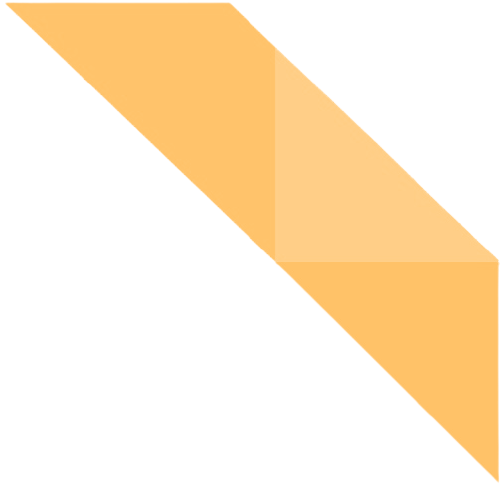
|  |
| --- |
|  |



|  |
| --- |
| **YourJS** |
| **포팅 매뉴얼** |
|  |
|  |
|  |
|  |
|  |



목차

[**I. 개요 2**](#_1fob9te)

[**1. 프로젝트 개요 2**](#_3znysh7)

[**2. 프로젝트 사용 도구 2**](#_2et92p0)

[**3. 개발환경 2**](#_tyjcwt)

[**4. 외부 서비스 2**](#_3dy6vkm)

[**5. Gitgnore 처리한 핵심 키들 3**](#_1t3h5sf)

**[II. 빌드 3](#_4d34og8)**

[**1. 환경변수 형태 3**](#_2s8eyo1)

[**2. 빌드하기**](#_17dp8vu) 4

[**3. 배포하기**](#_3rdcrjn)5

# 개요

## 프로젝트 개요

구직 활동을 하면서 구직자들은 일정 관리, 자기소개서 관리 등 다양한 어려움을 겪습니다.

YourJS는 이러한 어려움을 겪는 IT 개발자분들을 위해 편리한 일정 템플릿을 제공하고 태그 기반 자기소개서 항목 검색을 통해 동일한 자기소개서를 빠르게 찾을 수 있도록 돕습니다. 그리고 평소 까먹기 쉬웠던 전공 이수학점, 자격증 번호 등 세세한 정보까지 관리할 수 있는 서비스를 제공합니다.

Your Job finding Secretary. YourJS

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

Open API 관리 : Swagger

커뮤니케이션 : Notion, Discord

디자인 : Figma

UCC : 모바비

CI/CD : Jenkins, Kubernetes, Docker Hub

## 개발환경

Text Editor: Visual Studio Code IDE

Backend: Spring Boot

Frontend: React

Server : AWS EC2 Ubuntu 20.04 LTS

DB : MySQL, Mongo DB

## 외부 서비스

카카오 로그인 api : back/yourjs/src/main/resources/properties/env.properties (카카오 로그인시 사용)

## Gitignore 처리한 핵심 키들

Frontend : .env (front/yourjs/.env)

Backend : env.properties (back/yourjs/src/main/resources/properties/env.properties)

# 빌드

## 환경변수 형태

.env

REACT\_APP\_KAKAO\_API\_KEY={Kakao API Key}

REACT\_APP\_KAKAO\_REDIRECT\_URI=https://{도메인}/login/kakao

env.properties

# db

properties.file.url=jdbc:mysql://{MySQL주소}/yourjs?characterEncoding=UTF-8&serverTimezone=Asia/Seoul

properties.file.username={MySQL계정이름}

properties.file.password={MySQL계정비밀번호}

# jwt

jwt.header=Authorization

properties.file.jwtSecret={JWT Secret Key}

properties.file.jwtRefreshSecret={JWT Refresh Secret Key}

properties.file.jwtExpiration= 600

properties.file.jwtRefreshExpiration = 1209600

# mongoDB

env.mongoDB.username={Mongo Username}

env.mongoDB.password={Mongo PW}

env.mongoDB.cluster={Mongo Cluster}

# aws s3

env.s3.accessKey={S3 Access Key}

env.s3.secretKey={S3 Secret Key}

env.s3.bucketName={S3 버킷명 적는곳 버킷 yourjs-bucket 로 생성해야함}

env.s3.region={S3 지역}

# kakao login

kakao.client\_id={Kakao Client ID}

kakao.client\_secret={Kakao Client Secret}

## 빌드하기

0) 공통

1. AWS Ubuntu 환경 구성
2. 도커 설치
3. 쿠버네티스 설치(kubeadm)
4. 쿠버네티스에 ingress-nginx 설치
5. letsencrypt로 도메인 SSL 인증
6. git clone
7. Project Root 디렉토리로 이동

