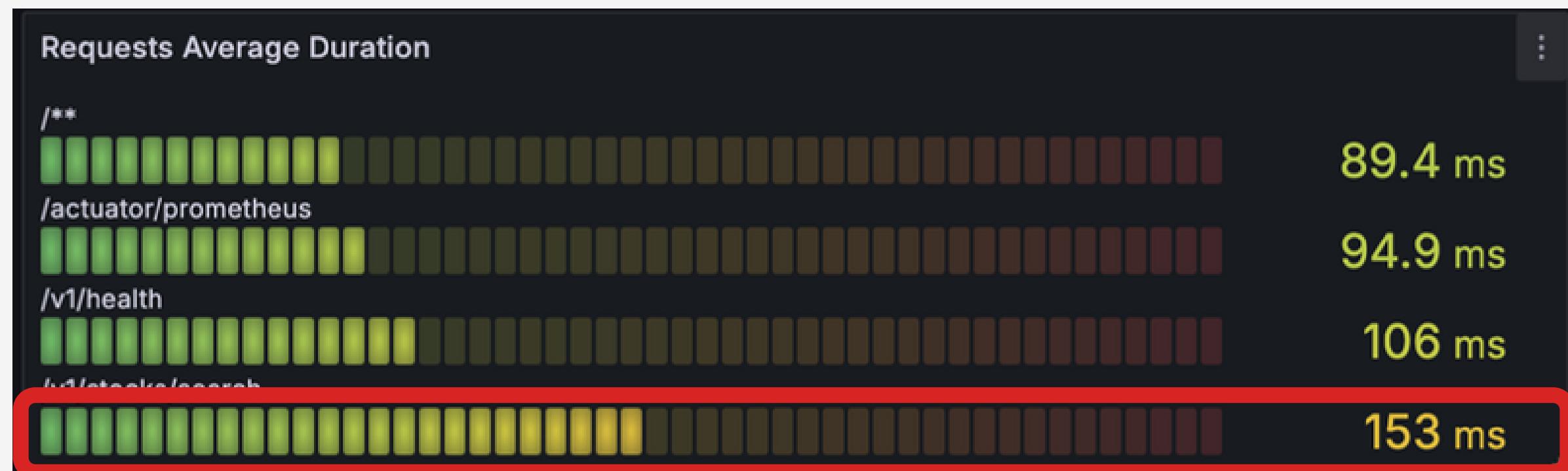
유효하지 않은 API 호출 0건



# 04 수행결과 키 포인트 4. Debounce 함수 적용



응답 속도 약 96% 개선



# 04 수행결과

## 프로젝트 구현 내용 : DevOps

# 데브옵스

운영용 도메인

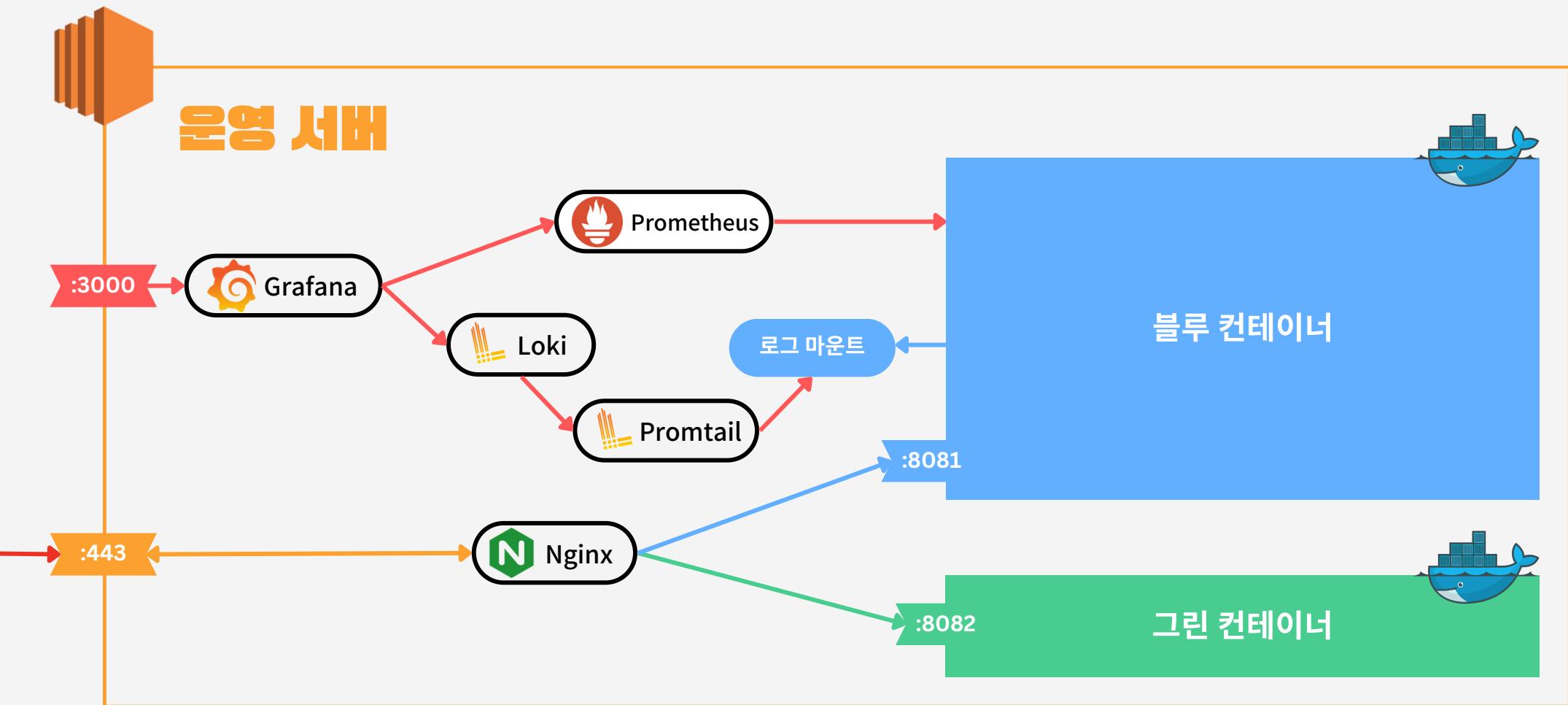
ustock.site

운영용 버켓

ustock-prod-bucket

개발용 버켓

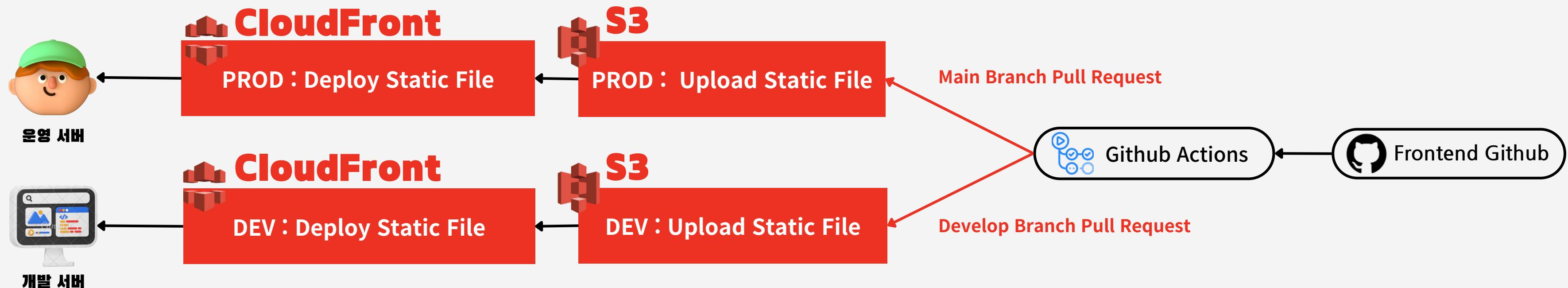
dev.ustock.site



개발 서버

# 04 수행결과 프로젝트 구현 내용 : DevOps

## FRONTEND DEPLOYMENT



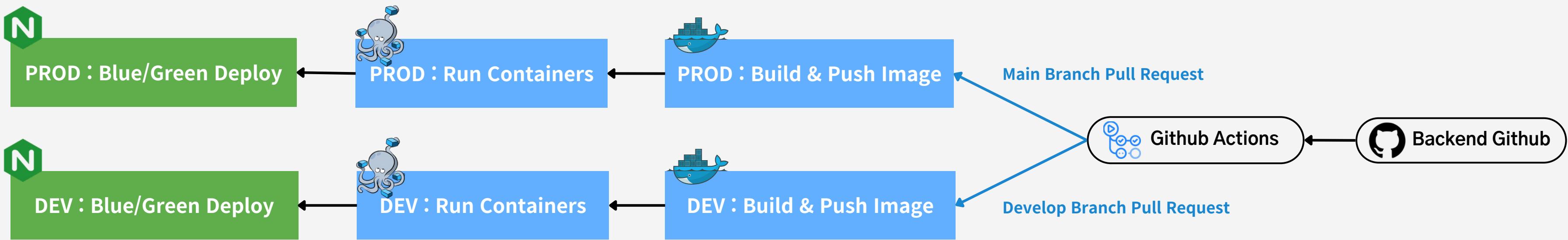
1 React 정적 파일 빌드

2 정적 파일 S3 업로드

3 CloudFront 배포

# 04 수행결과 프로젝트 구현 내용 : DevOps

## BACKEND DEPLOYMENT



1 Spring 이미지 빌드

3 Docker Compose 실행

2 Docker Hub 업로드

4 Blue/Green 무중단 배포

## 04 수행결과 키 포인트 1. 통합 모니터링 대시보드 구축

**AS-IS**



**로그 확인 어려움**

**컨테이너 로그 직접 확인**

**자세한 메트릭 확인 불가**

**API 응답 속도 파악 어려움**

**TO-BE**

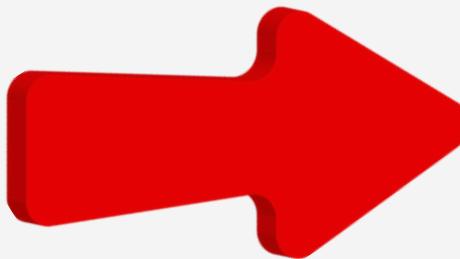


**통합 모니터링 대시보드**

**웹을 통해 간단하게 조회**

**JVM까지 확인 가능**

**엔드포인트별 응답 속도 측정**



# 04 수행결과 키 포인트 1. 통합 모니터링 대시보드 구축

## 간편하게 로그 확인

```
|> 2024-10-10 12:45:00.194 2024-10-10T12:45:00.000+09:00 INFO 1 --- [ustock] [scheduled-task-pool-4] c.h.u.service.cron.StockCronService : 현재가 크론잡 시작
|> 2024-10-10 12:34:43.622 2024-10-10T12:34:43.565+09:00 INFO 1 --- [ustock] [scheduled-task-pool-3] c.h.u.service.cron.StockCronService : 현재가 크론잡 종료
|> 2024-10-10 12:32:56.684 2024-10-10T12:32:56.496+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-1] c.h.u.e.common.GlobalExceptionhandler : com.hrth.ustock.jwt.CustomLogoutFilter.doFilter(Cut
|> 2024-10-10 12:32:56.684 2024-10-10T12:32:56.496+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-1] c.h.u.e.common.GlobalExceptionhandler : com.hrth.ustock.jwt.CustomLogoutFilter.doFilter(Cut
|> 2024-10-10 12:32:56.684 2024-10-10T12:32:56.496+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-1] c.h.u.e.common.GlobalExceptionhandler : com.hrth.ustock.jwt.JWTfilter.doFilterInternal(JWTf
