

포팅 매뉴얼



[A501] We See Play

SSAFY 서울 캠퍼스 7기

공동프로젝트 [2022.07.08 ~ 2022.08.19]

원찬호[팀장], 김현영, 김희성, 이도경, 조혜림, 홍제민

목차

1. 프로젝트 기술 스택

2. 빌드

3. AWS EC2 설정

4. DB 설정

5. 배포

6. 외부 서비스

7. 기타 참고사항

1.프로젝트 기술 스택

개발 환경

■ OS

- ☐ Windows 10

■ IDE

- ☐ **Visual Studio Code** ver.1.70.0
- ☐ **eclipse IDE** 2020-06 ver.4.16.0
- ☐ **Spring Tools 3** ver3.9.14.RELEASE

■ Database

- ☐ **MySQL Workbench** 8.0 CE

■ Server

- ☐ **AWS EC2 Ubuntu** ver.20.04.4 LTS
- ☐ **Docker** ver.20.10.17
- ☐ **Nginx** ver.18.0

기술 스택 버전

■ Frontend

- ☐ **HTML5, CSS3, Javascript**(ES6)
- ☐ **Node.js** ver.16.16.0
- ☐ **Vue** ver.3.2.38
- ☐ **Vue-cli** ver.5.0.8, **Vuex** ver.4.0.2
- ☐ **Vue-router** ver.4.1.2, **Vue-axios** ver.3.4.1

- ☐ **Sweetalert2** ver11.4.24
- ☐ **quasar** ver.2.7.5
- ☐ **openvidu-browser** ver.2.220

■ Backend

- ☐ **Java openjdk** ver.1.8.0.332
- ☐ **Spring Boot** ver.2.4.5
- ☐ **Gradle** ver.6.7
- ☐ **Querydsl** ver.4.4.0
- ☐ **Lombok** ver.1.18.24

■ DB

- ☐ **MySQL** ver.5.7.38

■ 형상관리

- ☐ Gitlab

■ 이슈 관리

- ☐ Jira

■ 커뮤니케이션

- ☐ Notion, MatterMost

2. 빌드

■ Frontend 빌드

- package.json 에 포함된 의존성 패키지들을 설치해준다.

```
npm install
```

- frontend 폴더에 .env 파일을 넣어준다. (7. 기타 참고사항' 참고)
- frontend 폴더에서 빌드를 실행한다.

```
npm run build
```

- 빌드에 성공하면 dist 폴더가 생성된다

■ Backend 빌드

- User Server (backend 폴더), Room Server (backend-openvidu 폴더) 두 개의 서버로 나뉘어져 있다.
- 각 폴더의 src/main/resources 폴더로 들어가서 keystore.p12 파일을 넣어준다. (7. 기타 참고사항 참고)
- 각 폴더에서 이미 빌드 된 폴더가 남아있을 수도 있으니 삭제 후 빌드를 수행한다.

```
./gradlew clean
```

```
./gradlew build
```

- 빌드에 성공하면 build 폴더가 생성되고 build/libs 폴더에 jar파일이 생성된다.

3.AWS EC2 설정

■ EC2 서버 시간 설정

```
sudo timedatectl set-timezone 'Asia/Seoul'
```

■ 자바 설치

```
sudo apt-get update  
  
sudo apt-get install openjdk-8-jdk  
  
java -version
```

자바 환경변수 설정

```
sudo vi /etc/profile
```

profile파일 편집 모드가 켜지면 아래 3줄을 추가하고 저장해준다.

