

서울 6반 2조

포팅 매뉴얼

ZIP-PLZ

빌드 및 배포 문서

JVM Version

openjdk version "21.0.4"
OpenJDK Runtime Environment (build 21.0.4+7-
Ubuntu-1ubuntu220.04)
OpenJDK 64-Bit Server VM (build 21.0.4+7-
Ubuntu-1ubuntu220.04, mixed mode, sharing)

Tomcat

Spring boot 내장 tomcat 사용
server.port=5000

Intellij

Runtime version: 17.0.11+1-b1207.24
amd64
VM: OpenJDK 64-Bit Server VM by
JetBrains s.r.o.

빌드 및 배포 문서

Auto Build

Jenkins /2.452.3
Docker version 27.0.3

환경변수

특이사항

- master branch 기준 시간당 1회 자동 빌드 진행
- 젠킨스 자동 빌드 후
 - /pythonProject/ai_server/app.py 실행
 - /var/lib/jenkins/workspace/zipplz-be-build/ZIPPLZ_BE/build/libs/ZIPPLZ_BE-0.0.1-SNAPSHOT.jar 실행

빌드 및 배포 문서

Web server

NginX / 1.18.0

/etc/nginx/nginx.conf

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;
```

```
events {
    worker_connections 768;
    # multi_accept on;
}
```

```
http {
    map $http_origin $allowed_origin {
        default "";
        "http://localhost:8080" $http_origin;
        "https://zipplz.site" $http_origin;
    }
```

```
# 타임아웃 설정
proxy_connect_timeout 6000s;
proxy_send_timeout 6000s;
proxy_read_timeout 6000s;
fastcgi_read_timeout 6000s;
```

```
sendfile on;
tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 65;
types_hash_max_size 2048;
```

```
include /etc/nginx/mime.types;
default_type application/octet-stream;
```

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2
TLSv1.3; # Dropping SSLv3, ref:
POODLE
ssl_prefer_server_ciphers on;
```

```
access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;
gzip on;
```

```
include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}
```

/etc/nginx/sites-available/default

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

    root /usr/share/nginx/html;
    index index.html index.htm index.nginx-debian.html;

    server_name example.com;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }

    location /api {
        proxy_pass http://localhost:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        add_header Cache-Control "no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0";
        add_header Pragma "no-cache";
        add_header Expires "0";
        proxy_http_version 1.1;
        proxy_pass http://$host;
        proxy_set_header Connection "keep-alive";

        # GET/POST/HEAD 요청은 브라우저에서 요청 처리
        # GET/POST/HEAD 요청은 "GET"으로 처리
        add_header "Access-Control-Allow-Origin" "http://zipplz.site";
        add_header "Access-Control-Allow-Methods" "GET, POST, OPTIONS, PUT, DELETE";
        add_header "Access-Control-Allow-Headers" "Origin, Authorization, Accept, Content-Type, X-Requested-With";
        add_header "Access-Control-Max-Age" 1209600;
        add_header "Content-Type" "text/plain; charset=UTF-8";
        add_header "Content-Length" 0;
        return 200;
    }
}

server {
    listen 303 http://zipplz.site;

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/zipplz.site/ssl/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/zipplz.site/ssl/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    }

    server {
        listen listen([::]) default_server;
        listen [::] default_server;

        root /usr/share/nginx/html;
        index index.html index.htm index.nginx-debian.html;

        server_name example.com; # managed by Certbot

        location / {
            proxy_pass http://localhost:8080;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;

            add_header Cache-Control "no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0";
            add_header Pragma "no-cache";
            add_header Expires "0";
            proxy_http_version 1.1;
            proxy_pass http://$host;
            proxy_set_header Connection "keep-alive";

            # GET/POST/HEAD 요청은 브라우저에서 요청 처리
            # GET/POST/HEAD 요청은 "GET"으로 처리
            add_header "Access-Control-Allow-Origin" "http://zipplz.site";
            add_header "Access-Control-Allow-Methods" "GET, POST, OPTIONS, PUT, DELETE";
            add_header "Access-Control-Allow-Headers" "Origin, Authorization, Accept, Content-Type, X-Requested-With";
            add_header "Access-Control-Max-Age" 1209600;
            add_header "Content-Type" "text/plain; charset=UTF-8";
            add_header "Content-Length" 0;
            return 200;
        }
    }

    location /api {
        proxy_pass http://localhost:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        add_header Cache-Control "no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0";
        add_header Pragma "no-cache";
        add_header Expires "0";
        proxy_http_version 1.1;
        proxy_pass http://$host;
        proxy_set_header Connection "keep-alive";