|> 2024-10-10 12:32:56.684 2024-10-10T12:32:56.496+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-1] c.h.u.e.common.GlobalExceptionhandler : com.hrth.ustock.controller.main.NewsController.myHC
|> 2024-10-10 12:32:56.684 2024-10-10T12:32:56.496+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-1] c.h.u.e.common.GlobalExceptionhandler : com.hrth.ustock.service.auth.CustomUserService.getU
|> 2024-10-10 12:32:56.684 2024-10-10T12:32:56.495+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-1] c.h.u.e.common.GlobalExceptionhandler : 해당 사용자를 찾을 수 없습니다.
|> 2024-10-10 12:30:00.152 2024-10-10T12:30:00.000+09:00 INFO 1 --- [ustock] [scheduled-task-pool-3] c.h.u.service.cron.StockCronService : 현재가 크론잡 시작
|> 2024-10-10 12:25:00.631 2024-10-10T12:25:00.433+09:00 INFO 1 --- [ustock] [scheduled-task-pool-3] c.h.u.service.cron.StockCronService : 현재가 크론잡 종료
|> 2024-10-10 12:25:00.138 2024-10-10T12:25:00.000+09:00 INFO 1 --- [ustock] [scheduled-task-pool-7] c.h.u.service.cron.StockCronService : 현재가 크론잡 시작
|> 2024-10-10 12:19:43.569 2024-10-10T12:19:43.441+09:00 INFO 1 --- [ustock] [scheduled-task-pool-1] c.h.u.service.cron.StockCronService : 현재가 크론잡 종료
|> 2024-10-10 12:15:00.894 2024-10-10T12:15:00.000+09:00 INFO 1 --- [ustock] [scheduled-task-pool-1] c.h.u.service.cron.StockCronService : 현재가 크론잡 시작
|> 2024-10-10 12:04:43.512 2024-10-10T12:04:43.448+09:00 INFO 1 --- [ustock] [scheduled-task-pool-2] c.h.u.service.cron.StockCronService : 현재가 크론잡 종료
|> 2024-10-10 12:00:00.830 2024-10-10T12:00:00.000+09:00 INFO 1 --- [ustock] [scheduled-task-pool-2] c.h.u.service.cron.StockCronService : 현재가 크론잡 시작