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64  
  
export PATH=$JAVA_HOME/bin:$PATH  
  
export CLASS_PATH=$JAVA_HOME/lib:$CLASS_PATH
```

적용시키고 리부트

```
source /etc/profile  
  
sudo reboot now  
  
echo $JAVA_HOME
```

■ nodejs 설치 후 확인

```
curl -sL https://deb.nodesource.com/setup_16.x | sudo bash -  
  
cat /etc/apt/sources.list.d/nodesource.list  
  
sudo apt -y install nodejs  
  
$ node -v
```

■ Docker 설치

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

■ Nginx 설치

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install nginx
```

■ snapd 설치

snapd로 certbot을 설치하는 것을 권장하기 때문에 snapd를 먼저 설치한다.

```
sudo apt update
```

```
sudo apt install snapd
```

■ certbot 설치

snapd를 이용해서 certbot 설치하기

```
sudo snap install --classic certbot
```

certbot이 잘 설치되었는지 확인

```
certbot --version
```

■ 인증서 발급받기

```
certbot certonly --nginx -d i7a501.p.ssafy.io
```

인증서 발급에 성공하면 /etc/letsencrypt/live 위치에 인증서가 위치한다.

4.DB 설정

■ 도커를 이용해서 MySQL 실행하기

```
sudo docker pull mysql:5.7.38
```

■ MySQL docker 컨테이너 생성 및 실행

```
sudo docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=fx5nqu8r4k -d -p 3900:3306
```

■ MySQL docker 컨테이너 접속

```
sudo docker exec -it mysql-container bash
```

■ MySQL 로그인

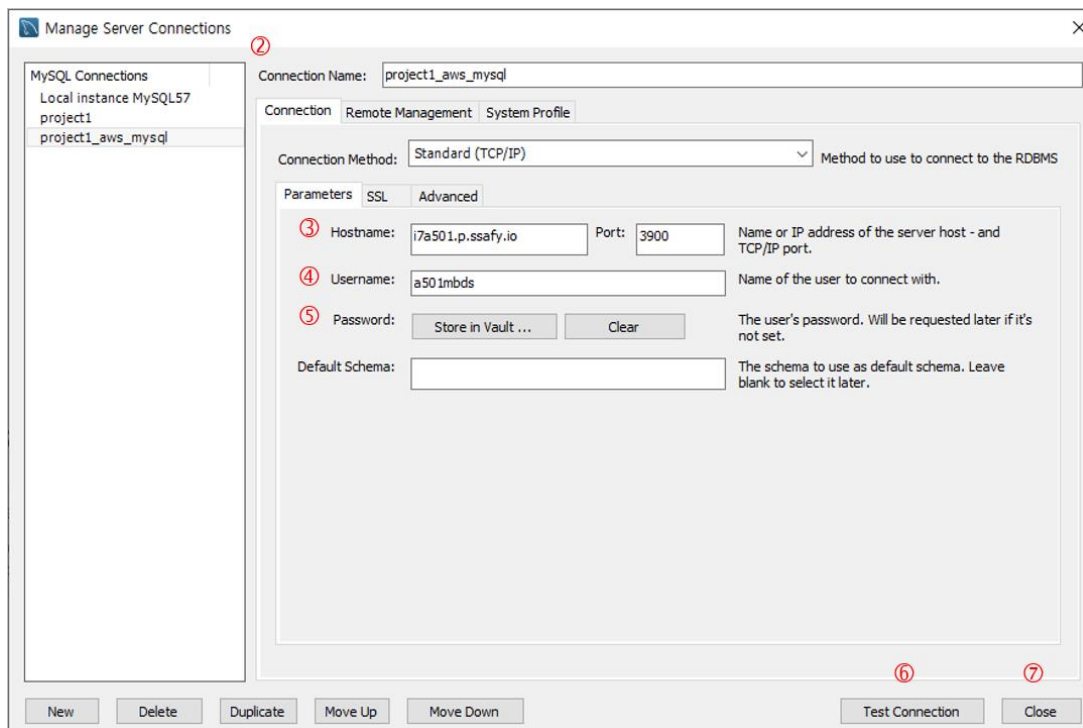
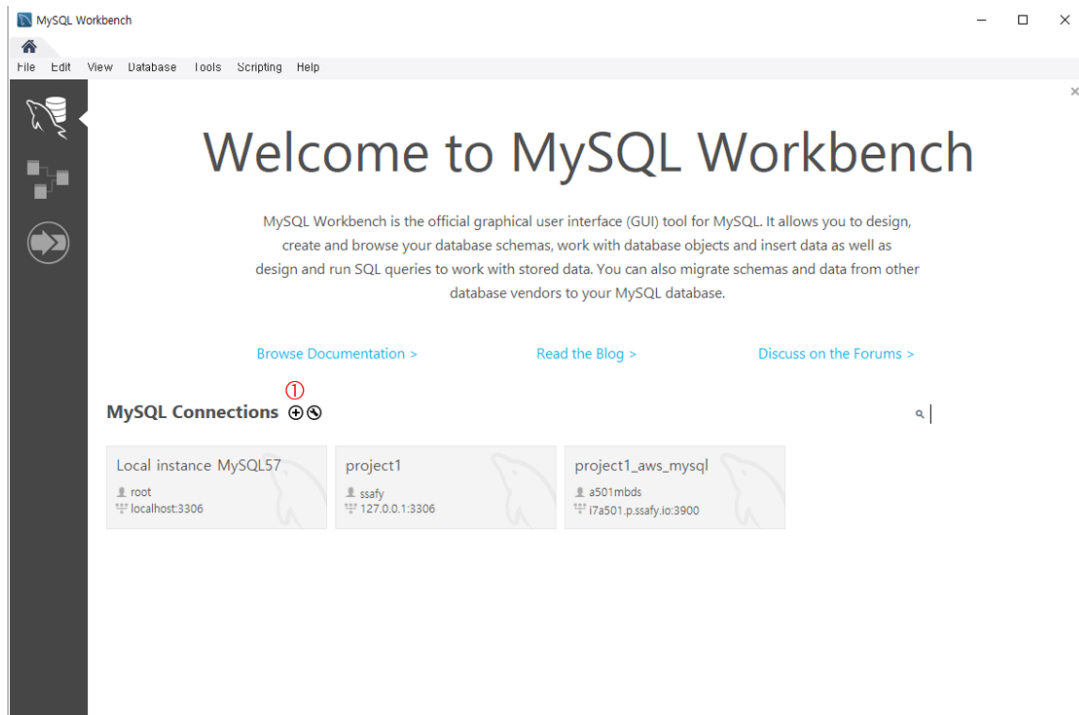
```
mysql -u root -p
```

