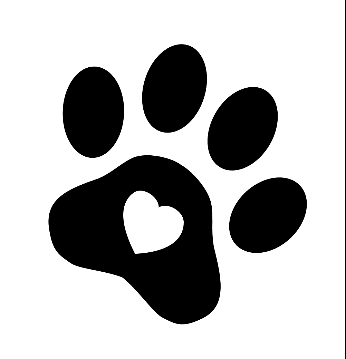
|  |
| --- |
|  |



|  |
| --- |
| **반가워** |
| **포팅 메뉴얼** |
|  |
|  |
|  |
|  |
|  |



목차

[I. 개요 2](#_Toc96072459)

[1. 프로젝트 개요 2](#_Toc96072460)

[2. 프로젝트 사용 도구 2](#_Toc96072461)

[3. 개발환경 2](#_Toc96072462)

[II. 빌드 3](#_Toc96072465)

1. 환경 변수 3

[2. 빌드 설정 6](#_Toc96072466)

[3. 배포 6](#_Toc96072467)

개요

## 프로젝트 개요

기존의 입양 서비스들은 사진으로 희망하는 입양 동물을 확인해야 합니다.

때문에 정보가 제한되어 어떤 성격인지, 다른 모습은 어떠한지 확인이 불가합니다.

“반가워”를 통해 입양 희망자는 유기 동물을 영상으로 확인할 수 있습니다.

또한, 직접 방문하기 이전에 먼저 다양한 정보를 확인할 수 있고, 여러 동물에 대한 확인이 가능합니다.

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost, Discord

디자인 : Figma

UCC : Movavi video editor

CI/CD : Jenkins

## 개발환경

VS Code : 1.75.1,

STS : 3.9.14.RELEASE

IntelliJ : 17.0.5+1-b653.25 amd64

JDK : openjdk 11.0.15(LTS)

Node.js : 16.15.0

SERVER : AWS EC2 Ubuntu 20.04 LTS

DB : mysql 8.0.30 for windows on x86\_64

# 빌드

## 1. 환경 변수

logging:  
 level:  
 root: info  
 org:  
 hibernate:  
 SQL: debug  
*# type: TRACE* com:  
 ssafy:  
 backend:  
 domain:  
 member:  
 controller: info  
  
 file:  
 name: ./demo.log  
  
server:  
 ssl:  
 key-store: classpath:ssl/keystore.p12  
 key-store-type: PKCS12  
 key-store-password: ssafy  
  
OPENVIDU\_URL: https://i8b209.p.ssafy.io:8445  
OPENVIDU\_SECRET: ssafy  
  
  
spring:  
  
 mvc:  
 pathmatch:  
 matching-strategy: *ant\_path\_matcher* cache:  
 type: *redis* redis:  
 host: i8b209.p.ssafy.io  
 port: 6379  
  
 devtools:  
 livereload:  
 enabled: true  
  
 jpa:  
 hibernate:  
 ddl-auto: create  
 properties:  
 hibernate:  
 show-sql: true  
 format\_sql: true  
 generate-ddl: true  
 database: *mysql* database-platform: org.hibernate.dialect.MySQL8Dialect  
 defer-datasource-initialization: true  
  
 datasource:  
 driver-class-name: com.mysql.cj.jdbc.Driver  
 url: jdbc:mysql://i8b209.p.ssafy.io:3306/hello\_db?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTimeBehavior=convertToNull&rewriteBatchedStatements=true&useSSL=false  
 username: ubuntu  
 password: ssafy  
  
 security:  
 oauth2:  
 client:  
 registration:  
 google:  
 client-id: 524697668949-st4sbncc7brsgpfb2kaqc604cvr0hfrv.apps.googleusercontent.com  
 client-secret: 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  
 scope: profile, email  
 naver:  
 client-id: XVUGCiFeoQqa6Qc9Q24L  
 client-secret: cd9Zbm09h9  
 scope: name, email  
 client-name: Naver  
 authorization-grant-type: authorization\_code  
 redirect-uri: http://localhost:9999/oauth/naver  
 provider:  
 naver:  
 authorization-uri: https://nid.naver.com/oauth2.0/authorize  
 token-uri: https://nid.naver.com/oauth2.0/token  
 user-info-uri: https://openapi.naver.com/v1/nid/me  
 user-name-attribute: response  
  
 mail:  
 host: smtp.gmail.com  
 port: 587  
 username: test@gmail.com  
 password: test  
 properties:  
 mail:  
 smtp:  
 auth: true  
 starttls:  
 enable: true  
   
 servlet:  
 multipart:  
 enabled: true  
 location: /temp  
 file-size-threshold: 1MB  
 max-file-size: 40MB  
 max-request-size: 20MB  
   
file:  
 upload:  
 basicPath: /upload\_test/images  
 userPath: user  
 animalPath: animal  
  
jwt:  
 header: Authorization  
 secret: d1659096d8c5bd34c2e96fdb235c123afccf23450af0870d95813d820c0dd44661507fe41d5a12c5b796299d260f314d9a5bdca31050fc00a2a4b44c9d31f93c  
 validity-second:  
*# # test1  
# access: 120  
# refresh: 240  
# # test2 - no used refresh* access: 3600  
 refresh: 3600

## 2. 빌드 설정

1) Front

npm i

npm start

2) Back-spring

Gradle 실행

Bootjar 실행

## 3. 배포

Nginx 설정

server {

listen 443 ssl;

server\_name i8b209.p.ssafy.io;

ssl\_certificate /etc/letsencrypt/live/i8b209.p.ssafy.io/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/i8b209.p.ssafy.io/privkey.pem;

location / {

proxy\_pass http://i8b209.p.ssafy.io:3000;

}

location /api {

proxy\_pass http://i8b209.p.ssafy.io:9999/api;

}

}

server {

if ($host = i8b209.p.ssafy.io) {

return 301 https://$host$request\_uri;

}

listen 80;

server\_name i8b209.p.ssafy.io;

return 404;

}

이후 sudo service nginx start