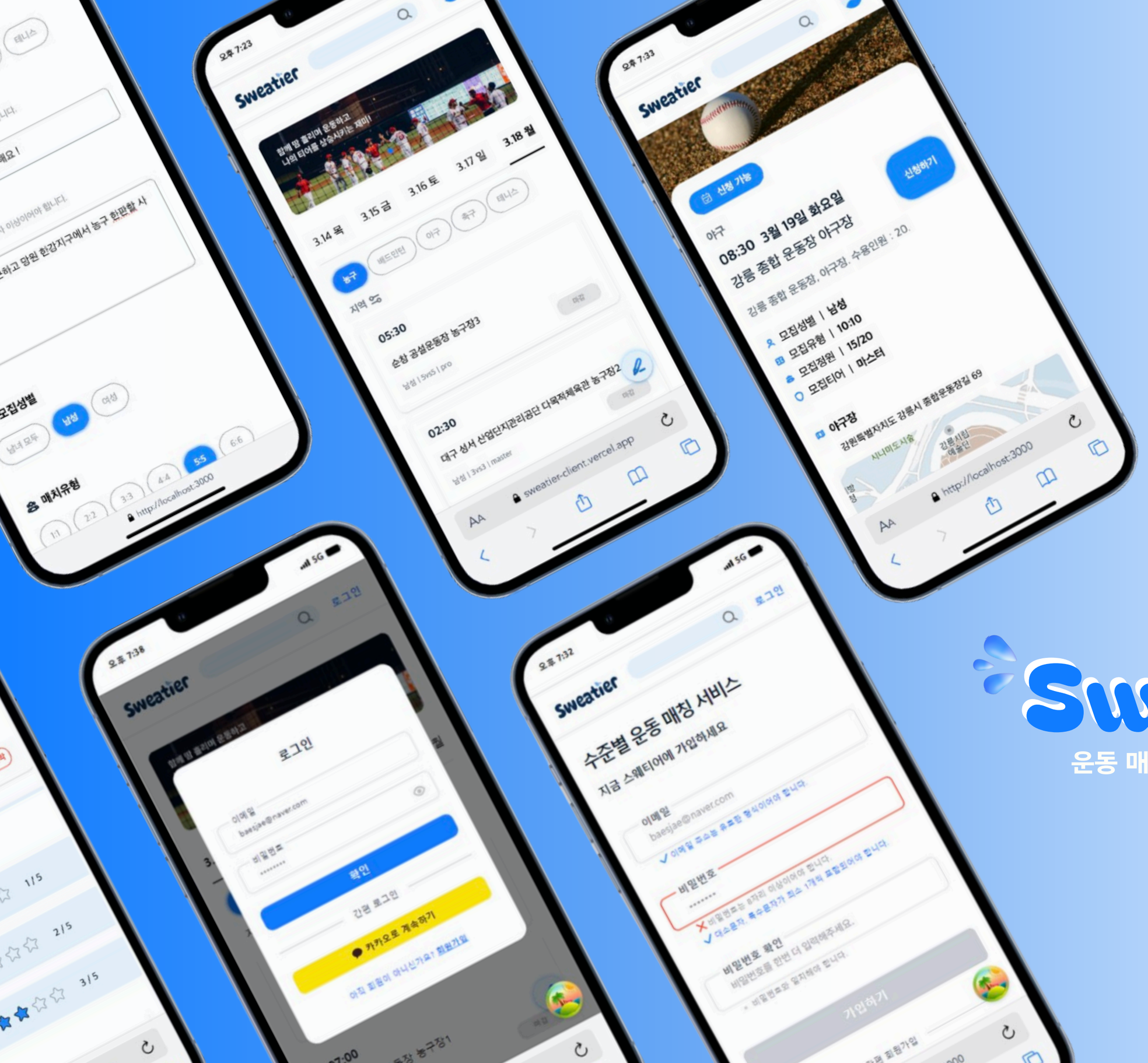

Sweatier
운동 매치 · 티어 평가 서비스

프로젝트 소개

스웨티어 설명

- 수준별 운동 매칭을 통해 실력이 비슷한 사람들과 운동을 즐기고
- 사용자의 운동 수준(티어)을 평가받을 수 있는 서비스입니다.