|> 2024-10-10 11:55:00.510 2024-10-10T11:55:00.443+09:00 INFO 1 --- [ustock] [scheduled-task-pool-4] c.h.u.service.cron.StockCronService : 현재가 크론잡 종료
|> 2024-10-10 11:55:00.809 2024-10-10T11:55:00.000+09:00 INFO 1 --- [ustock] [scheduled-task-pool-4] c.h.u.service.cron.StockCronService : 현재가 크론잡 시작
|> 2024-10-10 11:49:43.702 2024-10-10T11:49:43.475+09:00 INFO 1 --- [ustock] [scheduled-task-pool-1] c.h.u.service.cron.StockCronService : 현재가 크론잡 종료
|> 2024-10-10 11:45:00.236 2024-10-10T11:45:00.000+09:00 INFO 1 --- [ustock] [scheduled-task-pool-8] c.h.u.service.cron.StockCronService : 현재가 크론잡 시작
|> 2024-10-10 11:43:07.447 2024-10-10T11:43:07.299+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-2] c.h.u.e.common.GlobalExceptionHandler : com.hrth.ustock.jwt.CustomLogoutFilter.doFilter(Cut
|> 2024-10-10 11:43:07.447 2024-10-10T11:43:07.299+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-2] c.h.u.e.common.GlobalExceptionHandler : com.hrth.ustock.jwt.CustomLogoutFilter.doFilter(Cut
|> 2024-10-10 11:43:05.191 2024-10-10T11:43:05.148+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-8] c.h.u.e.common.GlobalExceptionHandler : com.hrth.ustock.jwt.JWTfilter.doFilterInternal(JWTf
|> 2024-10-10 11:43:05.191 2024-10-10T11:43:05.147+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-8] c.h.u.e.common.GlobalExceptionHandler : com.hrth.ustock.controller.main.NewsController.myHC
|> 2024-10-10 11:43:05.191 2024-10-10T11:43:05.147+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-8] c.h.u.e.common.GlobalExceptionHandler : com.hrth.ustock.service.auth.CustomUserService.getU
|> 2024-10-10 11:43:05.191 2024-10-10T11:43:05.147+09:00 INFO 1 --- [ustock] [http-nio-8080-exec-8] c.h.u.e.common.GlobalExceptionHandler : 해당 사용자를 찾을 수 없습니다.
```