1) Front

1. front/yourjs/ 위치에 .env 작성(환경변수 형태 참고)
2. front/yourjs/src/common/apis.js 2 줄

export const SERVER\_IP = '[https://{도메인}/api](about:blank)'; 으로 수정

1. Project Root 디렉토리로 이동
2. docker build -t {Docker Hub Repository 이름}yourjs-front ./front/yourjs
3. docker login -u {Docker Hub 계정명} -p {Docker Hub 비밀번호}
4. docker push {Docker Hub Repository 이름}:yourjs-front

2) Back

1. back/yourjs/src/main/resources/properties 위치에 env.properties 작성(환경변수 형태 참고)
2. back/yourjs/src/main/java/com/project/yourjs/api/service/UserService.java 247 줄 sb.append("&redirect\_uri=[https://{도메인}/login/kakao](about:blank)"); 으로 수정
3. Project Root 디렉토리로 이동
4. docker build -t {Docker Hub Repository 이름}yourjs-back ./back/yourjs
5. docker login -u {Docker Hub 계정명} -p {Docker Hub 비밀번호}
6. docker push {Docker Hub Repository 이름}:yourjs-back

## 배포하기

0) 공통

Project Root 디렉토리로 이동

1) Front

1. kubectl create ns ns-yourjs

2. kubectl create secret docker-registry my-secret \

--docker-server=https://index.docker.io/v1/ \

--docker-username={Docker Hub 계정명} \

--docker-password={Docker Hub 비밀번호} \

--docker-email={Docker Hub 이메일} \

-n ns-yourjs

3. kubectl create secret tls yourjstls \

--key={SSL인증 키 디렉토리}/privkey.pem \

--cert={SSL인증 키 디렉토리}/fullchain.pem \

-n ns-yourjs

4. front/yourjs/k8s/k8s/yourjs-front-deployment.yaml 수정

…

replicas: 4

strategy:

rollingUpdate:

maxSurge: 2

maxUnavailable: 1

type: RollingUpdate

… 위 부분 삭제

…

spec:

containers:

* name : yourjs-front

image: {Docker Hub Repository 이름}:yourjs-front

… 수정

5. kubectl apply -f ./front/yourjs/k8s/yourjs-front-deployment.yaml -n ns-yourjs

6. kubectl apply -f ./front/yourjs/k8s/yourjs-front-service.yaml -n ns-yourjs

7. kubectl apply -f ./front/yourjs/k8s/yourjs-front-ingress.yaml

2) Back

1. kubectl create ns ns-yourjs (front 배포과정에서 ns 생성했으면 할 필요 X)

2. kubectl create secret docker-registry my-secret \

--docker-server=https://index.docker.io/v1/ \

--docker-username={Docker Hub 계정명} \

--docker-password={Docker Hub 비밀번호} \

--docker-email={Docker Hub 이메일} \

-n ns-yourjs

3. kubectl create secret tls yourjstls \

--key={SSL인증 키 디렉토리}/privkey.pem \

--cert={SSL인증 키 디렉토리}/fullchain.pem \

-n ns-yourjs

4. front/yourjs/k8s/k8s/yourjs-back-deployment.yaml 수정

…

replicas: 4

strategy:

rollingUpdate:

maxSurge: 2

maxUnavailable: 1

type: RollingUpdate

… 위 부분 삭제

…

spec:

containers:

* name : yourjs-back

image: {Docker Hub Repository 이름}:yourjs-back

… 수정

5. kubectl apply -f ./back/yourjs/k8s/yourjs-back-deployment.yaml -n ns-yourjs

6. kubectl apply -f ./back/yourjs/k8s/yourjs-back-service.yaml -n ns-yourjs

7. kubectl apply -f ./back/yourjs/k8s/yourjs-back-ingress.yaml