



시연

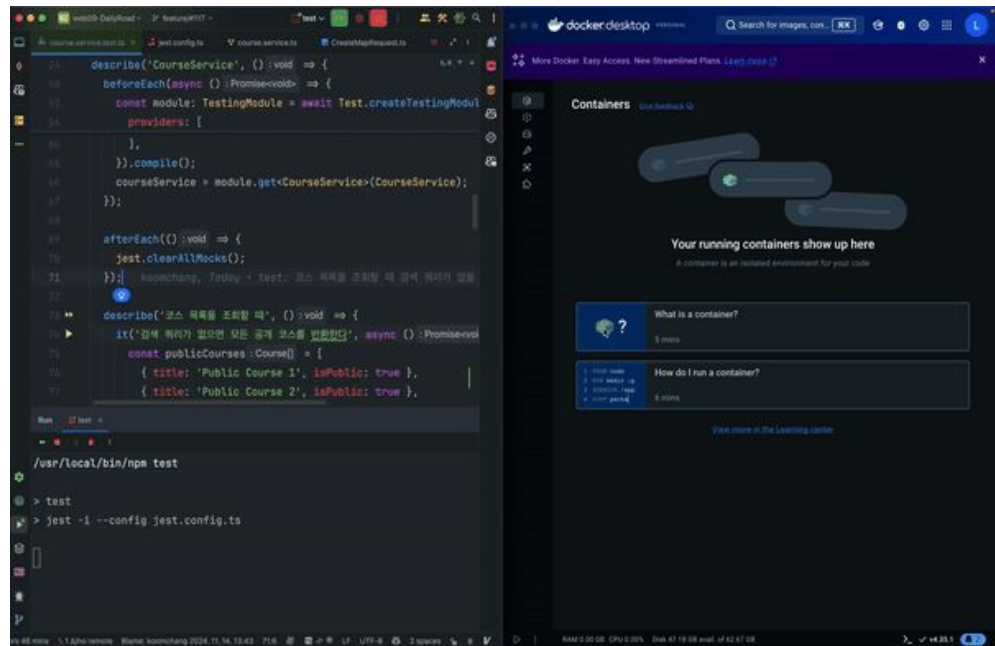
제목도 임팩트있게 가도 좋을듯
테스트 뭐시기 뭐시기

테스트 - 레이어별 테스트 대상

순서 흐름도 정해보면 좋을듯

테스트 - 백엔드 테스트 규칙

테스트 - 컨테이너 사용



컨테이너를 이용한
테스트

버그 수정

리팩토링 할 때 점검

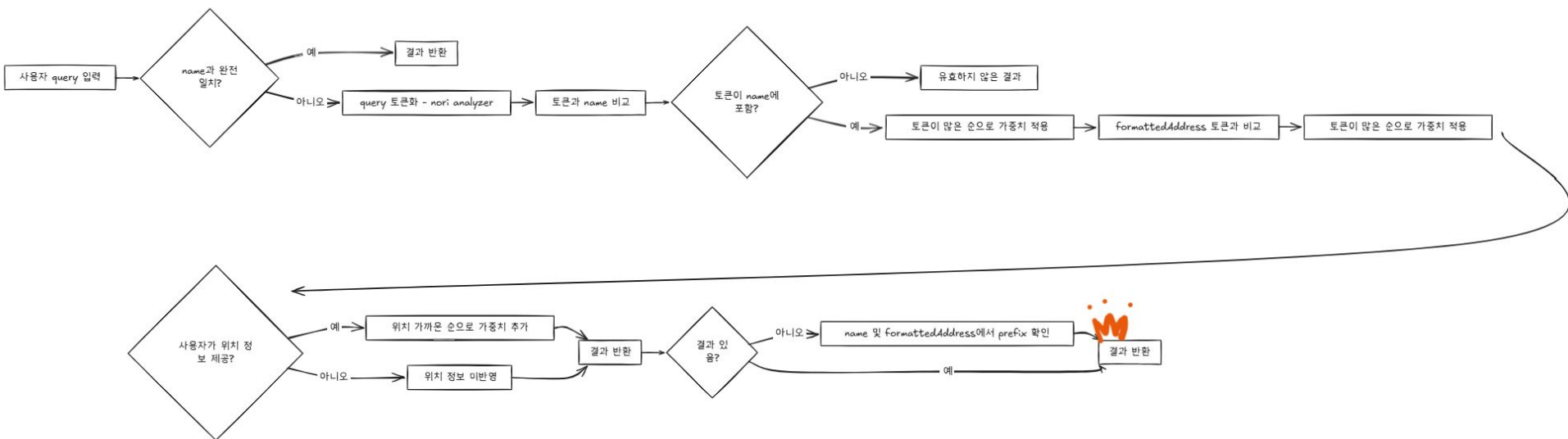
신뢰도

TDD의 장점에 대해서 생각해봄

Elasticsearch 도입기

검색 기능 개선 - 검색 로직 설계 (가중치 적용)