## 상세한 서버 메트릭



# 04 수행결과

## 팀 협업 방식



**Figma**

프로젝트 기획

화면 설계



**GitHub**

Issue 관리

Github Project 활용  
FE, BE 진행상황 공유



**Swagger**

API 명세서 정리

간단한 명세서 확인



**Grafana**

간편한 로그 확인

메트릭을 확인하여  
최적화 진행



**자체 평가**

# 05 자체평가의견

잘한 점



## 원활한 커뮤니케이션

협업툴을 이용해서 진행상황 공유를 하고 원활한 협업을 도모

### 피그마

화면 ID	
페이지 제목	모의 투자 페이지
페이지 설명	하단바에서 모의투자 메인으로 들어온 경우
내용	<p>1 종목명 네임으로만 검색 가능</p> <ul style="list-style-type: none"><li>• 초성 x</li><li>• 검색 결과가 없는 경우 '검색 결과 없음'</li><li>• 예) 삼성 입력 시 삼성이 불</li><li>• 리스트 순위말고 현재 검색</li><li>• 검색 시 주식 코드 / 주식명</li></ul> <p>2 선택된 카테고리의 종목들만 표시</p> <p>선택 된 버튼 폰트 색상: #0000FF 선택 되지 않은 버튼 폰트 색상: #CCCCCC</p> <p>3 수익률/ 거래량/ 관심종목 버튼</p> <ul style="list-style-type: none"><li>• 종목을 누르면 해당 종목의 차트와 상세 정보 표시</li><li>• 등락율 소수점 2번째 자리 표시</li></ul> <p>4 사용자가 좋아요를 표시할 수 있는 버튼</p> <ul style="list-style-type: none"><li>• 사용자가 이미 매수했다면 눌렀을 때 fill #FF0000</li><li>• 누르지 않았을 때 #8F94A0</li></ul>

### 스웨거

**News** 뉴스 관련 API

**Portfolio** 포트폴리오 관련 API

**GET** /v1/portfolio 보유 포트폴리오 리

**POST** /v1/portfolio 포트폴리오 생성

**GET** /v1/portfolio/{pfId} 포트폴리오

**DELETE** /v1/portfolio/{pfId} 포트폴리오

**PUT** /v1/portfolio/{pfId}/hol

**POST** /v1/portfolio/{pfId}/hol

**DELETE** /v1/portfolio/{pfId}/hol

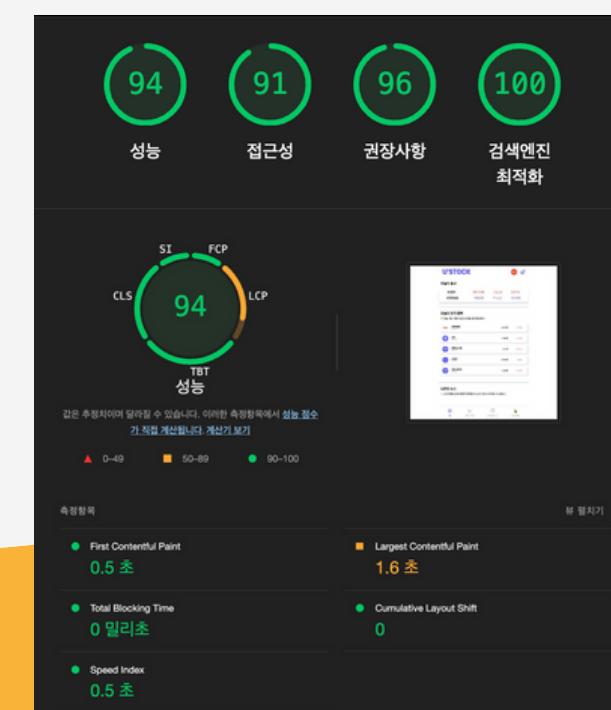
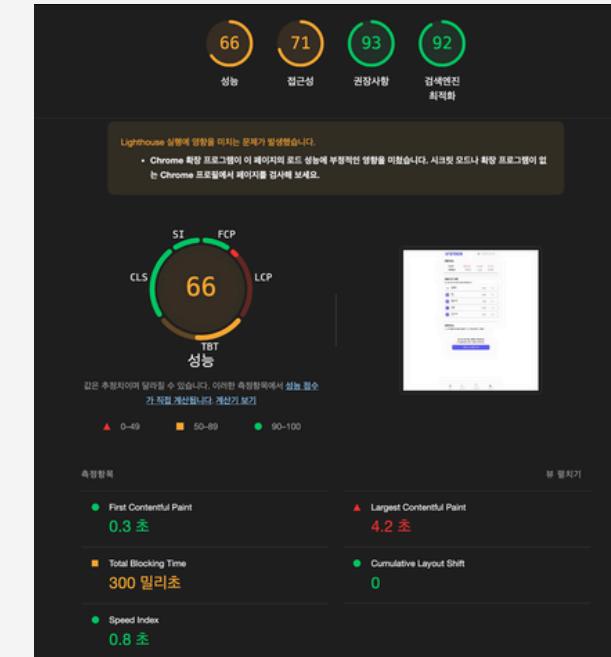
**PATCH** /v1/portfolio/{pfId}/hol

