



E101 : ZOOMGLE

삼성청년SW아카데미 부울경캠퍼스 7기

공통프로젝트 [6주 2022.07.11 ~ 2022.08.19]

포팅 매뉴얼

담당 컨설턴트 : 김민정

민경대[팀장], 황인빈, 조찬홍, 김혜라, 김동욱, 서광석

=== 목차 ===

1. 기술스택	*****	3
2. 빌드 상세내용	*****	4
3. 배포 특이사항	*****	6
4. DB 계정	*****	7
5. 프로퍼티 정의	*****	8
6. 외부 서비스	*****	11

각자의 생활 속에서 시간이 없어 친구들과의 만남이 어려워지는 경험은 모두가 자연스레 겪게 되는 문제일 것입니다. 최근에는 다양한 플랫폼을 통해 서로 게임, 통화, 모임 등을 즐기기도 하지만 게임 화면과 목소리로만 교류하는 것에 피로감을 느끼기도 할 것입니다.

이런 고민과 피로감의 해결을 위해 화상 통화로 서로 얼굴을 보고, 게임을 즐길 수 있는 서비스를 제공합니다. 친구를 추가하고 게임 일정을 잡아 친구를 초대할 수 있습니다. 대기실에서 모두가 다 모이게 되면 게임을 시작할 수 있습니다. 게임을 시작하면 주사위를 굴리고, 미션을 수행하며 골인 지점에 누구보다 빨리 도착하여 탈출하시기 바랍니다.

게임을 즐긴 뒤에는 일지에서 게임의 기록과 게임 중 찍은 사진 또한 확인할 수 있습니다.

1. 프로젝트 기술 스택

가. 이슈관리 : Jira

나. 형상관리 : Gitlab

다. 커뮤니케이션 : Mattermost, Notion

라. 개발 환경

A. OS : Windows 10, Mac

B. IDE

i. IntelliJ 2022.1.3 (Ultimate Edition)

ii. Visual Studio Code 1.70.0v

C. DataBase : MySQL Workbench 8.0CE

D. Server : AWS EC2 (MobaXterm)

i. Ubuntu 20.04 LTS

ii. Docker 20.10.17v (server/client 동일)

E. 상세 사용

i. BackEnd

1) Java (OpenJDK Zulu 8.33.0.1-win64)

2) Spring Boot Gradle 6.7

ii. FrontEnd

1) HTML5, CSS, JavaScript(ES6)