플로우 차트 피그마로 이쁘게 바꾸기



싱글 노드로 구성할 수 밖에 없던 배경
SFP 대처를 위한 방안 소개

스케줄러를 이용한 배치 전략
`need_sync` 이런거로 개선 가능할수도?
`boolean` 필드로 해서
마지막 주에 시간 테스트 해서 개선되면
널자

이것도 가능하면 낼 ㄱㄱ

검색 기능 개선 - 결과

사진 결과

1. 오타처리
2. 동의어 처리
3. 토큰 검색
4. 위도 경도 검색 -> 이걸 테스트 필요.
널 테스트 ㅋㅋ

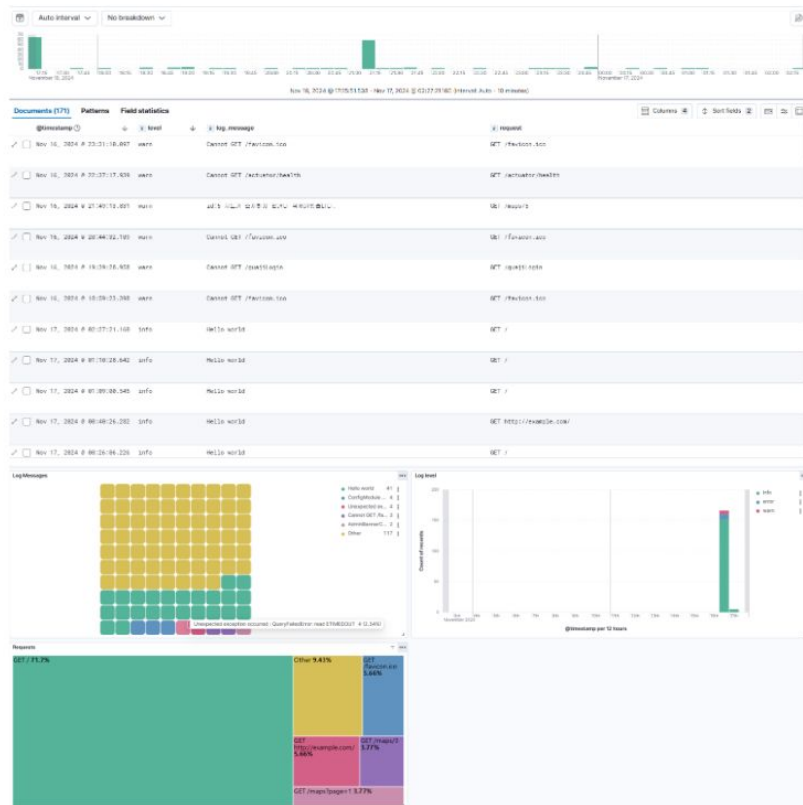
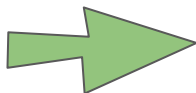
부록 - ES 이용했을 때 성능 문제는 없는지?

부록은 다 뒤로 뺄거임

오늘의강

에러 발생시 로그를 서버에 접속해서 확인해야 했음

```
ubuntu@daily-road:~$ docker logs nest
node:events:497
    throw er; // Unhandled 'error' event
    ^
Error: read ETIMEDOUT
    at TCP.onStreamRead (node:internal/stream_base_commons:218:20)
    at TCP.callbackTrampoline (node:internal/async_hooks:130:17)
Emitted 'error' event on Socket instance at:
    at emitErrorNT (node:internal/stream_base_commons:169:8)
    at emitErrorCloseNT (node:internal/stream_base_commons:128:3)
    at process.processTicksAndRejections (node:internal/process/task_queues:82:21) {
  errno: -110,
  code: 'ETIMEDOUT',
  syscall: 'read'
}
Node.js v20.18.0
```



