



코드 사용 매뉴얼

진행 상태

★ Now

버전

- Docker 가 설치되어 있어야 합니다.
- 개발 환경 : Android gradle 8.1, Flutter 3.27.1, openjdk version "17.0.13", Dart 3.6.0

Git Project Clone

1. 다음의 <https://github.com/zzerogu/cnumap> 링크에서 Git 클론을 통해 로컬에 프로젝트를 클론한다.
2. 본 프로젝트는 크게 backend, database, frontend 로 구성되며, [데이터베이스 도커 실행](#) → [백엔드 도커 실행](#) → [프론트엔드 실행](#) 의 순서로 초기 세팅을 진행해야한다.

Backend 매뉴얼

.env 파일 생성 (/cnumap 디렉터리 기준)

```
cd backend
cd rust_server
echo 'DATABASE_URL="postgres://postgres:postgres@host.docker.internal:5432/osm?sslmode=disable"' > .env
echo 'LOCAL_DATABASE_URL="postgres://postgres:postgres@localhost:5432/osm?sslmode=disable"' >> .env
```

```
echo 'DOCKER_DATABASE_URL="postgres://postgres:postgres@host.docker.internal:5432/osm?sslmode=disable"' >> .env

cd ..
echo 'POSTGRES_USER=postgres' > .env
echo 'POSTGRES_PASSWORD=postgres' >> .env
echo 'POSTGRES_DB=osm' >> .env
```

- `rust_server/.env` 파일 설명:
 - `cargo run` 실행 시: `DATABASE_URL` 에 `LOCAL_DATABASE_URL` 설정
 - `docker-compose up --build` 실행 시: `DATABASE_URL` 에 `DOCKER_DATABASE_URL` 설정

Migrations 파일 생성 (/cnumap 디렉터리 기준)

```
cd backend
cd rust_server
mkdir migrations
```

주의: `migrations` 폴더는 Docker 실행 시 필요합니다.

실행 순서 (Docker 버전)

실행 전 세팅:

- `rust_server/.env` 파일에서 `DATABASE_URL` 값을 `DOCKER_DATABASE_URL` 로 설정
- 터미널 2개 사용
- `/cnumap` 디렉터리로 이동

1. Docker 네트워크 생성 (처음 1회 실행)

```
docker network create navigation-network
```

2. 데이터베이스 컨테이너 빌드 및 실행 (터미널 1)

```
cd database
```

```
docker-compose up --build
```

3. 백엔드 서버 컨테이너 빌드 및 실행 (터미널 2)

```
cd backend  
docker-compose up --build
```

실행 순서 (로컬 버전)

실행 전 세팅:

- `rust_server/.env` 파일에서 `DATABASE_URL` 값을 `LOCAL_DATABASE_URL` 로 설정
- `/cnumap` 디렉터리로 이동
- 터미널 2개 사용
- `cargo` 가 설치되지 않은 경우 설치 필요

1. Docker 네트워크 생성 (처음 1회 실행)

```
docker network create navigation-network
```

2. 데이터베이스 컨테이너 빌드 및 실행 (터미널 1)

```
cd database  
docker-compose up --build
```

3. 백엔드 서버 로컬 실행 (터미널 2)

```
cd backend  
cd rust_server  
cargo run
```

▲ 주의: 로컬 버전에서는 맵 렌더링과 내비게이션 기능이 지원되지 않습니다.

서버 정보 및 Swagger 주소

- Swagger 주소: <http://localhost:8000/swagger-ui/#/>

서버 포트 번호:

- `rust_server`: 8000
- `tegola_server`: 8080
- `valhalla_server`: 8002

⚠️ 주의사항:

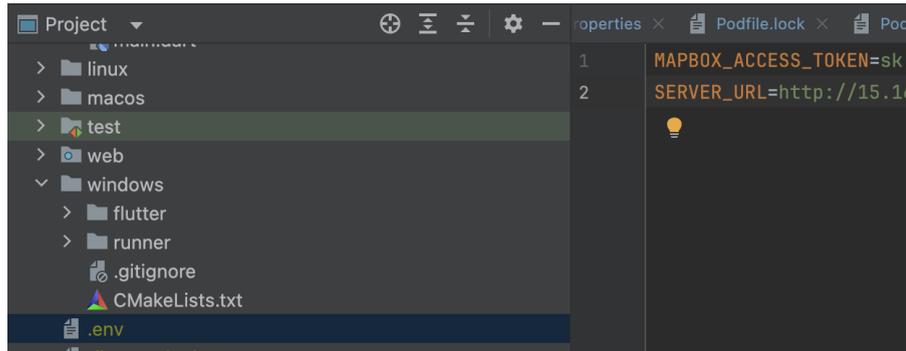
- Docker 첫 빌드는 30분 이상 소요될 수 있습니다.
- 로컬 실행 시 맵 관련 API는 지원되지 않습니다.

Frontend 매뉴얼

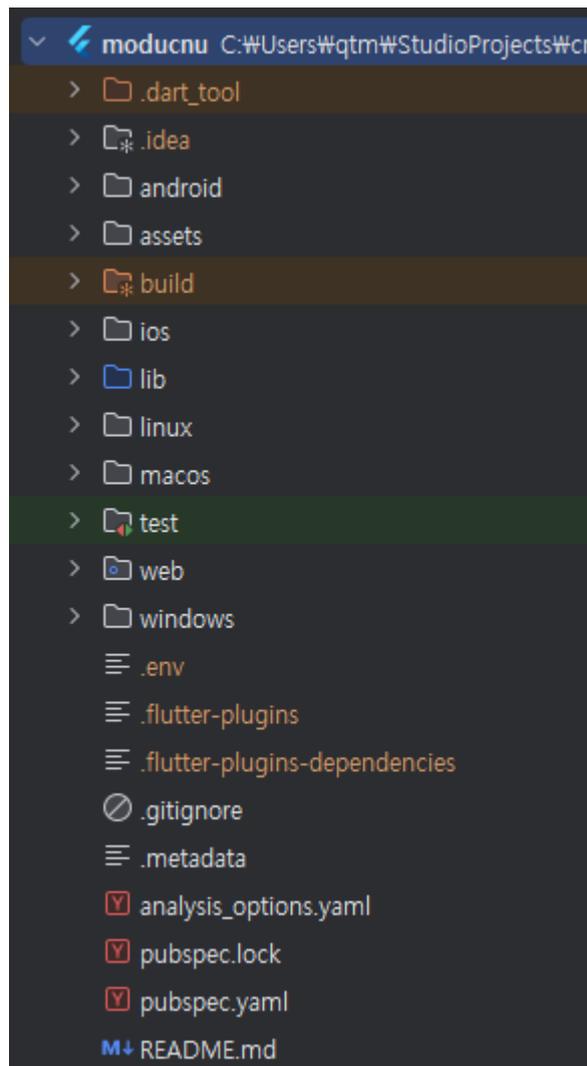
⚠️ 본 프로젝트는 Android Studio(Iguana) (혹은 Visual Studio) , Flutter 3.27.1 (+dart sdk), Android gradle(8.1, open jdk 17) 가 설치되어있어야합니다. 충돌이 난다면, 버전 확인 부탁드립니다.

1. MapBox API 키를 발급받는다.
 - a. MapBox API Key 발급) <https://docs.mapbox.com/help/getting-started/access-tokens/>
2. 프론트엔드 프로젝트 루트 경로에 다음처럼 `.env` 파일을 생성하여, 그 내부에 API키와 주소를 입력한다 (아래 코드 참고)

```
MAPBOX_ACCESS_TOKEN={MAPBOX API 키}
SERVER_URL=http://localhost:8000/
```



3. 안드로이드 스튜디오 혹은 Visual Studio 를 열고, 로컬에 클론한 파일 경로 중 `cnumap\frontend\moducnu` 의 경로로 프로젝트를 연다



4. 혹시 dart sdk 가 등록되어있지 않다면 `Edit configuration` 에서 dart sdk 를 등록한다.
5. 그 후 Emulator 을 통해 디바이스를 선택(iOS, Android 가능, 크롬 불가능)하고, `main.dart` 우측의 `run` 버튼을 통해 실행한다.

a. 에뮬레이터가 없다면, 에뮬레이터 다운이 사전에 필요하다) <https://ship-jh.tistory.com/8>

6. 기기가 실행되면, 테스트가 가능하다.