2) React.js 17.0.2

3) react-redux 8.0.2, react-router-dom 6.3.0, react-saga 1.1.3

4) openvidu-browser 2.22.0, html2canvas 1.4.1, axios 0.27.2

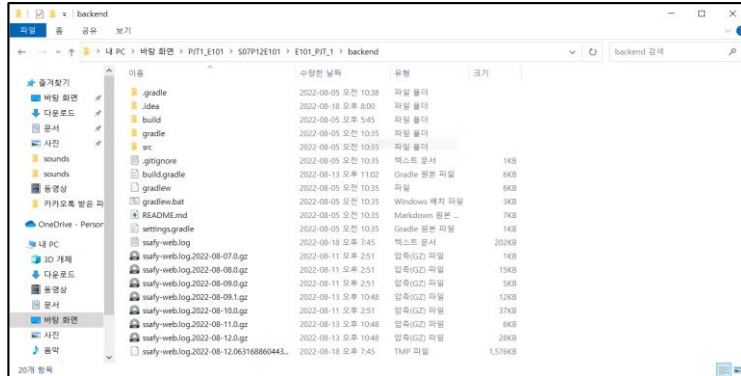
iii. AWS S3

2. 빌드 상세내용

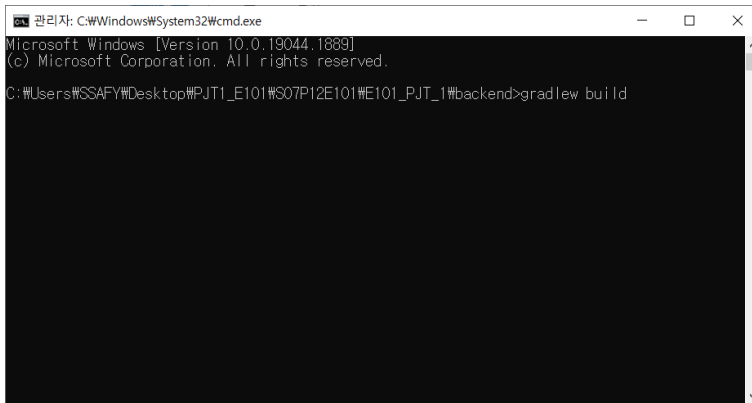
2-1. 백엔드 빌드 방법

가. Git clone <https://lab.ssafy.com/s07-webmobile1-sub2/S07P12E101.git>

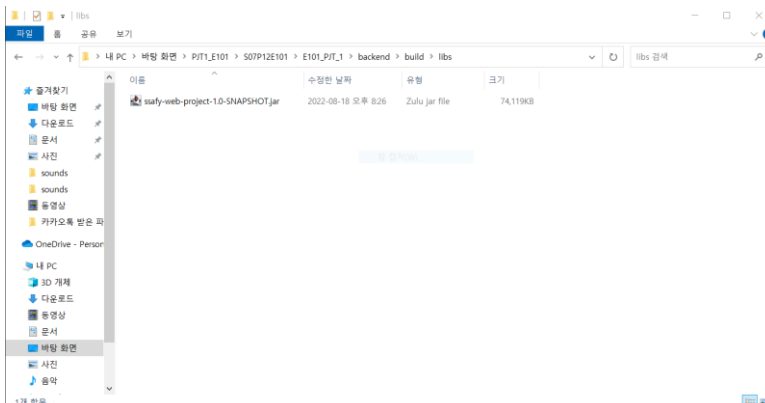
나. 클론 한 폴더에서 S07P12E101 > E101_PJT_1 > backend 폴더에 들어갑니다.



다. 주소창에서 cmd를 쳐서 명령 프롬프트창을 열어 [gradlew build] 명령어로 빌드합니다.



라. build > libs 폴더에서 생성된 jar 파일을 확인 할 수 있습니다.



2-2. 프론트엔드 빌드 방법

가. frontend 폴더에서 `npm install -g yarn` 명령어를 사용하여 yarn을 설치합니다.

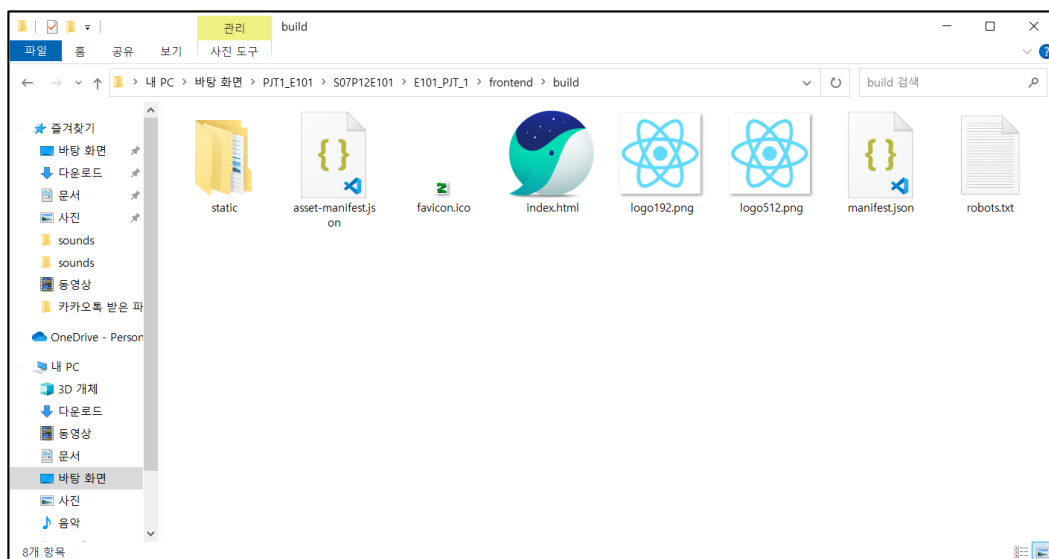
나. `yarn install`을 사용하여 프로젝트에 필요한 파일을 설치합니다.

```
C:\Users\SSAFY\Desktop\PJT1_E101\S07P12E101\E101_PJT_1\frontend>yarn install  
yarn install v1.22.19
```

