

# 포팅 매뉴얼

## 1. 프로젝트 기술 스택 - 버전

---

- Java - 1.8
- Spring Boot - 2.7.2
- JPA
- MySQL - 8.0.30
- Vue.js - 3.2.37
- openvidu - 2.22.0
- nginx - 1.18.0
- docker - 20.10.17
- aws - ubuntu 20.04
- GitLab
- Jira

## 2. gitignore, properties, env

---

- Spring Boot gitignore

```
HELP.md
.gradle
build/
!gradle/wrapper/gradle-wrapper.jar
!**/src/main/**/build/
!**/src/test/**/build/

### STS ###
.apt_generated
.classpath
.factorypath
.project
.settings
.springBeans
.sts4-cache
bin/
!**/src/main/**/bin/
!**/src/test/**/bin/

### IntelliJ IDEA ###
.idea
*.iws
*.iml
*.ipr
out/
!**/src/main/**/out/
!**/src/test/**/out/

### NetBeans ###
/nbproject/private/
/nbbuild/
/dist/
/nbdist/
/.nb-gradle/

### VS Code ###
.vscode/

### application.properties ###
**/application.properties

**/social.properties
```

```
# 빌드 시스템에서 자동 생성된 폴더
**/src/main/resources/static/

# log
ssafy-web.log*
```

- application.properties

```
#it will be set build date by gradle. if this value is @build.date@, front-end is development mode
build.date=@build.date@
server.port=8080
server.address=localhost
server.servlet.contextPath=/
# Charset of HTTP requests and responses. Added to the "Content-Type" header if not set explicitly.
server.servlet.encoding.charset=UTF-8
# Enable http encoding support.
server.servlet.encoding.enabled=true
# Force the encoding to the configured charset on HTTP requests and responses.
server.servlet.encoding.force=true

# for SPA
spring.resources.static-locations=classpath:/dist/
spa.default-file=/dist/index.html
spring.mvc.throw-exception-if-no-handler-found=true
spring.resources.add-mappings=false

# Swagger
springfox.documentation.swagger.use-model-v3=false

#database
spring.jpa.hibernate.naming.implicit-strategy=org.springframework.boot.orm.jpa.hibernate.SpringImplicitNamingStrategy
spring.jpa.hibernate.naming.physical-strategy=org.springframework.boot.orm.jpa.hibernate.SpringPhysicalNamingStrategy
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect
spring.data.web.pageable.one-indexed-parameters=true
spring.datasource.url=jdbc:mysql://13.124.166.107:3306/utf8?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTimeBehavior=convertToNull
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.hikari.username=a701
spring.datasource.hikari.password=utf8a701!
```

```
# jwt
jwt.secret=dyAeHub00c8Ka0fYB6XEQoEj1QzRlVgtjNL8PYs1A1tymZvvqkcEU7L11mkKHeDa
# unit is ms. 15 * 24 * 60 * 60 * 1000 = 15days
jwt.expiration=1296000000

#logging
logging.file.name=./ssafy-web.log
logging.level.root=INFO
logging.level.com.samsung.security=DEBUG
logging.level.org.springframework.web=DEBUG
logging.level.org.apache.tiles=INFO
logging.level.org.springframework.boot=DEBUG
logging.level.org.springframework.security=DEBUG

spring.devtools.livereload.enabled=true

#gzip compression
server.compression.enabled=true
server.compression.mime-types=application/json,application/xml,text/html,text/xml,text/plain,application/javascript,text/css

#for health check
management.servlet.context-path=/manage
management.health.db.enabled=true
management.health.default.enabled=true
management.health.diskspace.enabled=true
```

- social.properties

```

KAKAO_REST_API_KEY=790ccad61f70f651660c30495f5020e1
KAKAO_REDIRECT_URI=http://localhost:3000/redirect

NAVER_CLIENT_ID=QBcghV_7ghmMTkJBAYNr
NAVER_CLIENT_SECRET=BY9_may6aW
NAVER_CALLBACK_URL=http://localhost:3000/redirect

GOOGLE_CLIENT_ID=559017688233-mjteopl8jj7akcc85tp72dlhi1vol7r4.apps.googleusercontent.com
GOOGLE_CLIENT_SECRET=60CSPX-od0E0GrDCzmILYrZ_AU2PJz8b7oQ
GOOGLE_REDIRECT_URI=http://localhost:3000/redirect

```

- Vue gitignore

```

.DS_Store
node_modules
/dist

# local env files
.env.local
.env.*.local
.env

# Log files
npm-debug.log*
yarn-debug.log*
yarn-error.log*
pnpm-debug.log*
*.log

# Editor directories and files
.idea
.vscode
*.suo
*.ntvs*
*.njsproj
*.sln
*.sw?

# node_modules
node_modules/

