

포팅 매뉴얼

602 박종선, 윤혜진, 이경무, 한재승, 허설

버전 정리

gradle 7.5

node 16.16

npm 8.17.0

mysql 8.0

openvidu 2.22.0

redis 5.0.7

JAVA_HOME = /usr/lib/jvm/java-8-openjdk-amd64

사용된 포트 번호

front-Vue : 7602

back-Springboot : 8080

Openvidu : 4443 (그외에 사용되는 포트 다수)

mysql : 3306

Redis : 6379

젠킨스 계정

hseol

gjtjFWkd1!

그외의 정보들은 Properties 에 담겨있다.

properties

```
#it will be set build date by gradle. if this value is @build.date@, front-end is development mode

server.port=8080

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

# Swagger
springfox.documentation.swagger.use-model-v3=false
spring.mvc.pathmatch.matching-strategy = ANT_PATH_MATCHER

spring.jpa.show-sql=true
#MySQL

spring.datasource.url=jdbc:mysql://17a602.p.ssafy.io:3306/jangbo?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTimeBehavior=convert
#spring.datasource.url=jdbc:mysql://localhost:3306/jangbo?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTimeBehavior=convertToNull&

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.hikari.username=root
spring.datasource.hikari.password=jangbo602

#MySQL
spring.jpa.database=mysql
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.show_sql=true
spring.jpa.hibernate.naming.strategy=org.hibernate.cfg.ImprovedNamingStrategy

#logging
#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
#logging.level.org.hibernate.SQL=debug

#Redis
spring.cache.type=redis
spring.redis.host=i7a602.p.ssafy.io
spring.redis.port=6379

# JWT Secret Key
spring.jwt.secret=parkjongsunparkjongsunparkjongsunparkjongsunparkjongsunparkjongsunparkjongsunparkjongsun

#S3 Bucket
# AWS Account Credentials
cloud.aws.credentials.accessKey=AKIA4AH6N5ICTV77QQW2
cloud.aws.credentials.secretKey=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

# AWS S3 bucket Info (S3 ????)
cloud.aws.s3.bucket=jangbo-image
cloud.aws.region.static=ap-northeast-2
cloud.aws.stack.auto=false

##AWS S3 Bucket URL ????.?
cloud.aws.s3.bucket.url=https://jangbo-image.s3.ap-northeast-2.amazonaws.com/key-name

# file upload max size
spring.servlet.multipart.max-file-size=20MB
spring.servlet.multipart.max-request-size=20MB

#ssl certificates
server.ssl.enabled=true
server.ssl.key-store = classpath:keystore.p12
server.ssl.key-store-password = jangbo602
server.ssl.key-password = jangbo602
server.ssl.keyStoreType = PKCS12
```