        # GET/POST/HEAD 요청은 브라우저에서 요청 처리
        # GET/POST/HEAD 요청은 "GET"으로 처리
        add_header "Access-Control-Allow-Origin" "http://zipplz.site";
        add_header "Access-Control-Allow-Methods" "GET, POST, OPTIONS, PUT, DELETE";
        add_header "Access-Control-Allow-Headers" "Origin, Authorization, Accept, Content-Type, X-Requested-With";
        add_header "Access-Control-Max-Age" 1209600;
        add_header "Content-Type" "text/plain; charset=UTF-8";
        add_header "Content-Length" 0;
        return 200;
    }
}

server {
    listen 303 http://zipplz.site;

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/zipplz.site/ssl/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/zipplz.site/ssl/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

    }

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

        root /usr/share/nginx/html;
        index index.html index.htm index.nginx-debian.html;

        server_name example.com; # managed by Certbot

        location / {
            # First attempt to serve request as file, then
            # as directory, then fall back to displaying a 404.
            try_files $uri $uri/ =404;

            # Bypass the proxy if the browser requests .url
            if ($request_uri ~ /\.url) {
                proxy_pass http://$host;
                return 200;
            }

            proxy_pass http://localhost:8080;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;

            add_header Cache-Control "no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0";
            add_header Pragma "no-cache";
            add_header Expires "0";
            proxy_http_version 1.1;
            proxy_pass http://$host;
            proxy_set_header Connection "keep-alive";

            # GET/POST/HEAD 요청은 브라우저에서 요청 처리
            # GET/POST/HEAD 요청은 "GET"으로 처리
            add_header "Access-Control-Allow-Origin" "http://zipplz.site";
            add_header "Access-Control-Allow-Methods" "GET, POST, OPTIONS, PUT, DELETE";
            add_header "Access-Control-Allow-Headers" "Origin, Authorization, Accept, Content-Type, X-Requested-With";
            add_header "Access-Control-Max-Age" 1209600;
            add_header "Content-Type" "text/plain; charset=UTF-8";
            add_header "Content-Length" 0;
            return 200;
        }
    }

    location /api {
        proxy_pass http://localhost:8080;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        add_header Cache-Control "no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0";
        add_header Pragma "no-cache";
        add_header Expires "0";
        proxy_http_version 1.1;
        proxy_pass http://$host;
        proxy_set_header Connection "keep-alive";

        # GET/POST/HEAD 요청은 브라우저에서 요청 처리
        # GET/POST/HEAD 요청은 "GET"으로 처리
        add_header "Access-Control-Allow-Origin" "http://zipplz.site";
        add_header "Access-Control-Allow-Methods" "GET, POST, OPTIONS, PUT, DELETE";
        add_header "Access-Control-Allow-Headers" "Origin, Authorization, Accept, Content-Type, X-Requested-With";
        add_header "Access-Control-Max-Age" 1209600;
        add_header "Content-Type" "text/plain; charset=UTF-8";
        add_header "Content-Length" 0;
        return 200;
    }
}
```

특이사항

- 기존의 도메인은 구매한 새 도메인으로 redirect
- flask/spring boot 서버에 따라 포트번호 프록시 설정 및 pre-flight 고려

프로퍼티 파일 목록

DB 유형

MYSQL

포트 번호

3306

접속할 DB명

ZIP_PLZ

사용자 이름-비밀번호

ssafy-ssafy

JDBC URL

jdbc:mysql://localhost:3306/ZIP_PLZ

Driver

com.mysql.jdbc.Driver

시연 시나리오

1.

(고객)

- 메인페이지에서 원하는 시공을 선택하여 시공사 목록을 조회한다.
- 원하는 시공자를 선택하여 시공사 포트폴리오를 조회한다.
- 하단의 문의하기 버튼을 통해 고객과 해당 시공자의 채팅방이 생성된다.

2.

(시공사)

- 시공자는 생성된 채팅방으로부터 온 메시지를 보고 채팅방에 입장한다.
- 계약서 작성을 진행하기 위해, 화상채팅을 의뢰하고 화상채팅을 시작한다.

3.

(고객 & 시공사)

- 고객과 시공자는 화상채팅을 진행하고, 시공자는 계약서 초안 작성을 동시에 진행한다.
- 계약과 관련한 내용을 화상채팅으로 서로 공유하고, 화상채팅이 종료된다.

4.

(시공사)

- 시공자는 작성된 초안을 채팅방을 통해 고객에게 수락 요청을 보낸다

5.

(고객)

- 고객은 계약서 초안을 검토하고 수락 버튼을 누른다.

6.

(고객)

- 고객의 스케줄러의 달력에 해당 시공 일정이 등록된다.
- 고객의 달력 하단에 시공 일정의 상세 정보가 등록된다.

7.

(시공사)

- 시공자의 포트폴리오 달력에 해당 시공일정이 등록된다.