주요 기능설명

- 선호하는 날짜, 시간, 성별, 지역, 종목에 따른 경기 참여
- **티어**에 따른 수준별 경기 매칭 시스템
- 자신이 선호하는 컨디션의 경기를 직접 호스팅
- 경기 후 상호 평가를 통한 **티어 조정 시스템**
- 티어 관리를 통한 운동 참여자들의 동반 성장 도모

서비스 소개 - 핵심기능

👏 참여자 상호간 티어 평가

✨ 다양한 종목별 티어 산출 및 경기 신청 가능

유형	설명
사용자	일반 사용자
유저	가입한 사용자

티어	티어	승급조건	설명
비기너	beginner	경기 3회 이상	비기너 졸업 시 최소 등급은 아마추어
아마추어	amateur	~100%	세미프로로 승급 조건
세미프로	semi-pro	~60%	프로로 승급 조건
프로	pro	~30%	마스터로 승급 조건
마스터	master	~10%	

종목	종목
테니스	tennis
축구	soccer
농구	basketball
야구	baseball
배드민턴	badminton

1. 배치고사: 첫 티어 배정은 무조건 비기너 (경기 3회 이상 시 티어 재조정)
2. 별점 시스템은 유지
3. 티어는 전체 유저에서 %별로 부여 (백분율) → 매일 오전 12:00마다 배치

