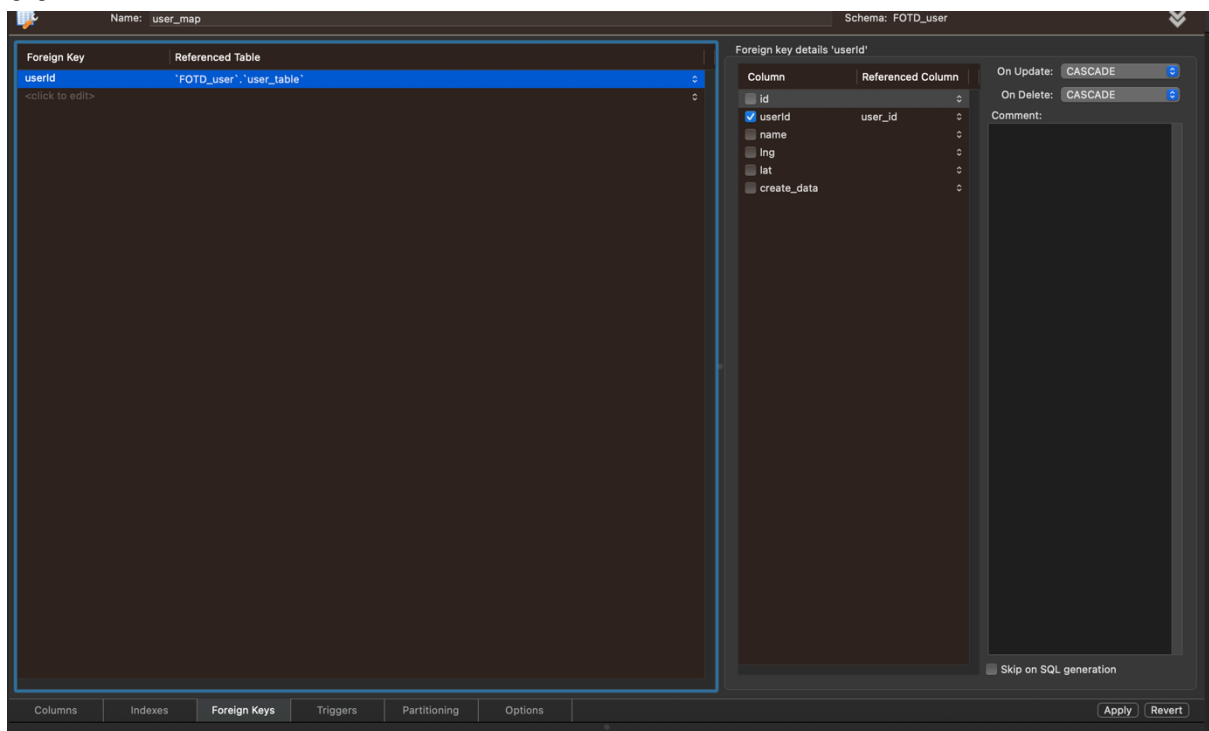


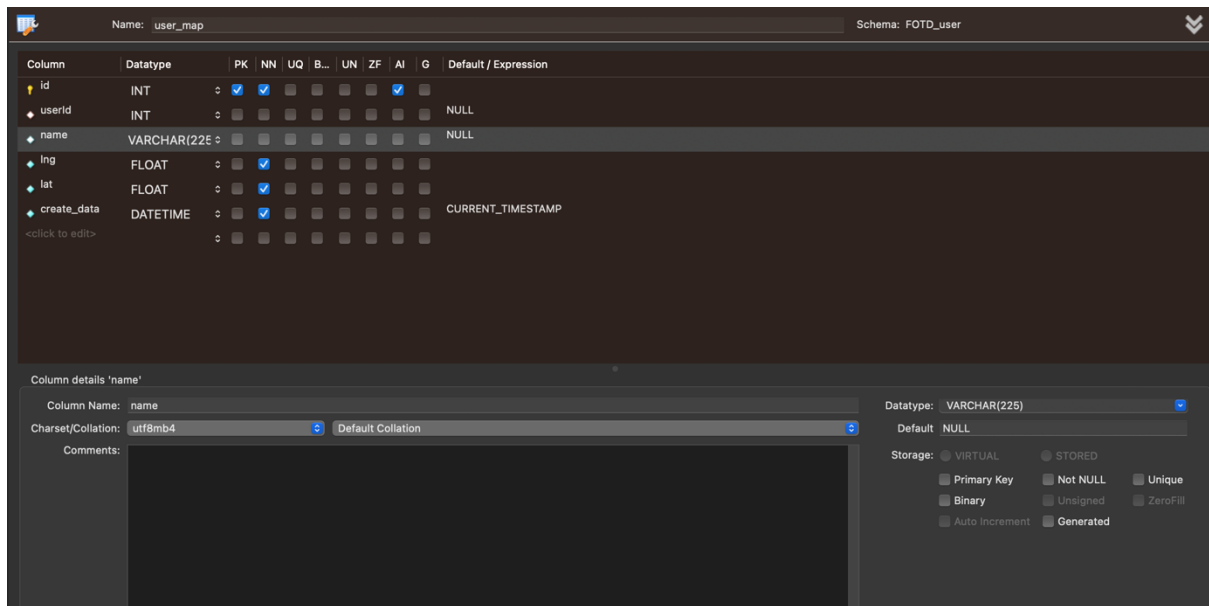
listSample

모듈 다운

```
npm install axios  
npm install body-parser  
npm install cors  
npm install detenv  
npm install express  
npm install mysql mysql2  
npm install sequelize  
npm install sequelize-cli
```

db





Charset/collation utf8mb4 utf8mb4_general-ci



``const userId = req.params.userId;``는 라우트의 URL 매개변수로부터 ``userId`` 값을 가져오는 것입니다. 이 값은 프론트엔드에서 요청을 보낼 때 해당 URL에 사용자의 식별 정보를 넣어서 보내는 것입니다.

``/api/user/:userId/``와 같이 라우트를 설정하면, ``:userId`` 부분에 실제 사용자의 아이디가 들어갑니다. 이를 받아오기 위해서 회원가입 백엔드에서는 사용자의 아이디를 생성하고 관리하는 로직이 필요합니다. 주로 사용자가 회원가입을 할 때, 사용자 아이디를 생성하고 데이터베이스에 저장하는 로직을 구현합니다.

일반적으로 회원가입 백엔드에서는 다음과 같은 작업을 수행합니다:

1. 사용자가 회원가입을 진행하면, 사용자 아이디를 생성하고 다른 필요한 정보(이메일, 비밀번호 등)와 함께 데이터베이스에 저장합니다.
2. 저장된 사용자 아이디는 후에 로그인이나 API 요청 시에 활용됩니다.
3. 사용자가 로그인할 때, 발급된 사용자 아이디를 클라이언트(프론트엔드)로 전달하여 해당 사용자에 대한 요청에 사용합니다.

따라서, 회원가입 백엔드에서 사용자 아이디 생성 및 저장 로직을 구현하고, 클라이언트에서는 해당 아이디를 사용하여 API 요청을 보낼 수 있도록 해야 합니다.