다. `yarn run build` 명령어를 사용하여 프로젝트를 빌드합니다.

```
C:\Users\SSAFY\Desktop\PJT1_E101\S07P12E101\E101_PJT_1\frontend>yarn run build  
yarn run v1.22.19  
$ react-scripts build
```

라. build 폴더에서 build된 파일들을 확인합니다.



3. 배포 특이사항

가. AWS EC2에 만들어진 백엔드 빌드 파일(jar파일), 프론트 엔드 빌드 파일(build폴더)를 업로드 후 수동 배포를 진행합니다.

나. 다음 명령어를 입력하여 백엔드 빌드를 진행합니다.

A. 현재 실행되고 있는 jar파일 확인

```
$ sudo ps -ef | grep ssafy-web-project-1.0-SNAPSHOT.jar
```

B. 찾은 PID를 이용하여 실행되고 있는 프로세스 종료

```
$ kill -9 <PID>
```

C. 새로운 jar파일 배포

```
$ java -jar ssafy-web-project-1.0-SNAPSHOT.jar  
$ nohup java -jar ssafy-web-project-1.0-SNAPSHOT.jar & (백그라운드로 실행)
```

다. 다음 명령어를 입력하여 프론트엔드 빌드를 진행합니다.

A. Openvidu 내부의 nginx와의 충돌을 방지하기 위해 openvidu서버를 종료

```
$ cd /opt/openvidu (오픈비두 폴더로 이동)  
$ sudo ./openvidue stop
```

B. 프론트엔드 빌드를 진행하기 위해 nginx 재실행

```
$ sudo service nginx restart
```

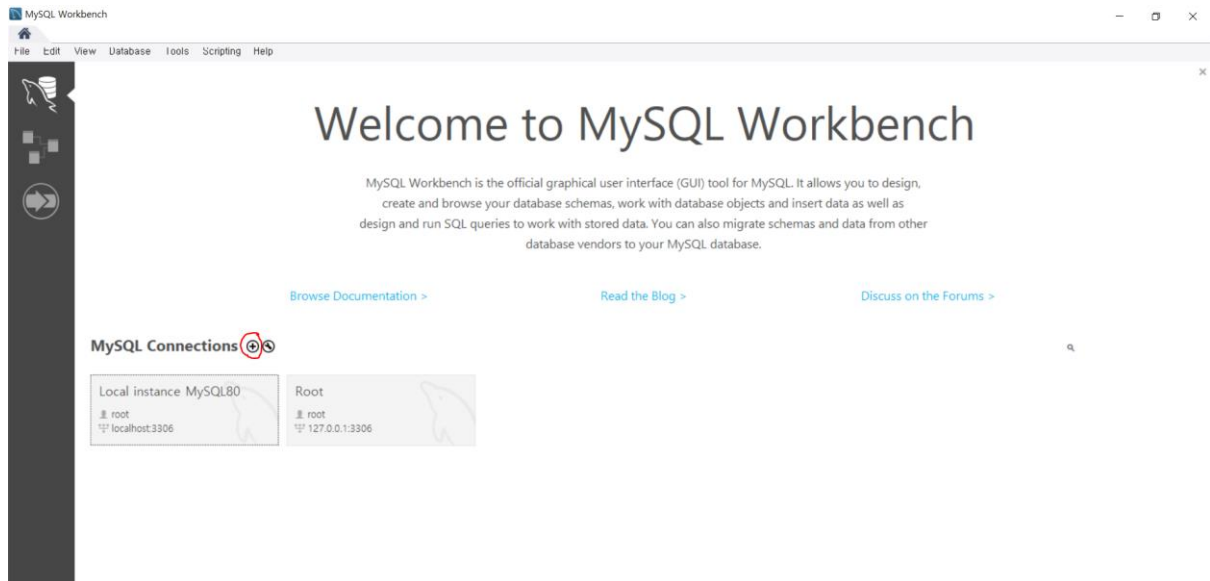
C. Nginx 재실행 후 openvidu 서버 실행

```
$ sudo ./openvidu start
```