마스터 : 10%
프로 : 20%
세미프로 : 30%
아마추어 : 40%

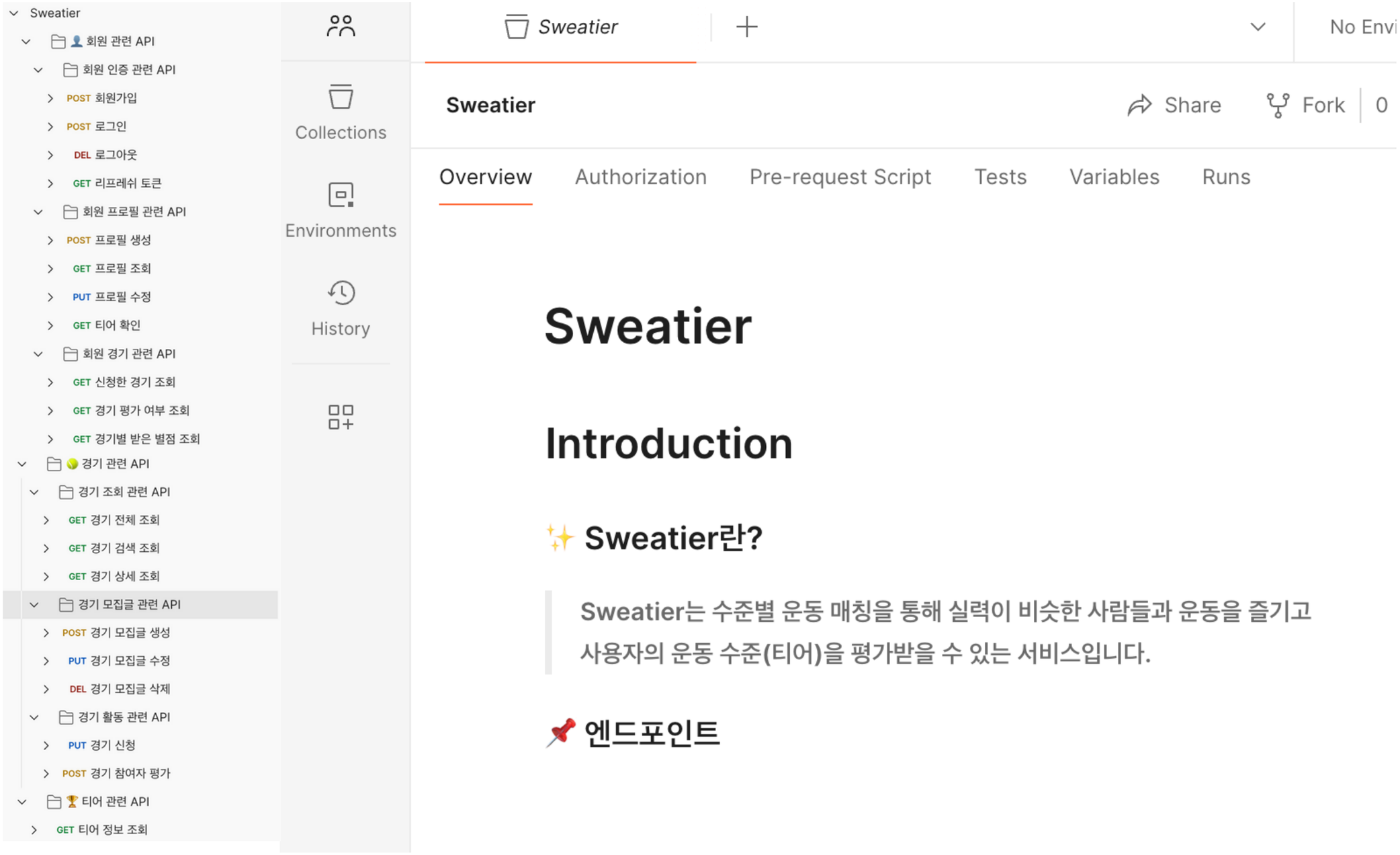
사용기술 및 솔루션

기술 스택

- Nest.js + Typescript
- Prisma (Node.js ORM)
- Postgresql

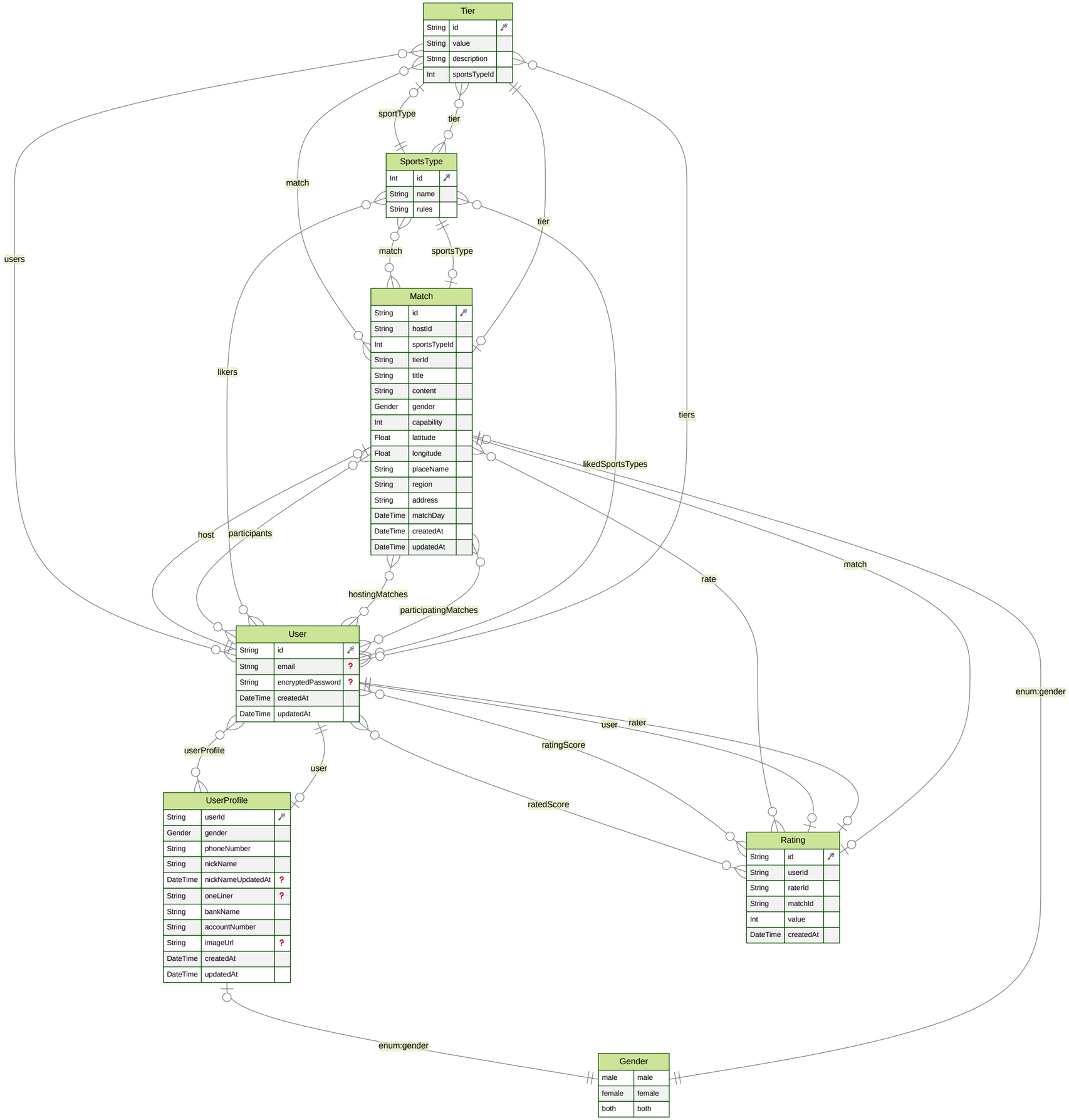
- **인프라**
- Github Actions
- AWS S3
- AWS ECR
- AWS Code Deploy
- Docker
- AWS Route 53
- NGINX