```

- .env

```

VUE_APP_KAKAO_REST_API_KEY=790ccad61f70f651660c30495f5020e1
VUE_APP_KAKAO_REDIRECT_URI=http://localhost:3000/redirect

VUE_APP_NAVER_CLIENT_ID=QBcghV_7ghmMTkJBAYNr
VUE_APP_NAVER_CLIENT_SECRET=BY9_may6aW
VUE_APP_NAVER_CALLBACK_URL=http://localhost:3000/redirect

VUE_APP_GOOGLE_CLIENT_ID=559017688233-mjteopl8jj7akcc85tp72dlhi1vol7r4.apps.googleusercontent.com
VUE_APP_GOOGLE_REDIRECT_URI=http://localhost:3000/redirect

VUE_APP_API_BASE_URL=http://localhost:8080
VUE_APP_OPENVIDU_BASE_URL=https://i7a701.p.ssafy.io:8442

VUE_APP_LECTURE_ROOM_URL=http://localhost:3000/lecture/

```

### 3. Openvidu

#### 설치

```
# 관리자 권한으로 실행해야 설치 가능
sudo su
```

```
# openvidu 설치를 위한 경로이동
cd /opt
```

```
# openvidu 설치
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_2.22.0.sh | bash
```

```
# openvidu 폴더 이동
cd openvidu
# openvidu 설정 파일
nano .env
# openvidu docker run
./openvidu start
```

## 설정

```
DOMAIN_OR_PUBLIC_IP={서버 URL}
# openvidu 서버 암호 -> api 호출시 필요
OPENVIDU_SECRET=MY_SECRET
# cetbot 사용을 위한 세팅
CERTIFICATE_TYPE=letsencrypt
LETSencrypt_EMAIL={이메일 주소}

# restApi 호출 포트
HTTP_PORT=8441
HTTPS_PORT=8442

# 녹화기능 사용
OPENVIDU_RECORDING=true
# 암호없이 녹화파일 접근
OPENVIDU_RECORDING_PUBLIC_ACCESS=true
```

## 실행

```
./openvidu start
```

## custom-layout

```
cd /opt/openvidu/custom-layout
```

- index.html과 openvidu-browser-2.22.0.min 넣어두기

## 4. nginx

### 설치 및 기본 세팅

```
apt-get install nginx
```

```
# nginx 설정파일 위치
cd /etc/nginx/sites-available
```

```
sudo vim {도메인}
```

```
sudo ln -s /etc/nginx/sites-available/{도메인} /etc/nginx/sites-enabled/{도메인}
```

## SSL

```
#certbot 설치를 위한 레포지토리 등록  
sudo add-apt-repository ppa:certbot/certbot
```

```
# certbot 설치  
sudo apt-get install python-certbot-nginx
```

```
sudo certbot --nginx -d {도메인}
```

이후 질문에 답변 SSL 적용완료

## 세팅

```
server_name {도메인};  
# /api 요청을 backend 서버로 전달  
location /api {  
    proxy_pass http://localhost:8080;  
}  
  
location / {  
    # npm run build로 생긴 dist 폴더 경로  
    root /home/ubuntu/S07P12A701/FE/utf/dist/;  
    index index.html index.nginx-debian.html;  
    try_files $uri $uri/ /index.html;  
}
```

```
# nginx 재시작  
sudo service nginx restart
```

## 5. 소셜 로그인

- 카카오
  - 애플리케이션 추가

전체 애플리케이션 (2)

애플리케이션 이름



애플리케이션 추가하기

- 로그인 활성화

### 활성화 설정

상태

ON

카카오 로그인 API를 활용하면 사용자들이 번거로운 회원 가입 절차 대신, 카카오톡으로 서비스를 시작할 수 있습니다.  
상태가 OFF일 때도 카카오 로그인 설정 항목을 변경하고 서버에 저장할 수 있습니다.  
상태가 ON일 때만 실제 서비스에서 카카오 로그인 화면이 연결됩니다.

카카오 로그인 ON

[동의 화면 미리보기](#)

### 활성화 설정

상태

ON

카카오 로그인 API를 활용하면 사용자들이 번거로운 회원 가입 절차 대신, 카카오톡으로 서비스를 시작할 수 있습니다.  
상태가 OFF일 때도 카카오 로그인 설정 항목을 변경하고 서버에 저장할 수 있습니다.  
상태가 ON일 때만 실제 서비스에서 카카오 로그인 화면이 연결됩니다.

#### ◦ Redirect URI 설정

### Redirect URI

삭제

수정

Redirect URI

http://localhost:3000/redirect  
https://i7a701.p.ssafy.io/redirect

- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)
- REST API로 개발하는 경우 필수로 설정해야 합니다.