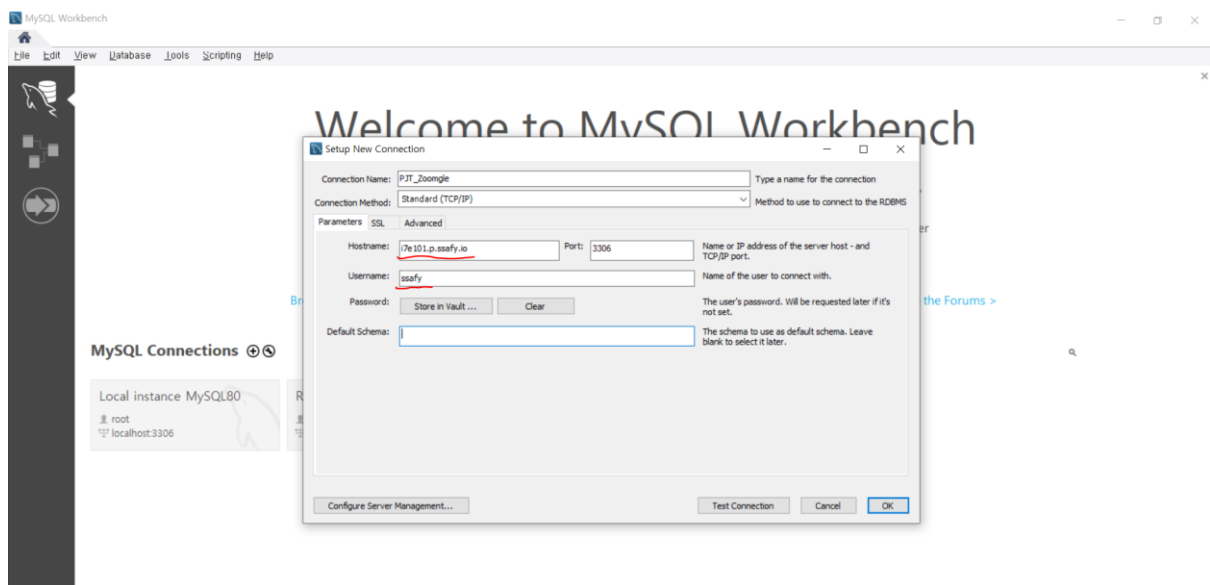
라. 배포가 완료되었는지 확인 후 마무리

4. DB 계정

가. MySQL workbench 계정 등록

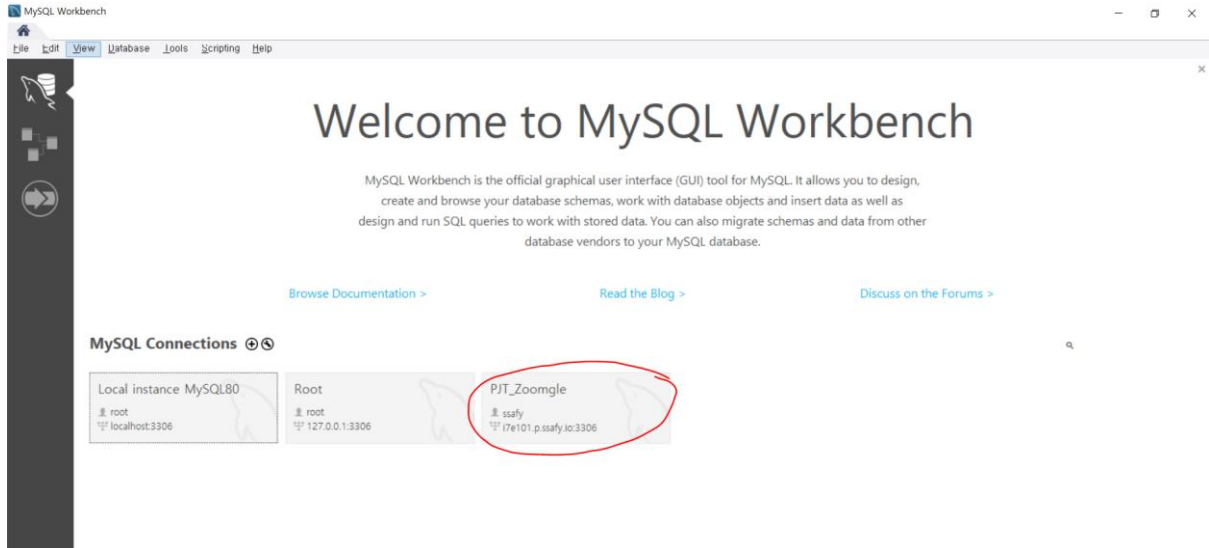


>> MySQL workbench 에 새로운 계정 등록을 위해 + 버튼 클릭



>> username : ssafy password : ssafy123!

>> EC2 계정 정보를 사용하여 연동된 DB를 불러옴



>> 저장된 connection 클릭하여 DB 접속

5. 프로퍼티 정의

가. Certbot SSL 인증서 발급

A. Certbot 설치

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

B. SSL 인증서 발급

```
$ sudo certbot --nginx
```

나. Nginx 세팅

A. 서버의 패키지 목록 업데이트

```
$ sudo apt update
```

B. Nginx 설치

```
$ sudo apt install nginx
```


C. Nginx 설정 파일 수정

```
# add server setting
server {
    listen 443 http2;
    server_name i7e101.p.ssafy.io;

    ssl on;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_certificate /etc/letsencrypt/live/i7e101.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/i7e101.p.ssafy.io/privkey.pem; # managed by Certbot

    location / {
        root /home/ubuntu/build;
    }

    location /api {
        proxy_pass http://localhost:8080;
        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;
    }
}
```

```
server {