API 명세서

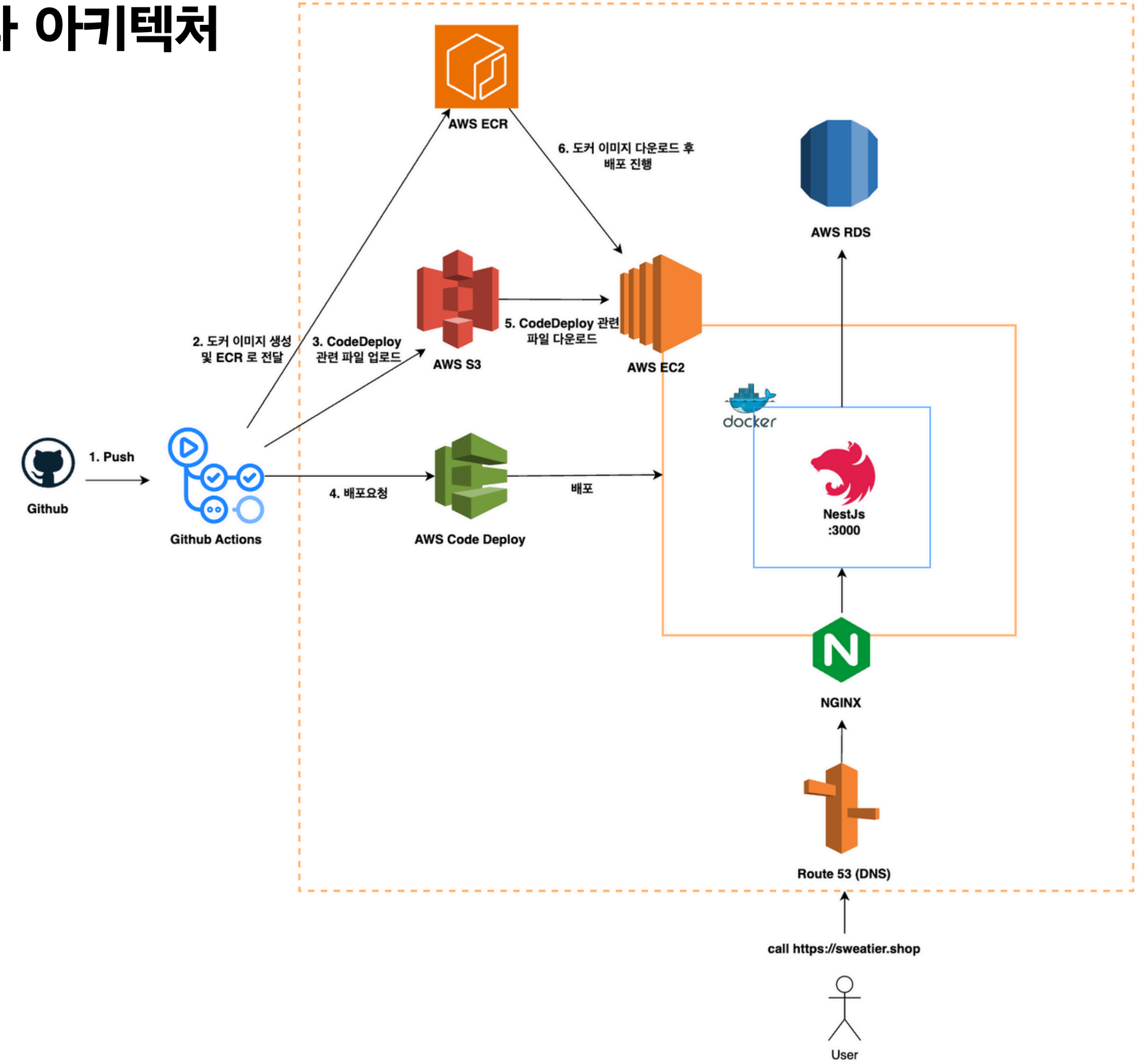


https://documenter.getpostman.com/view/32959422/2sA2xpRUCX#599c3aaf-e821-498f-aa36-ca94ec9e2b9f

ERD


Tier
String id
String value
String description
Int sportsTypeId
sportType
tier
SportsType
Int id
String name
String rules
match
tier
match
sportsType
users
Match
String id
String hostId
Int sportsTypeId
String tierId
String title
String content
Gender gender
Int capability
Float latitude
Float longitude
String placeName
String region
String address
DateTime matchDay
DateTime createdAt
DateTime updatedAt
likers
tiers
likedSportsTypes
host
participants
rate
match
hostingMatches
participatingMatches
User
String id
String email ?
String encryptedPassword ?
DateTime createdAt
DateTime updatedAt
enum:gender
user
rater
ratingScore
userProfile
user
ratedScore
UserProfile
String userId
Gender gender
String phoneNumber
String nickName
DateTime nickNameUpdatedAt ?
String oneLiner ?
String bankName
String accountNumber
String imageUrl ?
DateTime createdAt
DateTime updatedAt
Rating
String id
String userId
String raterId
String matchId
Int value
DateTime createdAt
enum:gender
Gender
male male
female female
both both

인프라 아키텍처


AWS ECR
6. 도커 이미지 다운로드 후
배포 진행
AWS RDS
2. 도커 이미지 생성
및 ECR 로 전달
3. CodeDeploy
관련 파일 업로드
AWS S3
5. CodeDeploy 관련
파일 다운로드
AWS EC2
docker
1. Push
Github
Github Actions
4. 배포요청
AWS Code Deploy
배포
NestJs
:3000
NGINX
Route 53 (DNS)
call https://sweatier.shop
User

담당업무

USER 및 MATCH API 구현

- 유저 회원가입,로그인, 티어, 프로필 등 유저 관련 API 구현
- 매치 필터링, 피어 평가 로직 등 코드 다수 리팩토링

더미 시드데이터 생성 로직 구현

- 연관관계가 복잡하게 얽혀있는 테이블에 따른 시드 데이터 생성 로직 구현

인프라 아키텍처 설계

- 무료 크레딧 종료에 따라 기존 GCP 기반 인프라 구조에서 AWS 서비스로 마이그레이션
- Github actions 와 CodeDeploy 를 사용해 배포 자동화가 가능하게 설계

