|  |
| --- |
|  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |



목차

[I. 개요 2](#_Toc127430178)

[1. 프로젝트 개요 2](#_Toc127430179)

[2. 프로젝트 사용 도구 2](#_Toc127430180)

[3. 개발환경 2](#_Toc127430181)

[4. 외부 서비스 2](#_Toc127430182)

[5. Gitgnore 처리한 핵심 키들 3](#_Toc127430183)

[II. 빌드 3](#_Toc127430184)

[1. 환경변수 형태 3](#_Toc127430185)

[2. 빌드하기 4](#_Toc127430186)

[3. 배포하기 5](#_Toc127430187)

[4. 서비스 이용 방법 6](#_Toc127430188)

[# OAuth 6](#_Toc127430189)

[Google Cloud Platform 6](#_Toc127430190)

[Naver Developers 7](#_Toc127430191)

[Kakao Developers 7](#_Toc127430192)

[# Map 7](#_Toc127430193)

[Kakao Developers 7](#_Toc127430194)

# 개요

## 프로젝트 개요

하나 뿐인 내 강아지에게 동네 친구를 만들어주세요!

반려동물 1500만 시대, 주변의 강아지 친구들도 늘어나고 있습니다. 강아지들에게 필수인 산책 다들 어떻게 하고 계신가요?

우리 TOGETDOG에서는 내 강아지에게 맞춤인 산책 친구들을 찾아볼 수 있습니다. 몸무게, 중성화 여부 등 안전을 고려한 추천을 사용해보세요. 위치 정보를 기반으로 제공되는 산책 친구 추천을 통해 지금 내 소중한 강아지에게 친구를 만들어주세요.

강아지와의 소중한 순간을 남길 수 있는 SNS 기능도 제공하고 있습니다!

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Slack, Mattermost

디자인 : Figma

UCC : DaVinci Resolve

CI/CD : Jenkins 🡪 Docker

## 개발환경

STS : 3.9.14 release

JDK: zulu-8

Gradle: 7.6

React: 17

Node.js : 18.12.1

SERVER : AWS EC2 Ubuntu 20.04.5 LTS

DB : MySQL 8.0.31

## 외부 서비스

Google Login API

Kakao Login API

Naver Login API

Google Gmail STMP

Kakao Map API

## Gitgnore 처리한 핵심 키들

React : .env

(classPath에 위치)

Spring : application.properties

(\src\main\resources, 또는 classPath에 위치)

# 빌드

## 환경변수 형태

* **application.properties**

# MySQL DB(배포)

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://43.201.82.20:3333/togetdog\_?useSSL=true&useUnicode=true&serverTimezone=Asia/Seoul&allowPublicKeyRetrieval=true

spring.datasource.username=ppopy

spring.datasource.password=rhdxhda807

# Security OAuth

# Kakao

spring.security.oauth2.client.registration.kakao.client-id= “카카오 클라이언트 키”

spring.security.oauth2.client.registration.kakao.client-secret= “카카오 시크릿 키”

spring.security.oauth2.client.registration.kakao.client-name=Kakao

spring.security.oauth2.client.registration.kakao.client-authentication-method=POST

spring.security.oauth2.client.registration.kakao.redirect-uri=https://togetdog.site/oauth2/code/{registrationId}

spring.security.oauth2.client.registration.kakao.authorization-grant-type=authorization\_code

spring.security.oauth2.client.registration.kakao.scope=profile\_nickname,account\_email

# Naver

spring.security.oauth2.client.registration.naver.client-id= “네이버 클라이언트 키”

spring.security.oauth2.client.registration.naver.client-secret=“네이버 시크릿 키”

spring.security.oauth2.client.registration.naver.client-name=Naver

spring.security.oauth2.client.registration.naver.redirect-uri=https://togetdog.site/oauth2/code/{registrationId}

spring.security.oauth2.client.registration.naver.authorization-grant-type=authorization\_code

spring.security.oauth2.client.registration.naver.scope=email,nickname

# Google

spring.security.oauth2.client.registration.google.client-id=“구글 클라이언트 키”

spring.security.oauth2.client.registration.google.client-secret= “구글 시크릿 키”

spring.security.oauth2.client.registration.google.redirect-uri=https://togetdog.site/oauth2/code/{registrationId}

spring.security.oauth2.client.registration.google.scope=email,profile

# JWT 시크릿 키

jwt.secret=

* **.env**

**# KAKO MAP API KEY**

REACT\_APP\_KAKAO\_MAP\_API\_KEY =

## 빌드하기

1) Front-react

cd togetdog\_client

chmod 755 bash.sh

./bash.sh

2) Back-spring

cd togetdog

chmod 755 gradlew

./gradlew build

chmod 755 bash

./bash.sh

## 배포하기

Nginx 설정

# 443 포트로 접근시 ssl을 적용한 뒤 3000포트로 요청을 전달해주도록 하는 설정.

server {

server\_name togetdog.site www.togetdog.site;

location / {

proxy\_pass http://127.0.0.1:3000;