    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;
    server_name i7e101.p.ssafy.io; # managed by Certbot

    location / {
        root /home/ubuntu/build;
    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/i7e101.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/i7e101.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 ($host = i7e101.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80 ;
    listen [::]:80 ;
    server_name i7e101.p.ssafy.io;
    return 404; # managed by Certbot
}
```

다. AWS EC2 세팅

A. MySQL 설치

i. 세팅을 위해 최신 상태 업데이트

```
$ sudo apt-get install
```

ii. MySQL 설치

```
$ sudo apt-get install mysql-server (진행 중에 Y)
```

iii. MySQL 접속

```
$ sudo mysql
```

iv. MySQL에서 계정 생성 후 권한 설정

```
> use mysql  
> CREATE USER '<ID>'@'%' IDENTIFIED BY '<PASSWORD>';  
> GRANT ALL PRIVILEGES ON *.* TO '<ID>'@'%' WITH GRANT OPTION;  
> FLUSH PRIVILEGES;  
> EXIT
```

v. 외부 접속 허용

```
$ cd /etc/mysql/mysql.conf.d (폴더 이동)  
$ sudo vi mysqld.cnf (파일 수정)
```

vi. 수정할 내용

```
bind-address = 0.0.0.0
```

vii. 포트 열고 재시작

```
$ sudo ufw allow 3306 (포트 열기)  
$ sudo systemctl restart mysql.service (재시작)
```