최초 접속 시 사용할 비밀번호 입력 [매뉴얼 기준: 비밀번호 : fx5nqu8r4k]

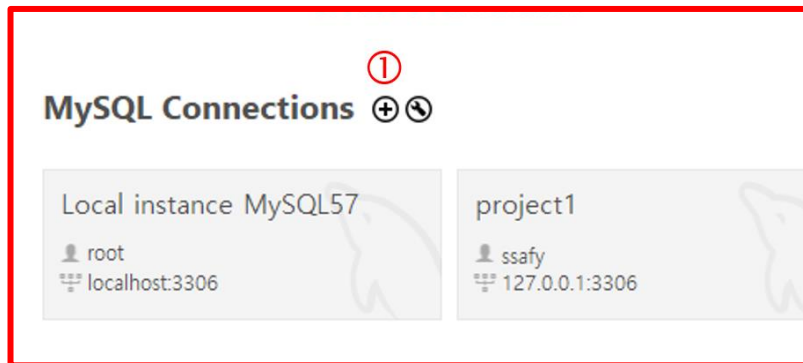
■ root 유저명 바꾸기

```
use mysql;  
  
update user set user='a501mbds' where user='root';  
  
flush privileges;  
  
exit
```

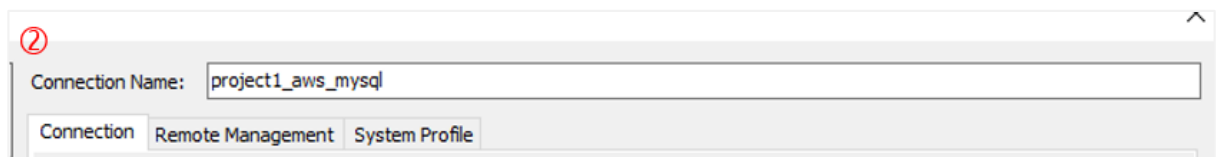
■ MySQL Workbench 접속 방법 (순서: 1 ~ 6)