로깅 및 모니터링 - Why



간단하게 설명 아니면 뒤로 빼서 합쳐도댐
설명할거면 기존 라이브러리와 차이점 -> 이래서
선택함. 요런느낌

로깅 및 모니터링 - 로깅 레벨 결정 기준

warn error 레벨 결정 기준
그에 따른 처리

누가 프로메테우스 이런거 왜 안췄어요?
물어볼것도 대비하면 좋을듯함
Slack webhook관련 이야기

모니터링 대상

로깅 및 모니터링 - 결과

키바나, 슬랙 사진

가능하다면! 로깅을 통해 어떤 것이 좋아졌는지

체감했다! 이야기

ex) 서버내려갔을 때 로그 못봐서 힘들었다

부록 - 로깅에서 키바나 필터 적용

마커 클러스터링 최적화 - Why

마커 클러스터링 최적화 - 발생한 문제

깜빡 거림

너무 잦은 클러스터 마커 렌더링

-> 둘다 재현 가능한 사진 및 gif 있음 좋음, 영상도

마커 클러스터링 최적화 - 해결 방법

마커 렌더링 시점 및 조건 변경

마커 클러스터링 최적화 - 결과

효과를 수치적으로 측정해볼 수 있음 더 좋을듯
안대면 그냥 깜빡 거림 사라짐?

위에 주제 이름 깔삼하게 정해보기

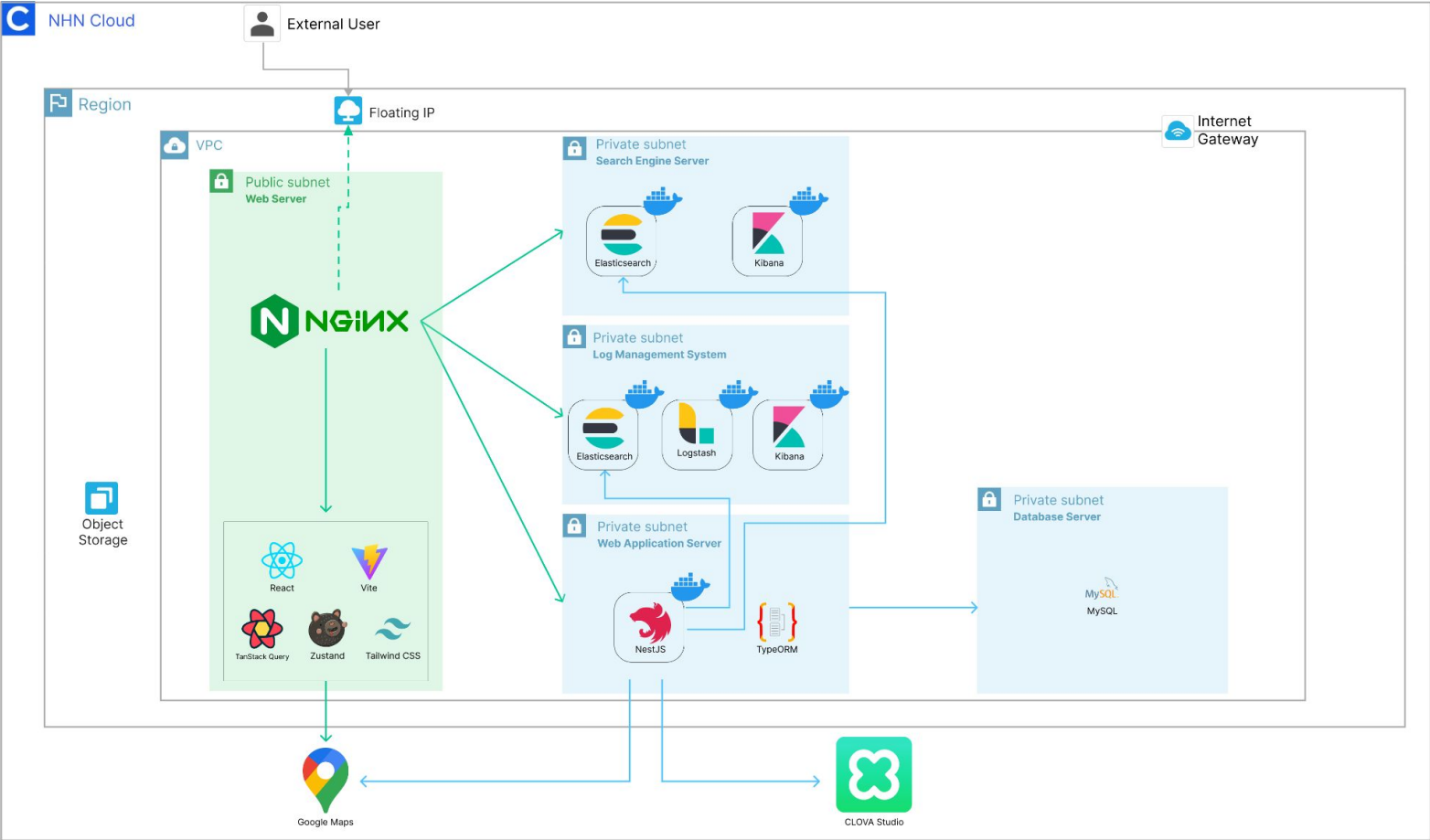
클로스터 도입 과정에서 생긴 마커 사라짐 현상
요것도 사진이나 영상 있음 굳