}

location /api {

proxy\_pass http://127.0.0.1:8080;

}

location /ws/chat {

proxy\_pass http://127.0.0.1:8080;

}

location /oauth2 {

proxy\_pass http://127.0.0.1:8080;

}

location /image {

root /home/ubuntu;

}

listen 443 ssl;

listen [::]:443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/togetdog.site/fullchain.pem; # managed by Cert>

ssl\_certificate\_key /etc/letsencrypt/live/togetdog.site/privkey.pem; # managed by Ce>

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

# 80 포트로 접근시 443 포트로 리다이렉트 시켜주는 설정

server {

if ($host = www.togetdog.site) {

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

} # managed by Certbot

if ($host = togetdog.site) {

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

} # managed by Certbot

if ($host = i8a807.p.ssafy.io) {

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

} # managed by Certbot

listen 80;

listen [::]:80;

server\_name www.togetdog.site togetdog.site i8a807.p.ssafy.io;

location / {

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

}

return 404; # managed by Certbot

}

이후 sudo systemctl start nginx

## 서비스 이용 방법

### # OAuth

### Google Cloud Platform

1. 서비스 전용 google 계정을 만들어 사용했습니다. 해당 계정으로 구글 클라우드 플랫폼으로 접속합니다.
2. TOGETDOG 서비스를 선택하고 API 및 서비스 탭으로 이동합니다.
3. OAuth 동의 화면에서 앱이 만들어져있다면 수정을 눌러 값을 입력하고 없다면 생성합니다.
4. 사용자 인증 정보에서 OAuth2.0 클라이언트 ID를 확인하고 이름을 눌러 서비스 URL과 리다이렉션 URL을 확인 및 입력합니다.

### Naver Developers

1. 네이버 개발자 센터로 이동합니다.
2. 내 어플리케이션에서 애플리케이션 등록을 선택하고 사용 API에서 네이버 로그인을 선택합니다.
3. 애플리케이션이 등록되었다면 내 애플리케이션 탭으로 이동하여 클라이언트 ID를 확인합니다. 해당 ID를 클릭하면 API의 상세 설정으로 이동합니다.
4. 이동되었다면 클라이언트 SECRET을 확인하고 저장합니다
5. API 설정탭에서 애플리케이션의 이름, 사용 API의 범위, 서비스 URL과 리다이렉션 URL, 로고 등을 입력합니다.
6. 네이버 로그인은 검수 전 미리 등록한 사용자만 가능하므로, 검수 전이라면 테스트 계정을 멤버 등록시킵니다.

### Kakao Developers

1. 카카오 개발자 센터로 이동합니다.
2. 내 에플리케이션 탭을 누르고 애플리케이션이 없다면 추가하기를 선택하여 서비스에 사용할 app을 추가해줍니다.
3. 만들어져있거나 만들었다면 해당 app을 선택하여 앱설정탭으로 이동할 수 있습니다. 앱 설정탭으로 이동했다면 앱키를 확인하여 사용합니다.
4. 카카오 로그인 탭의 동의항목에서 서비스에 필요한 동의항목을 선택합니다.
5. app설정에서 리다이렉션 URL을 입력합니다.

### # Map

### Kakao Developers

1. 카카오 개발자 센터로 이동합니다.
2. 내 에플리케이션 탭을 누르고 애플리케이션이 없다면 추가하기를 선택하여 서비스에 사용할 app을 추가해줍니다.
3. 만들어져있거나 만들었다면 해당 app을 선택하여 앱설정탭으로 이동할 수 있습니다. 앱 설정탭으로 이동했다면 앱키를 확인하여 사용합니다.
4. 카카오 로그인 탭의 동의항목에서 서비스에 필요한 동의항목을 선택합니다.
5. app설정에서 리다이렉션 URL을 입력합니다.