오픈비두

오픈비두를 돌려보기 위해서 사전에 [도커](#) 와 [도커 컴포즈](#) 설치 필요

`cd /opt` :apt-get으로 다운 받을 수 없는 서비스를 주로 모아놓는 디렉토리

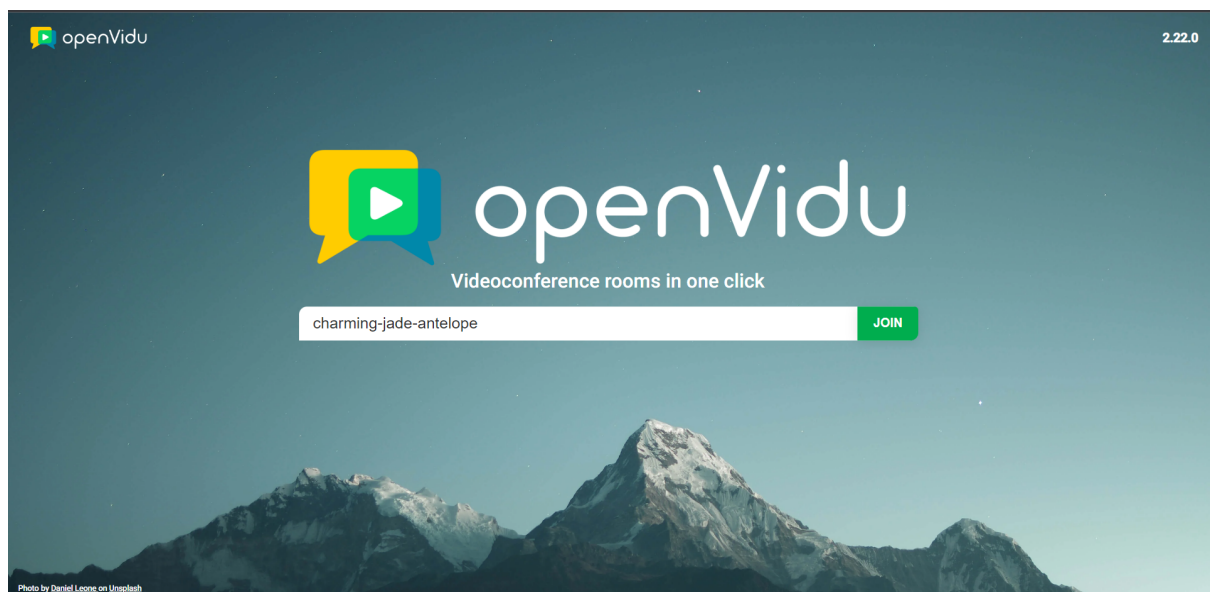
```
sudo curl https://s3-eu-west-1.amazonaws.com/aws.opnvidu.io/install_opnvidu_2.22.0.sh | bash
```

추천하는건 오픈비두 먼저 돌려본 다음에 포트 번호 변경

```
cd /opt/opnvidu
```

온프레미스 실행

```
./opnvidu start
```



실행 후 다시 정지

```
./openvidu stop
```

`sudo vi .env` 그 이후 환경설정에 들어가서 원하는 포트로 변경

안되는 포트 번호를 피해서 설정 (공식문서 참고)

순서는 상관없지만 만약 서버가 닫혀있다면

스프링부트 서버 실행

```
java -jar backend-0.0.1-SNAPSHOT.jar
```

오픈비두 서버 실행 (도커 컨테이너)

```
cd /opt/openvidu/ ./openvidu start
```

코드 변경 시

```
sudo nginx -t
```

 문법 검사

```
sudo service nginx status|start|stop|restart
```

 비슷한 문법으로 젠킨스 도 확인 가능

- 레디스는 먼저 설치하지 말고 오픈비두와 돌려봤을 때 인식된다면 그걸 사용할 것.

젠킨스

https ://i7a602.p.ssafy.io:9090

크롬 시크릿모드에서 될 확률이 제일 높다(조금 많은 코드가 유입되면 빌드되면서 죽어버린다..)

시크릿모드에서 계정 연결이 안되어있을때는 hseol gjtjFWkd1! 이걸로 하면 됨

```
sudo service jenkins restart
```

dist 파일 위치 `/var/lib/jenkins/workspace/front-back-build-run/frontend/dist`

엔진엑스에 이렇게 있고 젠킨스가 에러나면 위치를 `/var/www/html` 로 바꾼다.

엔진엑스

오픈비두를 먼저하는 것을 추천!!

프론트와 백을 같은 도메인에서 포트번호를 나눠주기 위한 웹서버

심볼릭링크를 이용해서 연결했고 코드 수정은 `/etc/nginx/sites-available/default` 에서만 했음

```
server {
    root /var/www/html/dist/;
    #root /var/lib/jenkins/workspace/front-back-build-run/frontend/dist;
    server_name i7a602.p.ssafy.io;
    location / {

        try_files $uri $uri/ =404;

    }

    location /be {
        proxy_pass https://i7a602.p.ssafy.io:8080;
        #proxy_redirect off;

        charset utf-8;
        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;
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Forwarded-Port $server_port;

    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/i7a602.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/i7a602.p.ssafy.io/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 {
    if ($host = i7a602.p.ssafy.io) {
```

```
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    listen [::]:80;
    server_name i7a602.p.ssafy.io;
    return 404; # managed by Certbot

}
```