타임존 이슈: Prisma ORM 의 고질병

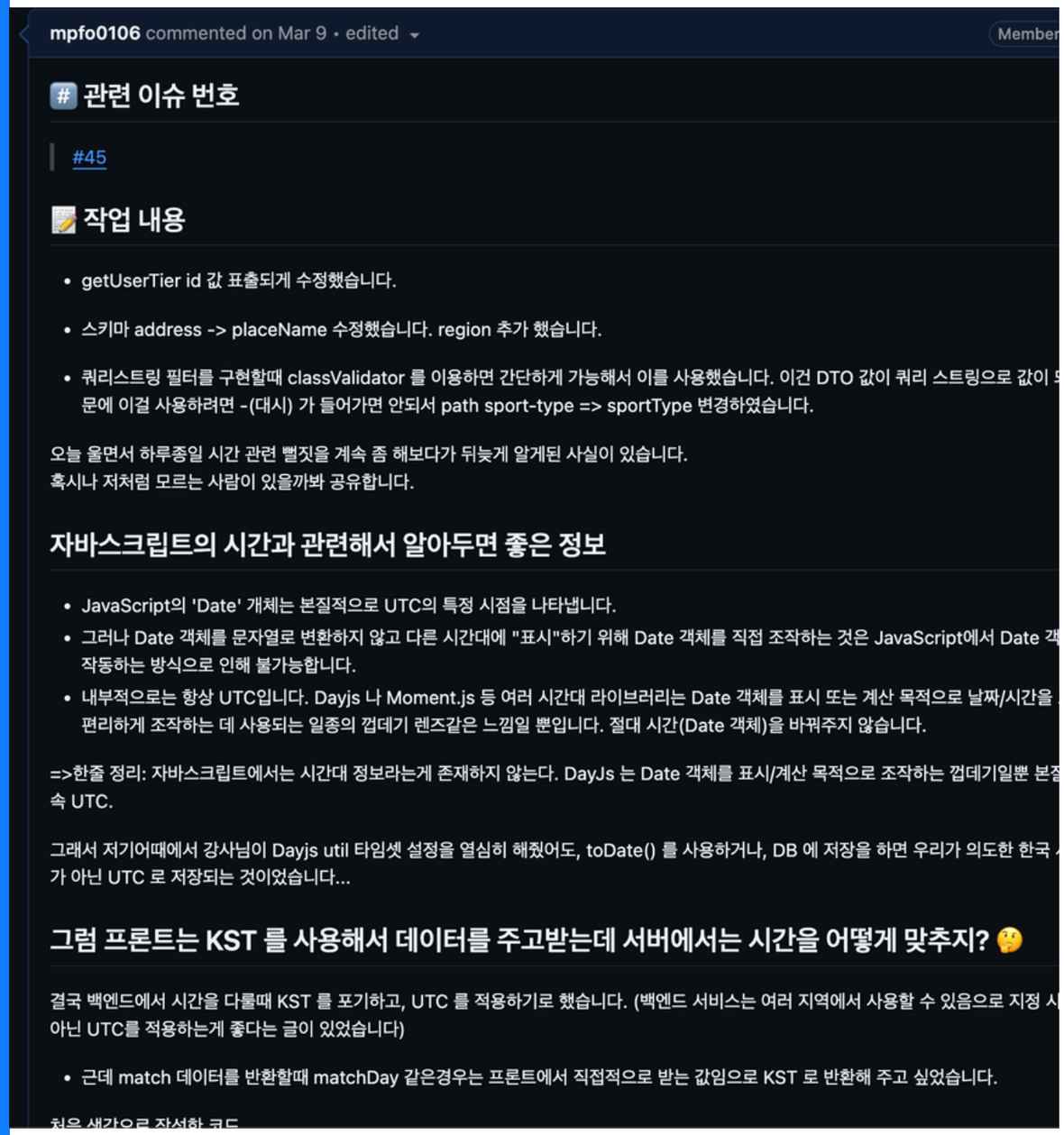
문제상황

- 2020년부터 이슈가 제기되었으나, 2024년 기준 여전히 타임존을 지원하지 않음..
- 이에 따라 모든 날짜는 데이터베이스 유형에 상관없이 UTC로 저장되며,
  클라이언트에서 읽을 때 Date 객체는 자동으로 UTC로 변환됨.
    - DB의 타임존을 변경해도 KST 타임존 정보를 저장할 수 없음.

해결방법

- 최종적으로 Prisma Client에서 KST 변환 미들웨어를 구현
- 데이터는 UTC로 저장되지만 클라이언트에 KST로 변환된 데이터를 제공하도록 설정.