**GET** /v1/portfolio/check 포트폴리오



## 주도적인 문제 해결

필수 요구사항을 넘어서 개선 방안을 고민하고 반영



# 05 자체평가의견

잘한 점

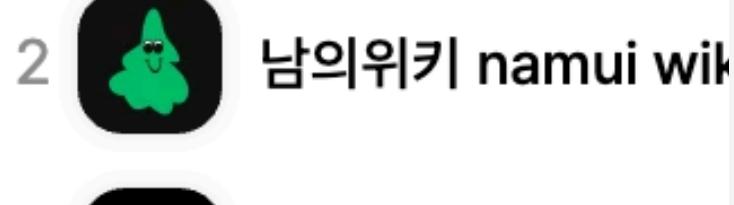


## 적극적인 서비스 홍보

디스콰이엇 PRODUCT OF THE WEEK 달성

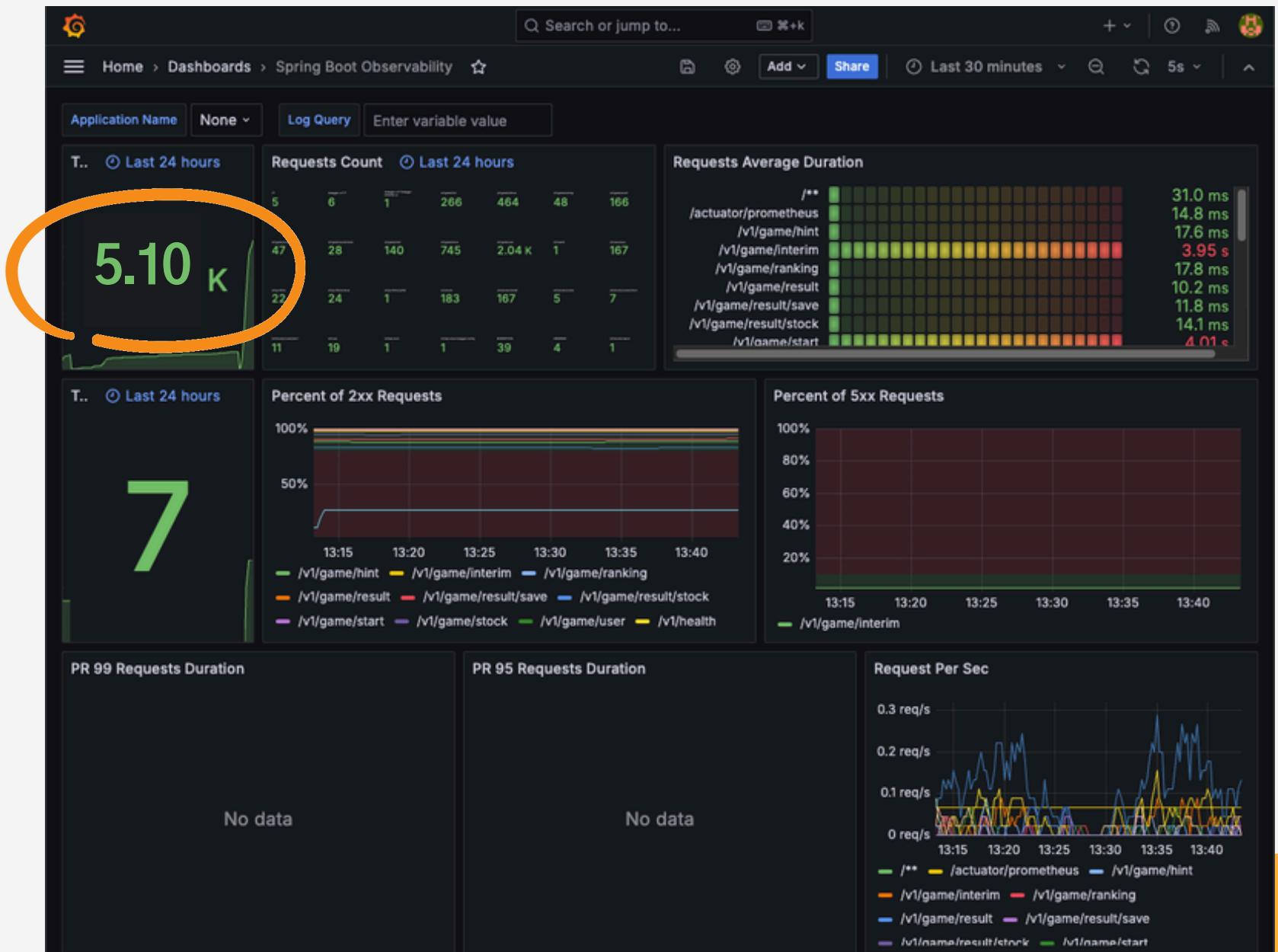
자체 게임 대회를 열어 사용자 참여 유도 + 피드백 수렴

🔥 트렌딩 프로덕트



## 서비스 모니터링 및 이슈 트래킹

지속적인 서비스 모니터링을 통해 빠른 이슈 대응, 사용자 경험 향상



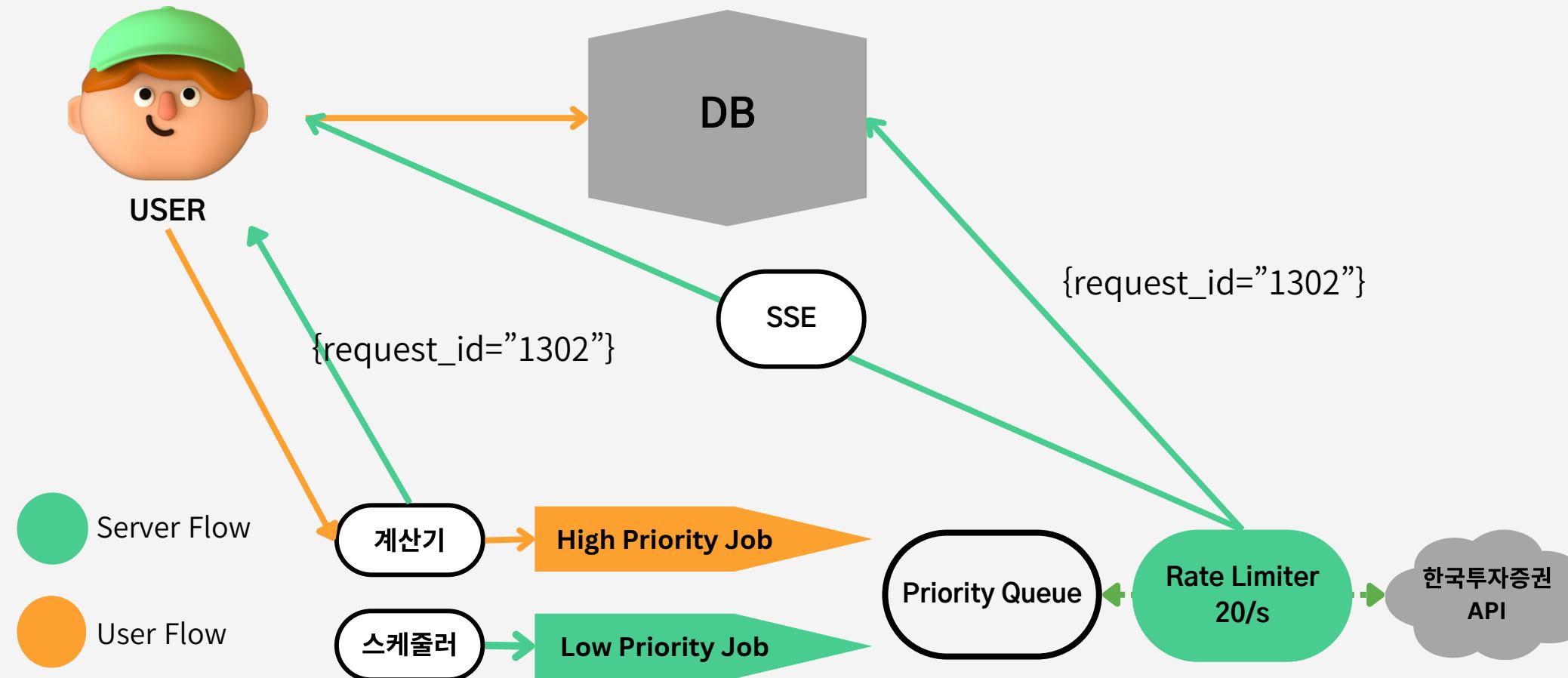
# 05 자체평가의견 아쉬운 점



## Rate Limiter 개선

요청 횟수를 최대로 활용하고, 사용자 요청을 더 안정적으로 처리할 수 있도록 개선할 예정