B. Docker 설치

i. 세팅을 위해 최신 상태 업데이트

```
$ sudo apt-get update
```

ii. Docker 설치

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io  
$ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)"  
-o /usr/local/bin/docker-compose  
$ sudo chmod +x /usr/local/bin/docker-compose
```

6. 외부 서비스

가. Openvidu Server 구축

A. openvidu 사용을 위한 포트 확보

```
ufw allow ssh  
ufw allow 80/tcp  
ufw allow 443/tcp  
ufw allow 3478/tcp  
ufw allow 3478/udp  
ufw allow 40000:57000/tcp  
ufw allow 40000:57000/udp  
ufw allow 57001:65535/tcp  
ufw allow 57001:65535/udp  
ufw enable
```

B. openvidu 설치

```
$ cd /op      # /opt dir에 설치하는 것이 권장사항  
  
$ sudo curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh |  
sudo bash
```

C. 설정파일 수정

```
$ sudo su          # 권한을 변경해야 하위 접근이 가능함

$ cd openvidu
$ vi .env
```

i. 다음 부분만 수정하면 됨

```
DOMAIN_OR_PUBLIC_IP=i7e101.p.ssafy.io
# [Linux 서버의 public ip 주소 또는 서비스 도메인]

OPENVIDU_SECRET=e101ssafy71      # 사용자 임의 할당

CERTIFICATE_TYPE=letsencrypt
# default는 selfsigned 로 '보안 문제'를 야기함
# SSL 키가 있다면 owncert 로 하며, /owncert dir 내에 SSL키가 있어야 함
# letsencrypt를 사용할 때는, LETSENCRYPT_EMAIL을 유효한 이메일로 사용해야한다.

LETSENCRYPT_EMAIL=[유효한 이메일 주소]    # 사용자 임의 할당

# PORT 값은 사용하려는 포트 값으로 입력
HTTP_PORT=800
HTTPS_PORT=4443
```

D. Openvidu 서버 실행

```
$ sudo ./openvidu start
```

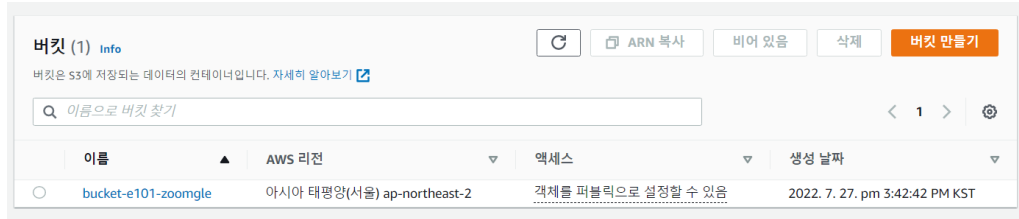
E. Docker Container 확인

```
$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	NAMES	CREATED	STATUS	PORTS
ee728853fb87	openvidu/openvidu-call:2.22.0	"docker-entrypoint.s..."	openvidu_app_1	3 hours ago	Up 3 hours	
dc3abfc0ea66	kurento/kurento-media-server:6.16.0	"/entrypoint.sh"	openvidu_kms_1	3 hours ago	Up 3 hours (healthy)	
4afdb885a72f	openvidu/openvidu-coturn:2.22.0	"docker-entrypoint.s..."	openvidu_coturn_1	3 hours ago	Up 3 hours	0.0.0.0:3478→3478/tcp, 0.0.0.0:3478→3478/udp, :::3478→3478/tcp, :::3478→3478/udp, 5349/tcp
b6b04f8dcfb	openvidu/openvidu-server:2.22.0	"/usr/local/bin/entr..."	openvidu_server_1	3 hours ago	Up 3 hours	
e100c92ea978	openvidu/openvidu-proxy:2.22.0	"docker-entrypoint.s..."	openvidu_nginx_1	3 hours ago	Up 3 hours	

나. AWS S3

A. 버킷 생성



B. 버킷 액세스 설정



C. 버킷 액세스 키

access key : AKIAVALGCDCSAX5KW5U6

secret key : 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