



포팅 매뉴얼

프로젝트 사용 도구

형상 관리: Gitlab

이슈 관리: Jira

커뮤니케이션: Mattermost, Notion

디자인: Figma

UCC: PPT, 모바비

CI/CD: Jenkins

개발환경

- IntelliJ: 2023.3.2
- Springboot: 3.2.2
- VS Code: 1.85.1
- Vue3: 3.4.15
- Vite: 5.0.10
- Node.js: 20.10.0
- JVM: 17.0.9 (
- SERVER: AWS EC2 Ubuntu 20.04.6 LTS

- MySQL: 8.0.34

외부 서비스

MongoDB: application.yml에 해당 내용 있음

GitIgnore 처리한 파일

Vue: .env

Spring boot: application.yml (\src\main\resources 또는 classPath에 위치)

환경변수 형태

.env

```
VITE_APP_OPEN_VIDU_API_URL = 오픈비두 서버주소
```

application.yml

```
# kafka setting
kafka:
  server: 카프카 서버 주소
  group-id: 그룹아이디
  offset: latest

# Mongo DB
data:
  mongodb:
    uri: mongoDB uri

# mySQL
datasource:
  url: mySQL uri
  username: mySQL 아이디
```

```
password: mySQL 비밀번호
driver-class-name: com.mysql.cj.jdbc.Driver

# myBatis
mybatis:
  mapper-locations: classpath:/mapper/**/*.xml
```

실행하기

1. Front

해당 프로젝트 위치로 이동

- project Setup
npm install
- Compile and Hot-Reload for Development
npm run dev

2. BackEnd

해당 프로젝트 위치로 이동

- project Setup
gradle 실행
- Compile
Run 실행

3. Kafka

- Kafka-compose.yml

```
version: '3.8'
services:
  zookeeper:
    image: wurstmeister/zookeeper
    container_name: zookeeper
    ports:
      - '2181:2181'
  kafka:
```

```
image: wurstmeister/kafka
container_name: kafka
ports:
  - '9092:9092'
environment:
  KAFKA_ADVERTISED_HOST_NAME: 127.0.0.1
  KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
volumes:
  - /var/run/docker.sock:/var/run/docker.sock
```

- powershell로 해당 폴더까지 옴
- docker-compose -f kafka-compose.yml up

빌드하기

1. Front

해당 프로젝트 위치로 이동

- project Setup
npm install
- Compile and Minify for Production
npm run build

2. BackEnd

- gradle 실행
- Bootjar 실행

배포하기

Nginx 설정

```
server {
    listen 80;
    # front
    location / {
        root /usr/share/nginx/html;
        # index index.html index.htm
    }
}
```

```
        try_files $uri $uri/ /index.html;  
    }  
}
```

- `sudo service nginx start`