### TO-BE: Rate Limiter



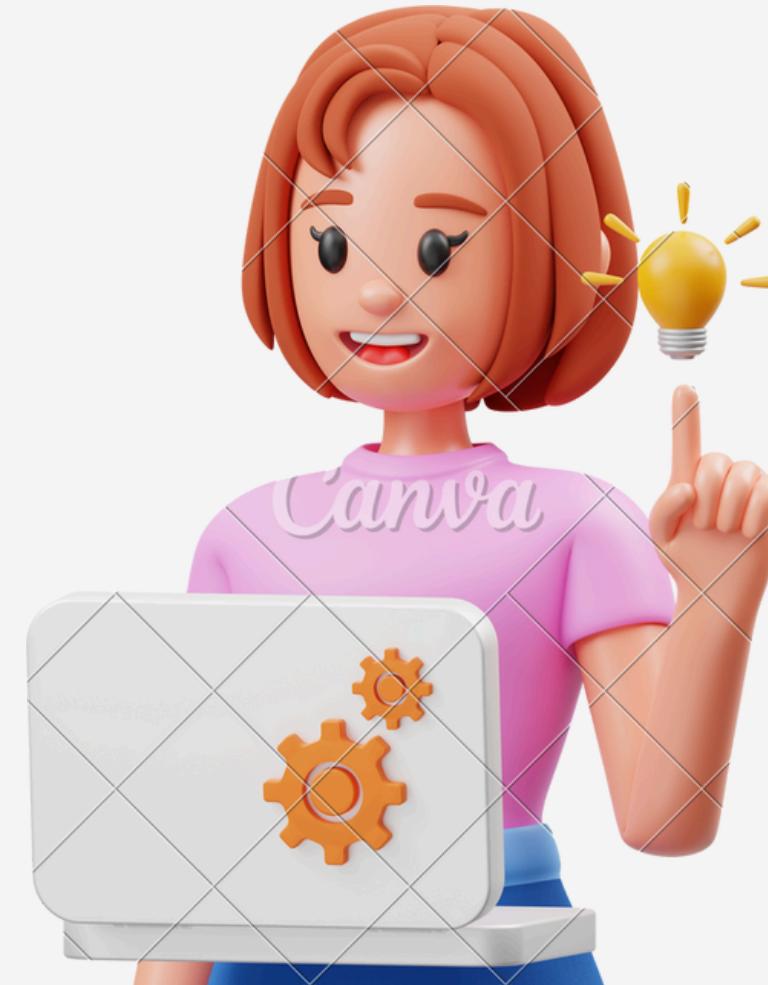
# 05 자체평가의견 아쉬운 점



## Grafana 알림 기능 추가

서버 에러가 발생하면 바로 인지할 수 있도록 알림 기능을 추가할 예정

### TO-BE: ✅Grafana Alert



# 05 자체평가의견 성과



가입유저 52명  
활동유저 199명  
(10월9일 기준)



시간당 트래픽  
5,000여건



보안이슈 X



디스코아이엇  
프로덕트

🔥 트렌딩 프로덕트

1	SPOT!
2	USTOCK

🔥 트렌딩 프로덕트

1	USTOCK 새로 나온
2	남의위키 namui wiki

게임, 교육, 금융  
ustock 창업자/경영진 @U'STOCK  
**USTOCK**  
스껄~하게 주식과 친해져볼까?  
0명의 리뷰  
0.0  
★★★★★  
DISCOUNT PRODUCT OF THE WEEK 1