자유도 높음 -> 마커 관리 주체 분산 -> 마커 사라짐
현상 해결

코드 작성 규칙 설정

부록 - 서비스 아키텍처

클로바 지우고 아키텍처 내용 재구성하여 수정



부록 - 코드 리뷰

0 Open ✓ 24 Closed Author Label Projects Milestones Reviews Assignee Sort

TestContainer를 이용한 Test 환경 구성 ✓ BE 33 테스트 7 설정

#71 by koomchang was merged 3 hours ago · Approved week2

지도/코스에 핀 추가 API 구현 ✓ BE 2 리팩터링 2 기능추가 1 버그

#67 by Miensoap was merged 3 hours ago · Approved week2

구글 소셜 로그인을 지원하는 유저, Auth 모듈 구현 ✓ BE 2 기능추가

#64 by hyohyt12 was merged 6 hours ago · Approved week2

타입 분리, 리턴 인라인화, fetch api 를 then 메소드 체이닝 사용, UserRepository 의 SoftDeleteRepository

#63 by hyohyt12 was closed yesterday

PlaceRepository 테스트 작성 (인텔리전스이러바 다투 남겨주세요) ✓ BE 33 테스트

#61 by koomchang was closed 11 hours ago · Draft week2

코스 생성/삭제/조회/검색/수정 API 구현 ✓ BE 2 리팩터링 2 기능추가

#60 by Miensoap was merged 6 hours ago · Approved week2

PlaceRepository에서 Repository 대신 SoftDeleteRepository로 리팩토링 ✓ BE 2 리팩터링 2 기능

#59 by koomchang was merged yesterday · Approved week2

구글 소셜 로그인을 지원하는 유저, Auth 모듈 구현 ✓ BE 2 기능추가

#58 by hyohyt12 was closed yesterday · Changes requested week2

지도 정보 / 공개범위 수정 ✓ BE 2 리팩터링 2 기능추가

#53 by Miensoap was merged yesterday · Approved week2

Axis 초기 세팅, FE 배포 수정, 지도 생성 Form 구현 ✓ FE 2 기능추가 1 버그

#49 by HTHd was merged 2 days ago · Approved week2

koomchang 3 days ago

p2: nullable: true 이면 profileImage?: string; 으로 들어가는게 좋을 것 같습니다

Miensoap 3 days ago

수정했습니다!

<https://stackoverflow.com/questions/64635617/how-to-set-a-nullable-database-field-to-nullable>

! 키워드를 사용하는 사례도 있네요

Reply...

Resolve conversation

backend/resources/sql/DDL.sql Outdated

```
16 + CREATE TABLE PLACE
17 + (
18 +     id INT PRIMARY KEY AUTO_INCREMENT,
19 +     google_place_id VARCHAR(255) UNIQUE NOT NULL,
```

koomchang 3 days ago · edited

p4: 고정된 크기의 문자열인 경우 VARCHAR 대신 CHAR 을 사용하면 읽기 작업에서 더 빠른 성능을 낼 수 있습니다.

[Understanding CHAR vs VARCHAR in MySQL: A Comprehensive Guide](#)

Miensoap 3 days ago

google place id 의 경우 보통 27~35자 사이의 문자열인 것으로 확인해, (CHAR(58)) 으로 수정했습니다.

