

# 포팅매뉴얼

서비스명 : SSAFY 교육생들의 자율적 성장을 돕는 플랫폼, SSACLE

팀명 : A402

## 1. Develop Environment

### Back-end



#### version

- **Java Version:** 17 (Configured using Java Toolchain)
- **Spring Boot Version:** 3.4.2
- **Dependency Management Version:** 1.1.7
- **Database:** MySQL (Using `mysql-connector-j`)
- **ORM:** Spring Data JPA with QueryDSL (QueryDSL JPA 5.0.0 Jakarta)
- **Security:** Spring Security, JWT (io.jsonwebtoken:jjwt)
- **Caching & Storage:** Redis, AWS S3 (Amazon SDK)
- **API Documentation:** SpringDoc OpenAPI 2.6.0
- **Web:** Spring Boot Web
- **Testing Frameworks:** JUnit, Spring Security Test
- **Video Conferencing:** OpenVidu (openvidu-java-client:2.31.0)

### 프로젝트 실행하는 방법

#### 1. Git clone

```
git clone https://lab.ssafy.com/s12-webmobile1-sub1/S12P11A402.git
```

## 2. 폴더 이동

```
cd S12P11A402
```

## 3. 브랜치 변경

```
git checkout -t origin BE/dev
```

## 3. 작업 폴더 이동

```
cd back/ssacle
```

## 3. IntelliJ IDE 실행 및 의존성 설치

```
# 현재 폴더  
idea .
```

```
gradle
```

## 4. 상단 실행 버튼 눌러 실행

## +++ 빌드

```
./gradlew clean build -x test
```

# Frontend

## ! version

- **nodeJs** : 22.
- **Vite** : 6.0.5
- **React** : 19.0.0
- **Typescript** : 5.7.3
- React Router : 7.1.3
- Tailwind CSS : 3.4.17
- Tanstack Query : 5.64.2
- Zustand : 5.0.3
- openvidu-browser : 2.31.0
- Axios : 1.7.9
- Lodash : 4.17.21

## 실행 방법

## 프로젝트 실행하는 방법

### 1. Git clone

```
git clone https://lab.ssafy.com/s12-webmobile1-sub1/S12P11A402.git
```

## 2. 폴더 이동

```
cd S12P11A402
```

## 3. 브랜치 변경

```
git checkout -t origin FE/dev
```

## 3. 작업 폴더 이동

```
cd frontend
```

## 1. node module 설치

```
npm install
```

## 2. 실행

```
npm run dev
```

# Infra

### ! Version

- Docker : **27.5.1**
- Docker compose : **v2.32.4**
- Jenkins : 2.495
- Nginx **1.27.4**
- openvidu sever **2.31.0**

### ! 주요 포트 번호

| 서비스         | Port 번호     |
|-------------|-------------|
| spring boot | 8080        |
| React       | 80, 443     |
| MySQL       | 3306        |
| Jenkins     | 9090        |
| openvidu    | 8442, 8443, |
| redis       | 6379        |

## openvidu 실행하기 (EC2)

### 1. 폴더 이동

```
cd /opt/openvidu
```

## 2. 실행하기

```
./openvidu start
```

### docker-compose.yml 파일

- 파일 구조

```
/opt/openvidu

/home/ubuntu/deploy/
| — S12P11A402 # 프로젝트 Git repository
| — frontend
|   └─ docker-compose.yml
| — backend
|   └─ docker-compose.yml
|     └─ backend.env
| — nginx
|   └─ docker-compose.yml
|     └─ nginx.conf
| — mysql
|   └─ docker-compose.yml
|
```

#### frontend `docker-compose.yml`

```
services:
  frontend:
    build:
      context: /home/ubuntu/deploy/S12P11A402/frontend # Git repository
      dockerfile: Dockerfile
    container_name: frontend-container
    restart: always
    networks:
      - ssacle_network
    healthcheck:
      test: ['CMD', 'ls', '/usr/share/nginx/html/index.html']
      interval: 30s
      timeout: 10s
      retries: 3

networks:
  ssacle_network:
    external: true
```

#### backend `docker-compose.yml`

```
version: "3.8"

services:
  backend:
    build:
```

```
context: /home/ubuntu/deploy/S12P11A402/back/ssacle
dockerfile: Dockerfile
container_name: backend-container
restart: always
env_file:
  - backend.env
networks:
  - ssacle_network
```

```
redis:
  image: redis:latest
  container_name: redis-container
  restart: always
  ports:
    - "6379:6379"
  networks:
    - ssacle_network
```

```
networks:
  ssacle_network:
    external: true
```