팀원 및 Prisma 공식 이슈에 문제 상황 및 해결 공유



mpfo0106 commented on Mar 9 • edited

Member

관련 이슈 번호

#45

📝 작업 내용

- getUserTier id 값 표출되게 수정했습니다.
- 스키마 address -> placeName 수정했습니다. region 추가 했습니다.
- 쿼리스트링 필터를 구현할때 classValidator 를 이용하면 간단하게 가능해서 이를 사용했습니다. 이건 DTO 값이 쿼리 스트링으로 값이 ㅇ
  문에 이걸 사용하려면 -(대시) 가 들어가면 안되서 path sport-type => sportType 변경하였습니다.

오늘 울면서 하루종일 시간 관련 뻘짓을 계속 좀 해보다가 뒤늦게 알게된 사실이 있습니다.
혹시나 저처럼 모르는 사람이 있을까봐 공유합니다.

자바스크립트의 시간과 관련해서 알아두면 좋은 정보

- JavaScript의 'Date' 개체는 본질적으로 UTC의 특정 시점을 나타냅니다.
- 그러나 Date 객체를 문자열로 변환하지 않고 다른 시간대에 "표시"하기 위해 Date 객체를 직접 조작하는 것은 JavaScript에서 Date 객
  작동하는 방식으로 인해 불가능합니다.
- 내부적으로는 항상 UTC입니다. Dayjs 나 Moment.js 등 여러 시간대 라이브러리는 Date 객체를 표시 또는 계산 목적으로 날짜/시간을
  편리하게 조작하는 데 사용되는 일종의 껍데기 렌즈같은 느낌일 뿐입니다. 절대 시간(Date 객체)을 바꿔주지 않습니다.

=>한줄 정리: 자바스크립트에서는 시간대 정보라는게 존재하지 않는다. DayJs 는 Date 객체를 표시/계산 목적으로 조작하는 껍데기일뿐 본질
속 UTC.

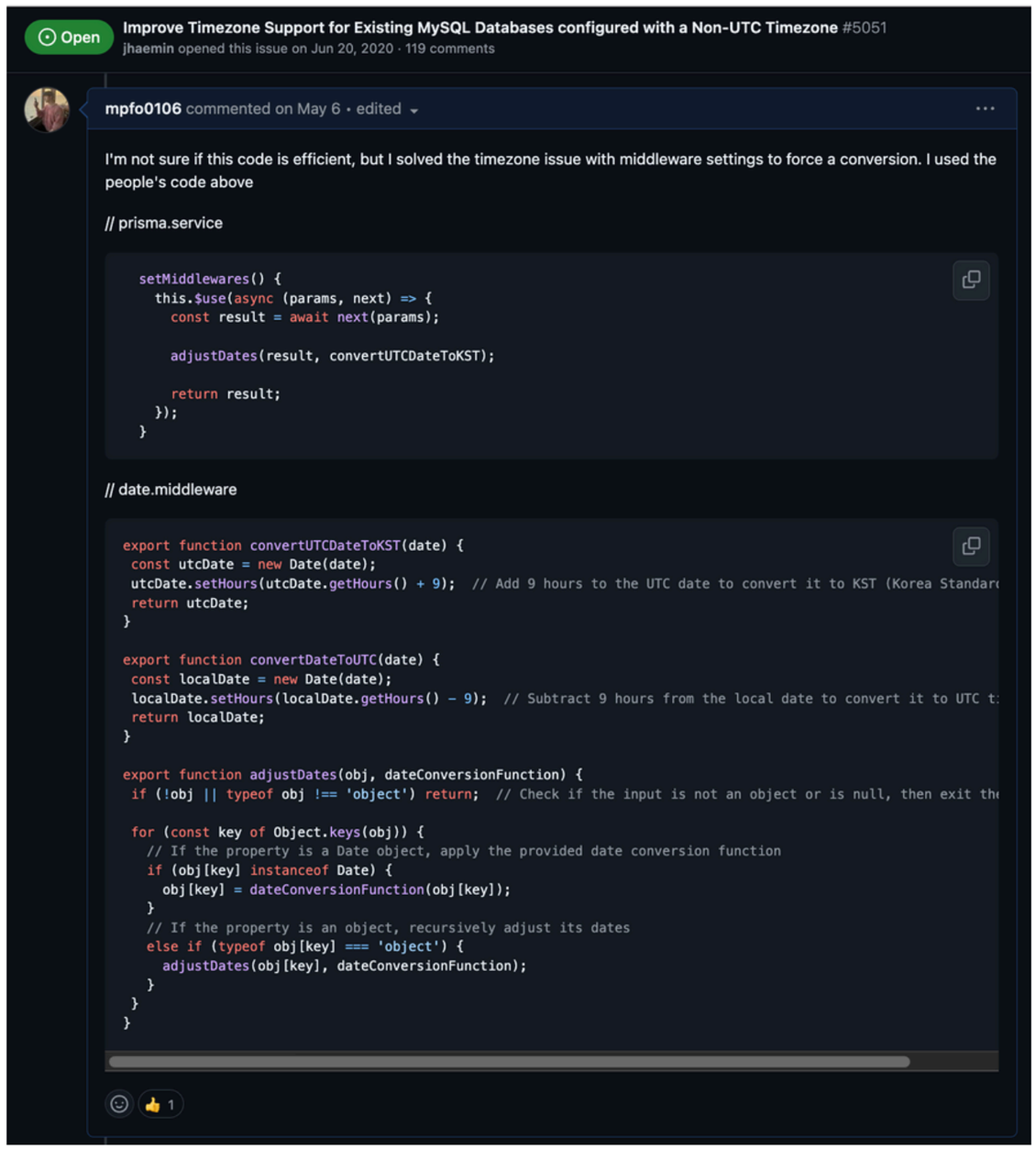
그래서 저기어때에서 강사님이 Dayjs util 타임셋 설정을 열심히 해줬어도, toDate() 를 사용하거나, DB 에 저장을 하면 우리가 의도한 한국
가 아닌 UTC 로 저장되는 것이었습니다...