1

1

1

1

16 + if (existingUser) {

Miensoap 8 hours ago Member ...

pt. findbyProviderAndauthId 에서 항상 위치를 반환
(provider, oauth을 기준으로 upsert로 저장하고 정보 반환) 하면
아기서 보기 처리가 필요 없을 것 같아요.

```
Repository<User>.upsert(
    entityOrEntities: _QueryDeepPartialEntity<User> | _QueryDeepPartialEntity<User>[],
    conflictPathsOrOptions: string[] | UpsertOptions<User>);
Promise<InsertResult>
```

Inserts a given entity into the database, unless a unique constraint conflicts then updates the entity Unlike save method executes a primitive operation without cascades, relations and other operations included. Executes fast and efficient INSERT ... ON CONFLICT DO UPDATE/ON DUPLICATE KEY UPDATE query.

해당 로직을 표현하는 이름이면 더 좋겠네요
InsertResult 타입 확인해 보시고, find 있어 자기가 꺼낼 수 있다면 사용

hyohyt12 7 hours ago Member / Author ...

upsert 를 사용하지 않았던 이유가 처음에 member 로 role 을 정하는데 만약에 admin 이라는 role 을 가진다면 로그인 할 때마다 role 이 member 로 번하기에 선택하긴 했습니다. 다른 좋은 방법이 있을까요? upsert 를 사용하면서 role 이 로그인할 때 번하게 되는...

Miensoap 7 hours ago

아하 다른 다른 문제가 있었군요
어드민 토큰은 따로 발급하는 등...

그럼 일단 두고 더 이야기해 보아요

backend/src/place/place.service.ts Outdated

```
... ... @@ -35,7 +33,7 @@ export class PlaceService {
35 33 place.detailPageUrl = detailPageUrl;
36 34
37 35 if (await this.placeRepository.findby(googlePlaceId(googlePlaceId))) {
38 - throw new ConflictException("이미 등록된 장소입니다.");
36 + throw new PlaceException(PlaceExceptionType.ALREADY_EXISTS);
```

Miensoap 2 days ago Member ...

q. 모두 PlaceException 과 같이 도메인별로 하나의 예외 타입을 사용하나요?

NotfoundException -> PlaceNotfoundException
구조와 비교했을 때 장점과 단점은 어떤 것이 있을까요?

koomchang 2 days ago Member / Author ...

네 도메인별로 하나의 예외 타입을 사용하요 exceptionType 에서 구체적인 예외처리를 정하는 것으로 생각했습니다.

말씀하신대 첫 처럼 도메인별로 다양한 예외타입을 사용하는 것도 하나의 방법이라고 생각해요. 더 구체적인 예외타입을 보여줄 수 있을 것 같습니다. 다만, exceptionType 객체를 정의하는 곳이 한 곳에 모여있지 않을 것 같다는 단점도 있을 것 같네요!

민서님은 어떤 방식이 더 좋을 것 같다고 생각하나요?

Miensoap 2 days ago Member ...

파일 관리 면에서는 코드를 한 데 모아 관리하게 되니 훨씬 깔끔할 것 같고,

남은 건 PlaceException(PlaceExceptionType.ALREADY_EXISTS) vs PlaceNotFoundException() 형식의 가독성 문제 같아요!

저는 후자가 깔끔해 보이긴 하는데, 취향 차이인 것 같습니다

koomchang 2 days ago Member / Author ...

취향 차이인 것 같긴한데 고민되는 것 같네요. 확실히 말씀하게 가독성이 좋아 보이긴 합니다!

Reply...

Resolve conversation

q : 질문

p1 : 꼭 반영해주세요

p2 : 반영해도 좋고 안 해도 괜찮아요

p3 : 사소한 의견입니다

[코드 리뷰 in 백스쿼드 개발 문화](#)

부록 - Slack Scheduler

11월 26일 화요일 ▾



Slackbot 오후 9:50

리마인더: 데일리 스크럼 10분 전입니다. 노션 스크럼 페이지를 미리 작성해주세요!

어제 ▾



Slackbot 오후 12:50

리마인더: 마스터 클래스 10분 전입니다.



Slackbot 오후 9:50

리마인더: 데일리 스크럼 10분 전입니다. 노션 스크럼 페이지를 미리 작성해주세요!

오늘 ▾



Slackbot 오후 7:00

리마인더: 금요일은 데모 발표 날입니다! 오후 9시부터 11시까지 발표 자료를 준비하는 시간입니다!



Slackbot 오후 9:50

리마인더: 데일리 스크럼 10분 전입니다. 노션 스크럼 페이지를 미리 작성해주세요!

부록 - 팀 목표 및 달성 정도

Q&A