감사합니다

'STOCK



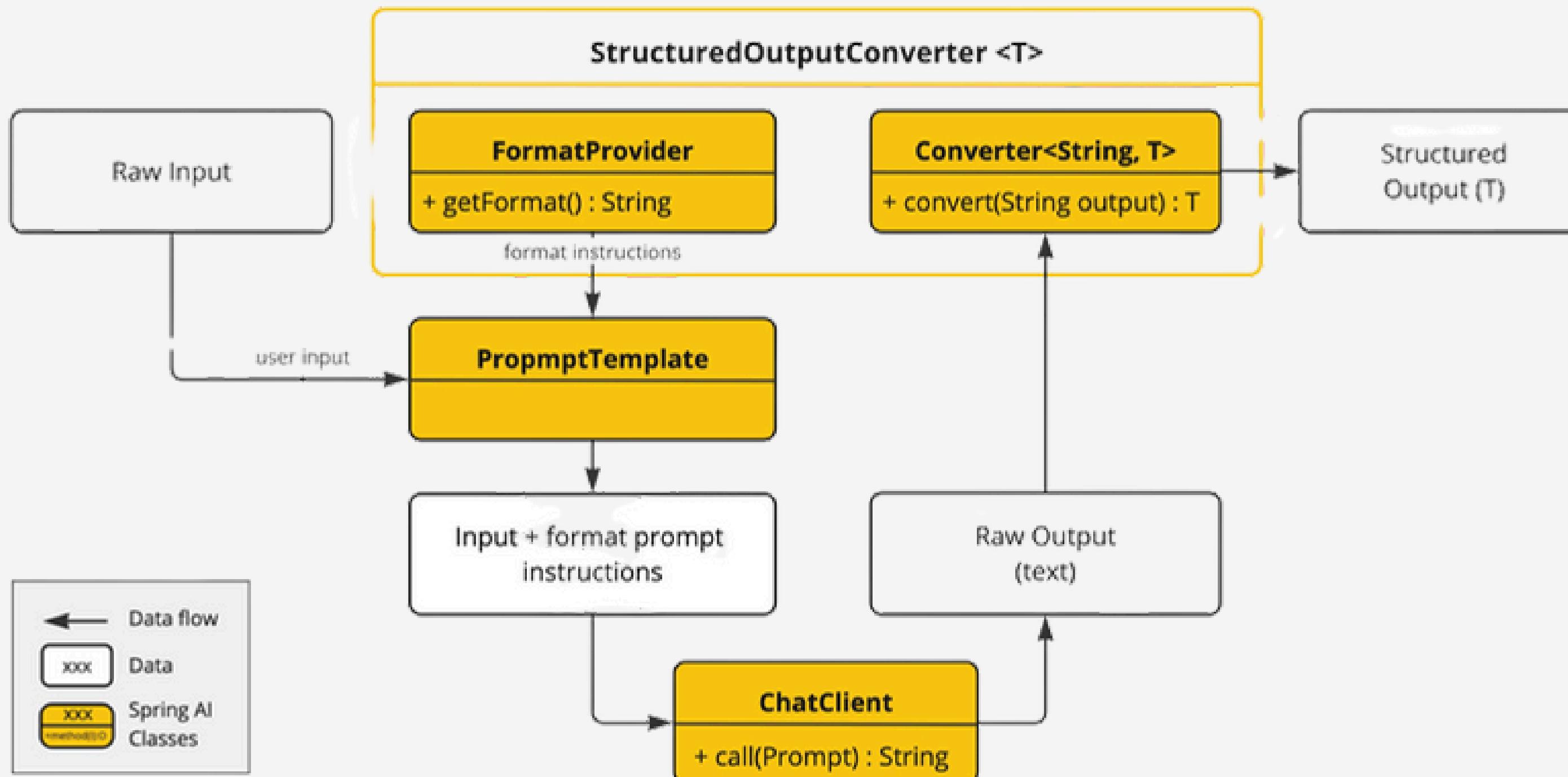


질의응답



HELPROADTOHEAVY

# Structured Output Converter





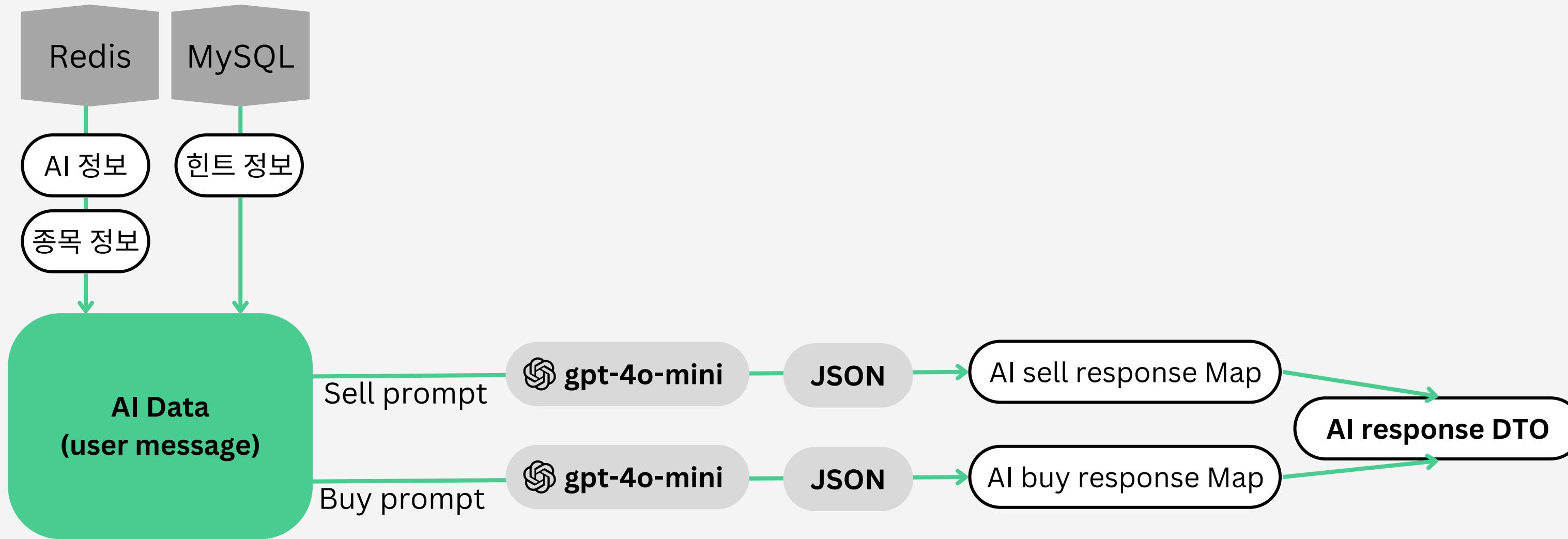
스껄~하게 주식과 친해져 볼까?



# 04 수행결과

## 키 포인트 3. 생성형 게임 AI 구현

### AS IS



### TO BE

로그 기록 & 조회기능 추가



## 05 자체평가의견 느낀 점



**기록을 체계적으로 해서 커뮤니케이션에 활용해야겠다**



**유저가 의도한 대로 움직이지 않아서,  
여러 경우의 수를 생각하고 유저의 입장을 한 번 더  
고려해서 개발해야겠다고 생각했다**



**실제로 배포를 해보니 서비스 자체를 개발하는 것 뿐만 아니라  
보안이나 로그 등 신경 쓸 게 많다는 걸 알았다**