그럼 프론트는 KST 를 사용해서 데이터를 주고받는데 서버에서는 시간을 어떻게 맞추지? 🤔

결국 백엔드에서 시간을 다룰때 KST 를 포기하고, UTC 를 적용하기로 했습니다. (백엔드 서비스는 여러 지역에서 사용할 수 있음으로 지정 시
아닌 UTC를 적용하는게 좋다는 글이 있었습니다)

- 근데 match 데이터를 반환할때 matchDay 같은경우는 프론트에서 직접적으로 받는 값임으로 KST 로 반환해 주고 싶었습니다.

처음 생각으로 작성한 코드



Open **Improve Timezone Support for Existing MySQL Databases configured with a Non-UTC Timezone** #5051
jhaemin opened this issue on Jun 20, 2020 · 119 comments

mpfo0106 commented on May 6 • edited

I'm not sure if this code is efficient, but I solved the timezone issue with middleware settings to force a conversion. I used the people's code above

// prisma.service

```
setMiddlewares() {
  this.$use(async (params, next) => {
    const result = await next(params);

    adjustDates(result, convertUTCDateToKST);

    return result;
  });
}
```

// date.middleware

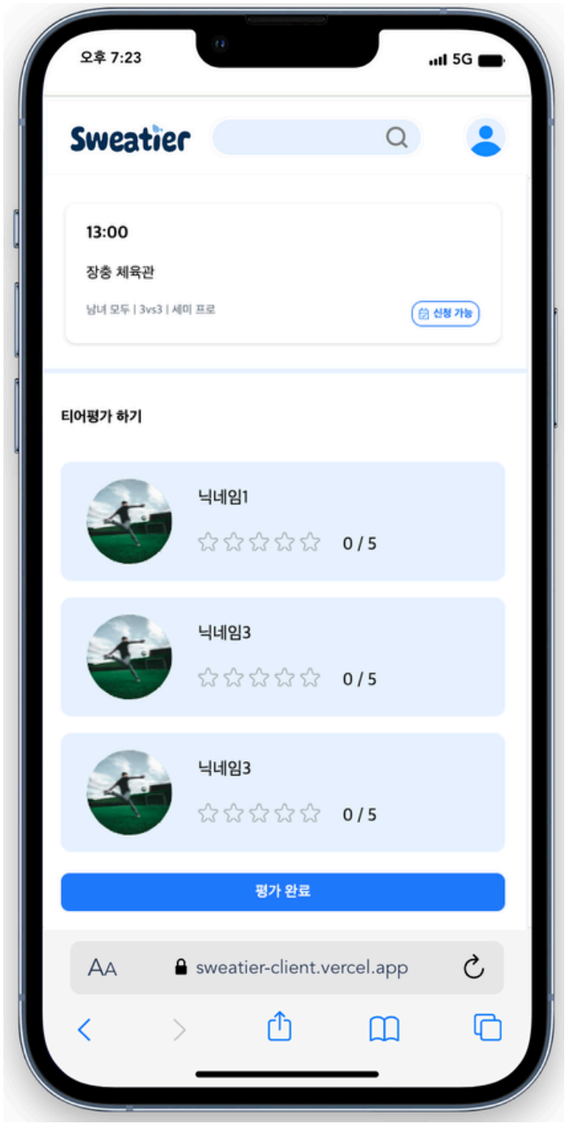
```
export function convertUTCDateToKST(date) {
 const utcDate = new Date(date);
 utcDate.setHours(utcDate.getHours() + 9);  // Add 9 hours to the UTC date to convert it to KST (Korea Standar
 return utcDate;
}

export function convertDateToUTC(date) {
 const localDate = new Date(date);
 localDate.setHours(localDate.getHours() - 9);  // Subtract 9 hours from the local date to convert it to UTC t
 return localDate;
}

export function adjustDates(obj, dateConversionFunction) {
 if (!obj || typeof obj !== 'object') return;  // Check if the input is not an object or is null, then exit th

 for (const key of Object.keys(obj)) {
   // If the property is a Date object, apply the provided date conversion function
   if (obj[key] instanceof Date) {
     obj[key] = dateConversionFunction(obj[key]);
   }
   // If the property is an object, recursively adjust its dates
   else if (typeof obj[key] === 'object') {
     adjustDates(obj[key], dateConversionFunction);
   }
 }
}
```