1. MySQL Connections 옆 [플러스(+)] 버튼 클릭

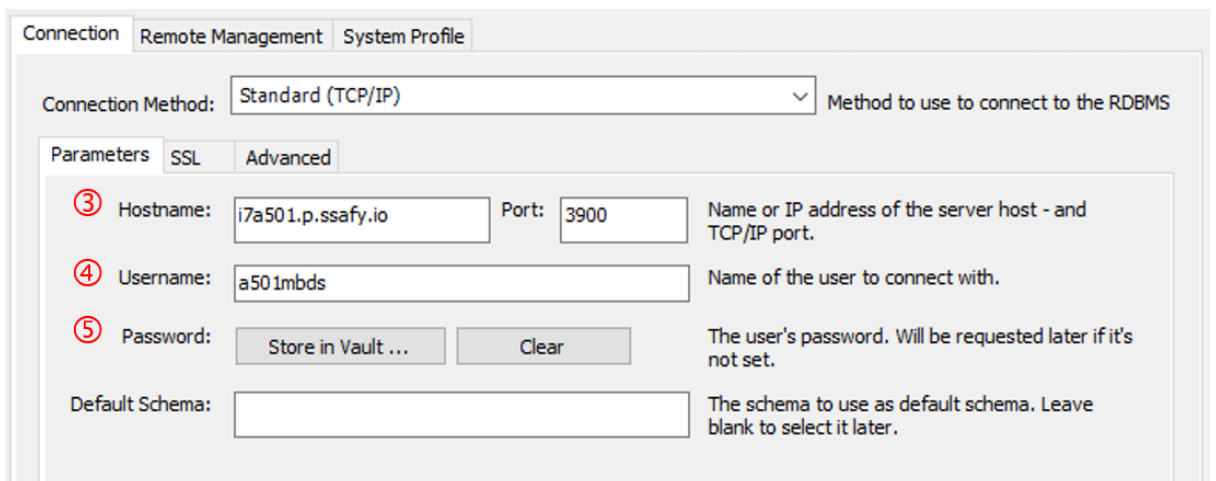


2. 원하는 Connection Name 입력



3. hostname에 i7a501.p.ssafy.io 입력, port에는 3900 입력

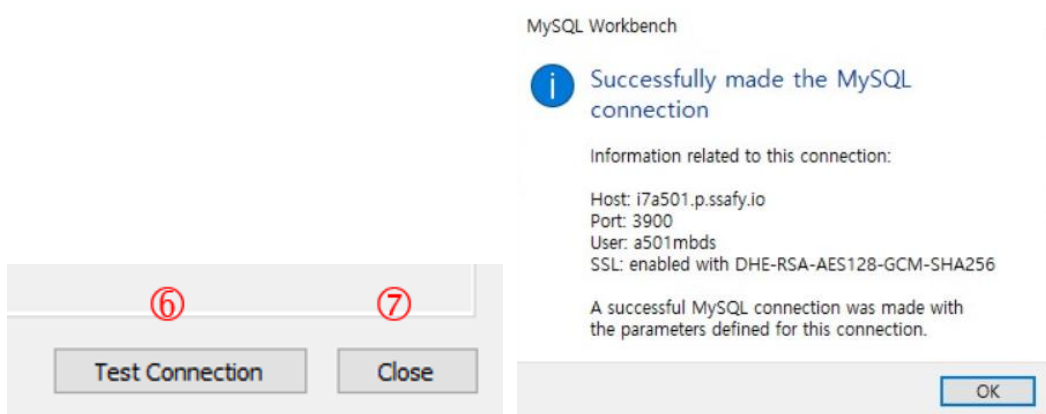
4. username에 a501mbds 입력



5. Password: [store in vault...]버튼 클릭 후, 비밀번호 fx5nqu8r4k 입력 후 OK 버튼 클릭



6. [Test Connection]클릭 후, ok 알림이 확인되면 Close 버튼 클릭하면 끝



5. 배포

■ backend 폴더에서 도커 빌드 실행

```
sudo docker build -t zz132456zz/a501_user .
```

■ backend-openvidu 폴더에서 도커 빌드 실행

```
sudo docker build -t zz132456zz/a501_room .
```