#### • 네이버

##### ◦ 서비스 URL, Callback URL 등록

PC 웹

×

⤴

서비스 URL

https://i7a701.p.ssafy.io

서비스 URL예시: (O) http://naver.com (X) http://www.naver.com

서비스 URL값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다.

불법/음란성 사이트 등 이용약관에 위배되는 사이트의 경우, 이용이 제한될 수 있습니다.

서비스하려는 사이트 URL과 동일한 사이트 URL로 해주셔야 **네이버 로그인 뱃지**가 노출됩니다.

네이버 로그인

Callback URL (최대 5개)

http://localhost:3000/redirect

-

https://i7a701.p.ssafy.io/redirect

+

텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.

Callback URL은 네이버 로그인 후 이동할 페이지 URL입니다. Callback URL값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다.

입력한 주소와 다른 Callback URL로 리다이렉트 될 경우, 이용이 제한될 수 있습니다.

## 애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 **내 애플리케이션** 메뉴의 서버 메뉴에 등록하신 애플리케이션 이름으로 서버 메뉴가 만들어집니다.

애플리케이션 이름

애플리케이션 이름

네이버 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 기급적 10자 이내로 간결하게 설정해주세요.

40자 이내의 영문, 한글, 숫자, 공백문자, 쉼표(,), "/", "-", "\_", 만 입력 가능합니다.

- 구글
  - OAuth 클라이언트 ID 발급

+ CREATE CREDENTIALS
DELETE

API 키

할당량과 액세스 권한을 확인하기 위해 간단한 API 키로 프로젝트를 확인합니다.

OAuth 클라이언트 ID

앱에서 사용자 데이터에 액세스할 수 있도록 사용자 동의를 요청합니다.

서비스 계정

로봇 계정을 사용하여 서버 간의 앱 수준 인증을 사용 설정합니다.

사용자 인증 정보 선택 도움말

사용할 사용자 인증 정보의 유형을 결정할 수 있도록 몇 가지 질문을 합니다.

◦ URI 등록

## 승인된 자바스크립트 원본 ?

브라우저 요청에 사용

URI 1 \*

http://localhost

URI 2 \*

http://localhost:3000

+ URI 추가

## 승인된 리디렉션 URI ?

웹 서버의 요청에 사용

URI 1 \*

http://localhost:3000/redirect

URI 2 \*

https://i7a701.p.ssafy.io/redirect

+ URI 추가

◦ 테스트 사용자 등록



## 테스트 사용자

+ ADD USERS

필터 속성 이름 또는 값 입력



사용자 정보	
hh950131@gmail.com	
koxogml@gmail.com	
thjsll11@gmail.com	
yangksks@gmail.com	

### ◦ 도메인 등록

## 앱 도메인

나와 내 사용자를 보호하기 위해 Google에서는 OAuth를 사용하는 앱만 승인된 도메인을 이용할 수 있도록 허용합니다. 다음 정보가 동의 화면에서 사용자에게 표시됩니다.

애플리케이션 홈페이지

사용자에게 홈페이지 링크를 제공합니다.

애플리케이션 개인정보처리방침 링크

사용자에게 공개 개인정보처리방침 링크를 제공합니다.

애플리케이션 서비스 약관 링크

사용자에게 공개 서비스 약관 링크를 제공합니다.

## 승인된 도메인

동의 화면 또는 OAuth 클라이언트 구성에서 도메인이 사용되면 여기에서 사전 등록해야 합니다. 앱이 인증을 거쳐야 하는 경우 [Google Search Console](#)로 이동하여 도메인이 승인되었는지 확인하세요. 승인된 도메인 한도에 대해 [자세히 알아보세요](#).

승인된 도메인 1 \*

승인된 도메인 2 \*

## 6. Mysql

```
# 업데이트 및 HTTP 패키지 설치
sudo apt update
```

```
sudo apt-get install -y ca-certificates \
    curl \
    software-properties-common \
    apt-transport-https \
    gnupg \
    lsb-release
```

```
# GPG 키 및 저장소 추가
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

```
# 도커 설치
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io
```

## Mysql 설치

```
docker pull mysql
```

```
# mysql 컨테이너 생성
docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=비밀번호 -d -p 3306:3306 mysql
```

```
# 도커 컨테이너 접속
docker exec -it mysql bash
```

## Mysql 세팅

```
mysql -u root -p
```

```
CREATE DATABASE utf8;

# 계정생성 및 외부접근 권한 부여
create user 계정이름@'%' identified by '비밀번호';
# 권한내용 메모리에 반영
flush privileges;
```

## 7. 빌드 및 배포 명령어

```
# gradle 빌드
./gradlew build

# jar 실행(백그라운드)
nohup java -jar fileName.jar &

# 프로세스 확인
ps -ef

# 프로세스 종료
kill -9 PID
```

```
# vue 빌드  
npm run build
```