1

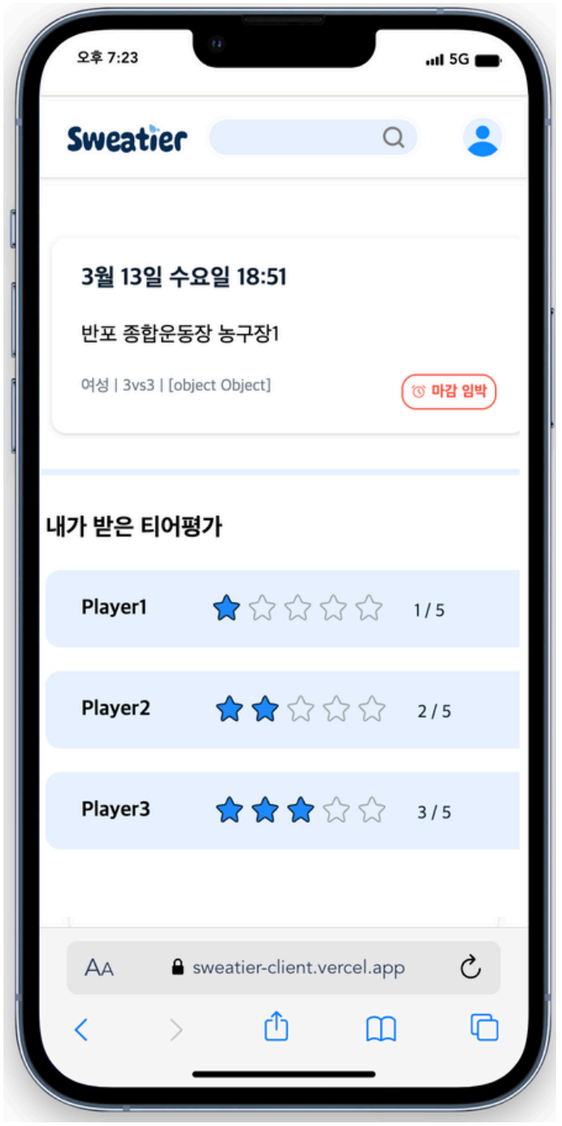
https://github.com/team-sweatier/sweatier-server/pull/56

https://github.com/prisma/prisma/issues/5051

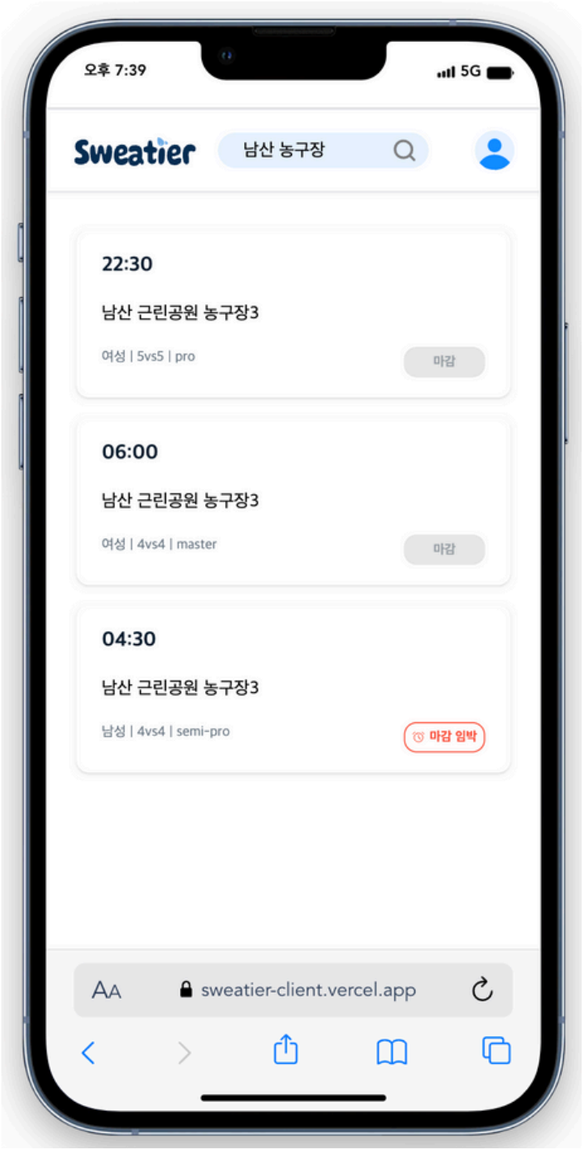
서비스 시연



평가 페이지



평가 조회 페이지



검색 결과 페이지

https://sweatier-client.vercel.app