■ docker hub로 push

```
sudo docker push zz132456zz/a501_user  
sudo docker push zz132456zz/a501_room
```

■ user server, room server docker로 실행

```
sudo docker run -d -p 8010:8010 zz132456zz/a501_user  
sudo docker run -d -p 8011:8011 zz132456zz/a501_room
```

■ OpenVidu On premises 설치

```
sudo su  
cd /opt  
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh  
| bash  
exit
```

■ OpenVidu 세팅

- .env 설정 파일을 편집모드로 연다.

```
cd openvidu  
  
sudo vi .env
```

- 아래의 항목들만 다음과 같이 변경해준다.

```
DOMAIN_OR_PUBLIC_IP=i7a501.p.ssafy.io  
  
OPENVIDU_SECRET=<사용할 secret 값>  
  
CERTIFICATE_TYPE=letsencrypt  
  
LESENCRYPT_EMAIL=<인증서 받을 때 입력한 이메일>  
  
HTTP_PORT=4442  
  
HTTPS_PORT=<사용할 포트 번호>
```

■ OpenVidu On Promises 실행

```
./openvidu start
```

- * 서버 local에서 nginx가 실행되고 있다면 nginx를 끄고 openvidu를 실행하고 다시 nginx를 실행시켜준다.

■ nginx 세팅

- default 설정 파일을 편집모드로 연다.

```
sudo vi /etc/nginx/sites-available/default
```

- 다음과 같이 변경해준다.

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

    server_name i7a501.p.ssafy.io;
    return 301 https://i7a501.p.ssafy.io$request_uri;
}

server {
    listen 443 ssl;
    listen [::]:443;
    server_name i7a501.p.ssafy.io;

    ssl_certificate /etc/letsencrypt/live/i7a501.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/i7a501.p.ssafy.io/privkey.pem;

    location / {
        root /home/ubuntu/front/S07P12A501/frontend/dist;
        index index.html index.htm;
        try_files $uri $uri/ /index.html;
    }

    location /api/v1/users {
        proxy_pass https://i7a501.p.ssafy.io:8010;
        proxy_redirect off;
        charset utf-8;

        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-NginX-Proxy true;
    }

    location /api/v1/rooms {
        proxy_pass https://localhost:8011;
        proxy_redirect off;
        charset utf-8;

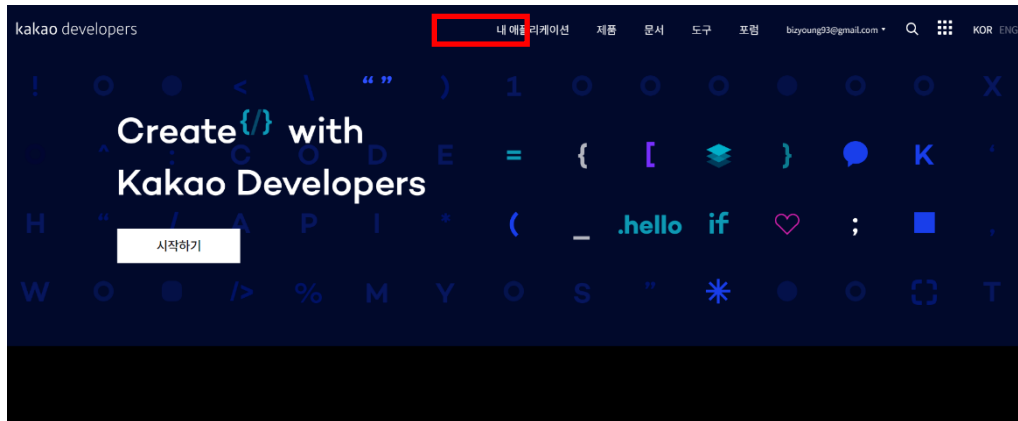
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For @proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-NginX-Proxy true;
    }

    location /api/v1/game {
        proxy_pass https://localhost:8011;
        proxy_redirect off;
        charset utf-8;

