



**Save  
Endangered  
Animal**

## **Save Endangered Animals(SEA)**

SSAFY 7기 서울 5반 6팀

(2022.08.22 ~ 2022.10.07)

### **포팅 매뉴얼**

팀장 : 김현영

팀원 : 강경은, 김경환, 성성민, 이종인

## < 목차 >

1. 기술 스택 -----	1
2. 빌드 주요 내용 -----	2
3. 배포 시 특이사항 -----	3
4. DB 연결 -----	3
5. 외부 서비스 -----	5

# 1. 기술 스택

## 가) 개발 환경

(ㄱ) OS

Local : Windows 10

(ㄴ) IDE

IntelliJ IDE 2022.1.3

Visual Studio Code 1.70.0

(ㄷ) UI / UX : Figma

(ㄹ) DataBase

MySQL workbench 8.0.20 (Windows 10)

(ㅁ) Docker : 20.10.17

## 나) 상세 스택

(ㄱ) Backend

JAVA

- openjdk 11
- Gradle 6.8
- Spring Boot 2.4.5

Python

- 3.10

(ㄴ) FrontEnd

- HTML5, CSS3, JavaScript(ES6)
- React 18.2.0,
- Node.js 16.17.0, npm 8.15.0
- Web3 1.6.1

(ㄷ) Smart Contract

- Solidity 0.8.4^, Remix 0.26.3

다) 이슈 관리 : Jira

라) 형상 관리 : Gitlab

마) 커뮤니케이션 : Notion, MatterMost

## 2. 빌드 주요 내용

가) Clone

```
git clone https://lab.ssafy.com/s07-blockchain-nft-sub2/S07P22A506.git
```

나) Frontend 빌드

[S07P22A506/frontend] 폴더로 들어갑니다.

```
cd S07P22A506/frontend
```

package-lock.json에 정의된 모든 dependencies에 대하여 install을 합니다.

```
npm install
```

install이 완료되었다면 빌드를 진행합니다.

```
npm run build or npm start
```

build를 하게 되면 frontend/build 폴더에 프로젝트가 빌드 됩니다. nginx에서 해당 폴더의 index.html을 띄워주도록 설정해 두었습니다.

nginx 설치

```
sudo apt install nginx
```

nginx.conf 설정

```
server {
    listen      80;
    listen  [::]:80;

    server_name _;

    location / {
        return 301 https://$host$request_uri;
    }
}
server {
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;
    root   {frontend 빌드 위치}
    index  index.html index.htm;
    location / {
        try_files $uri $uri/ /index.html =404;
    }
}
```

```
server_name _;  
    ssl_certificate {fullchain.pem 파일 위치};  
    ssl_certificate_key {privkey.pem 파일 위치};  
}
```

#### 다) Backend 빌드

[S07P22A506/backend] 폴더로 들어갑니다

```
cd S07P12A404/backend
```

Build를 하고 jar파일을 실행합니다.

```
./gradlew build  
  
cd build/lib/  
  
java -jar {jar 파일 이름}
```

#### 라) Replicate 빌드

[S07P22A506/replicate] 폴더로 들어갑니다.

```
cd S07P12A404/replicate
```

Python이 설치되어 있지 않다면 설치하고

설치되어 있다면 실행시킵니다.

```
Python app /main.py
```

### 3. 배포 시 특이사항

현재 AWS 서버와 Jenkins, Docker를 사용하여 자동배포를 하고 있기때문에 각 폴더 내에 Dockerfile이 존재합니다. Docker를 사용할 경우 도커 파일이 있는 경로로 이동한 후에 아래 명령어를 입력하면됩니다.

```
docker build
```

도커 이미지를 이미지 파일로 만드는 명령어

```
docker build -t { 이미지 이름 } .  
docker save { 이미지 이름 } -o { 파일 이름 }.tar
```

이미지 파일을 이미지로 만드는 명령어

```
docker load -i { 파일 이름 }.tar
```

도커 컨테이너 실행

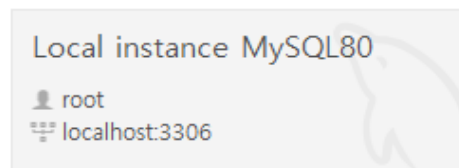
```
docker run {이미지 이름}
```

## 4. DB 연결

MySQL server는 EC2 내에서 동작 중인 상태입니다. DB를 Local PC에서도 관리하기 위해서는 아래의 방법을 따라야합니다.

1) Local PC MySQL workbench에서 new Connection

### MySQL Connections + ⓘ



2) 원하는 이름으로 Connection Name를 지정하고 username에 'a506', store in vault에 MySQL server에서 설정한 a506의 비밀번호 '비번!!!'를 적습니다.

3) Test Connection을 눌러 Success가 뜬다면 OK버튼을 누릅니다.

## 5. 외부 서비스

가) Stable Diffusion API : 인공지능 이미지 생성

나) Metamask : Ropsten 등 블록체인과 상호작용하는 Provider 제공 및 본 서비스를 이용하는 데 사용되는 Ethereum 지갑 관리 기능 제공