Nginx `docker-compose.yml`

```
services:
  nginx:
    image: nginx:latest
    container_name: nginx-container
    restart: always
    ports:
      - '80:80'
      - '443:443' # SSL
      - '8080:8080' # 개발 서버
    volumes:
      - /home/ubuntu/deploy/nginx/nginx.conf:/etc/nginx/nginx.conf:ro
      - /etc/letsencrypt:/etc/letsencrypt
    networks:
      - ssacle_network

networks:
  ssacle_network:
    external: true
```

`nginx.conf`

```
worker_processes auto; # CPU 코어 수에 맞게 자동 조정

events {
  worker_connections 1024;
}

http {

  server {
    listen 80;
    server_name i12a402.p.ssafy.io;

    # Certbot 인증서 갱신을 위한 경로
```

```

location /.well-known/acme-challenge/ {
    root /var/www/certbot;
}

# 80 포트로 접속 시 443 포트로 리다이렉트
location / {
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name i12a402.p.ssafy.io;

    ssl_certificate /etc/letsencrypt/live/i12a402.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/i12a402.p.ssafy.io/privkey.pem;

    location / {
        proxy_pass http://frontend:80;
        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 https;

    }

    # 프론트 - 백엔드 연결
    location /api/ {
        proxy_pass http://backend:8080/api;
        proxy_set_header Origin "";
        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 https;
    }

    location /swagger-ui/ {
        proxy_pass http://backend:8080/swagger-ui/;
        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 https;
    }

}

# 개발을 위한 포트 개방
server {
    listen 8080;

    location / {
        proxy_pass http://backend: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 https;
    }
}

```

```
}  
}
```

mysql `docker-compose.yml`

```
services:  
  db:  
    image: mysql:8.0  
    container_name: mysql-container  
    restart: always  
    environment:  
      MYSQL_ROOT_PASSWORD: Ssacle1234!  
      MYSQL_DATABASE: ssacle  
      MYSQL_USER: ssacle  
      MYSQL_PASSWORD: ssacle  
    ports:  
      - "3306:3306"  
    volumes:  
      - mysql_data:/var/lib/mysql  
    networks:  
      - ssacle_network  
  
volumes:  
  mysql_data:  
  
networks:  
  ssacle_network:  
    external: true
```

## 배포 환경 실행 가이드

### Nginx 실행

```
# 배포 환경 이동  
cd ~/deploy/nginx  
  
# docker compsoe 파일 실행  
docker compose up -d --build
```

### Frontend

```
# Git repository 이동  
cd ~/deploy/S12P11A402/frontend  
  
# git branch 변경  
git checkout -t origin FE/dev  
  
# git 최신 버전  
git pull origin FE/dev  
  
# 배포 환경 이동  
cd ~/deploy/frontend  
  
# docker compsoe 파일 실행  
docker compose up -d --build
```

### Backend

```
# Git repository 이동
cd ~/deploy/S12P11A402

# git branch 변경
git checkout -t origin BE/dev

# Git repository 이동
cd ~/deploy/S12P11A402/back/ssacle

# git 최신 버전
git pull origin BE/dev

# 배포 환경 이동
cd ~/deploy/backend

# docker compsoe 파일 실행
docker compose up -d --build
```

## 2. Setting

### Back-end

- `backend.env`

```
DB_URL=jdbc:mysql://db:3306/ssacle?useSSL=false&allowPublicKeyRetrieval=true
DB_USERNAME=root
DB_PASSWORD=****
JWT_SECRET_KEY=****
ACCESS_EXPIRE_MS=3600000
REFRESH_EXPIRE_MS=86400000
AWS_S3_ACCESSKEY=***
AWS_S3_SECRETKEY=***
AWS_S3_REGION=ap-northeast-2
AWS_S3_BUCKET=ssacle
MM_WEBHOOK_URL= ****
OPENVIDU_URL=https://i12a402.p.ssafy.io:8443/
OPENVIDU_SECRET=****
NOTION_API_KEY=****
NOTION_API_URL=https://api.notion.com/v1/pages
NOTION_MAIN_PAGE_ID=****
DATABASE_ID=****
REDIS_HOST=redis
REDIS_PORT=6379
GPT_API_KEY=****
```