        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-NginX-Proxy true;
    }
}
```

6.외부 서비스

■ 카카오 개발자 사이트 로그인 후 내 애플리케이션 클릭



■ 애플리케이션 추가하기 클릭



■ 아이콘과 앱 이름(We See Play), 사업자명(WSP) 입력

애플리케이션 추가하기

앱 아이콘 파일 선택

JPG, GIF, PNG
원장 사이즈 128px, 최대 250KB

앱 이름

사업자명

• 입력된 정보는 사용자가 카카오 로그인을 할 때 표시됩니다.
• 정보가 정확하지 않은 경우 서비스 이용이 제한될 수 있습니다.


■ 이후 생성된 애플리케이션 클릭

전체 애플리케이션 (1)

애플리케이션 이름

+

애플리케이션 추가하기

 WeSeePlay

ID 778746 OWNER Web

■ 사용할 API KEY 메모

앱 키

네이티브 앱 키	4b154a2029e0a9e80e7374f21b29e712
REST API 키	df96d4ac3c5f7e31da654cf8c1308d29
JavaScript 키	9f41e650ba635f9ead57e2ffea63af7
Admin 키	1142987caabd73615d35b8ebe6d5e858

■ 웹 플랫폼 추가

플랫폼

설정된 플랫폼 정보가 없습니다 [플랫폼 설정하기](#)

Web

Web 플랫폼 등록

■ 사이트 도메인 입력

Web 플랫폼 등록

사이트 도메인

JavaScript SDK, 카카오톡 공유, 카카오톡, 메시지 API 사용시 등록이 필요합니다.
여러개의 도메인은 줄바꿈으로 추가해주세요. 최대 10까지 등록 가능합니다. 추가 등록은 포럼(데브톡)으로 문의주세요.
예시: (O) <https://example.com> (X) <https://www.example.com>

<https://i7a501.p.ssafy.io>
<http://localhost:8080>
<http://localhost:8081>

기본 도메인

기본 도메인은 첫 번째 사이트 도메인으로, 카카오톡 공유와 카카오톡 메시지 API를 통해 발송되는 메시지의 Web 링크 기본값으로 사용됩니다.

<https://i7a501.p.ssafy.io>

취소

저장

■ 카카오 로그인 메뉴 클릭 후 활성화 상태 ON 변경

내 애플리케이션 > 제품 설정 > 카카오 로그인

앱 설정

요약 정보

일반

비즈니스

앱 키

플랫폼

팀 관리

제품 설정

카카오 로그인

동의항목

간편가입

카카오톡 채널

개인정보 국외이전

연결 끊기



WeSeePlay

ID 778746 | OWNER | Web

카카오 로그인 ON

동의 화면 미리보기

활성화 설정 2

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.

OpenID Connect 활성화 설정

■ Redirect URI 등록

Redirect URI

Redirect URI 등록

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

Redirect URI

Redirect URI

카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다.

여러개의 URI를 줄바꿈으로 추가해주세요. (최대 10개)

REST API로 개발하는 경우 필수로 설정해야 합니다.

예시: (O) <https://example.com/oauth> (X) <https://www.example.com/oauth>

<https://i7a501.p.ssafy.io/login/kakao>
<http://localhost:8080/login/kakao>
<http://localhost:8081/login/kakao>

1

취소

저장

2

■ 동의 항목 설정

개인정보

항목 이름	ID	상태
닉네임	profile_nickname	<div><div><div>● 필수 동의</div><div>설정</div></div></div>
프로필 사진	profile_image	<div><div><div>● 사용 안함</div><div>설정</div></div></div>
카카오계정(이메일)	account_email	<div><div><div>● 선택 동의 [수집]</div><div>설정</div></div></div>

7.기타 참고사항

■ Frontend .env 파일이 .gitignore 처리되어 있다.

□ Frontend build 전 exec 폴더의 .env 파일을 frontend 폴더에 넣어준다.

■ Backend keystore.p12 파일이 .gitignore 처리되어 있다.

□ Backend build 전 exec 폴더의 keystore.p12 파일을 backend, backend-openvidu 폴더 내 src/main/resources 폴더에 넣어준다.