물론, 클라이언트에서 해당 아이디를 사용하여 API 요청을 보내려면 아래와 같은 단계를 따라야 합니다. 아래 예시는 React Native를 사용한 Expo 프로젝트를 기준으로 설명하겠습니다.

1. **회원가입 및 로그인:** 사용자가 회원가입 시 생성된 아이디와 비밀번호 등을 서버에 전송하여 계정을 생성하고 로그인합니다.
2. **API 요청 보내기:** 사용자가 특정 작업을 수행하기 위해 API 요청을 보낼 때, 요청에 사용자 아이디를 포함시켜 보내야 합니다.
3. **Axios 또는 fetch 사용:** API 요청을 보내기 위해 Axios나 fetch와 같은 라이브러리를 사용합니다. 이 예시에서는 Axios를 사용하겠습니다.
4. **요청 보내기:** 클라이언트에서 요청을 보낼 때, 해당 요청의 URL에 사용자 아이디를 포함하여 보내는 것이 중요합니다. 예를 들어, 사용자 아이디가 `123`인 경우:

javascript

Copy code

```
import axios from 'axios';

const userId = 123; // 사용자 아이디

// 요청 보내기
axios.get(`https://your-api-url/api/user/${userId}`)
  .then(response => {
    console.log(response.data);
    // 이곳에서 받아온 데이터를 활용
  })
  .catch(error => {
    console.error(error);
    // 에러 처리
  });
```

위 코드에서 `https://your-api-url/api/user/\${userId}` 부분에서 `\${userId}` 자리에 사용자 아이디가 들어갑니다. 서버에서는 이 사용자 아이디를 이용하여 해당 사용자의 데이터를 조회합니다.

Regen

프론트 지도 띄우는 건 mapView나 WebView 사용하면 되는 거 같아요