

삼성 청년 SW 아카데미 서울캠퍼스 7기

공통 프로젝트 (2022/07/12 – 2022/08/19)

포팅 매뉴얼

노희진(팀장), 김연준, 박남주, 이지현, 김다은

[ 목차 ]

1. 기술 스택

2. 빌드 상세 내용

3. DB 설정

4. 프로퍼티 정의

5. 외부 서비스

================================================================스팀모아는 멀티 게임 플레이어가 게임을 고르고 파티원을 모집할 수 있도록 돕는 서비스입니다.

**1. 프로젝트 기술 스택**

가. 이슈 관리 : JIRA  
나. 형상 관리 : GitLab

다. 커뮤니케이션 : Mattermost, Notion

라. 개발 환경

1) OS : Windows 10, Mac OS

2) IDE

2-1) Back-End : IntelliJ Ultimate

2-2) Front-End : Visual Studio Code

2-3) UI/UX : Figma

3) DB : MySQL Workbmench 8.0 CE

4) Server : AWS EC2

4-1) GUI : MobaXterm

마. 상세 내용

1) Back-End

1-1) JAVA : OpenJDK version "1.8.0\_192"

1-2) Spring Boot Gradle : 2.7.1

1-3) Swagger2 : 2.9.2, JWT : 3.10.3, jjwt : 0.9.1, junit : 5.6.0, querydsl-jpa

2) Front-End

2-1) HTML5, CSS3, JavaScript(ES6)

2-2) React(버전 필요)

2-3) 사용한 모듈, 라이브러리들 버전 포함

3) Server

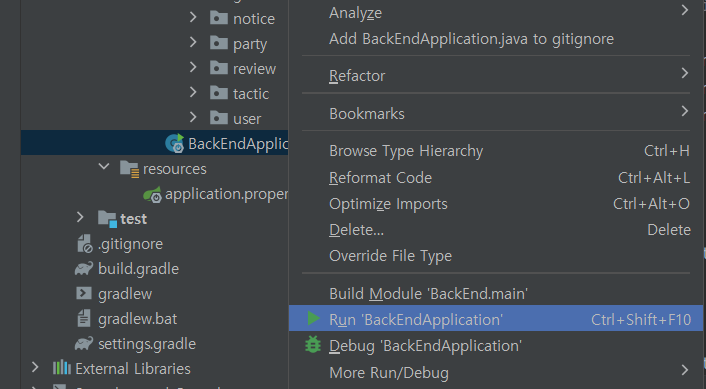
AWS EC2

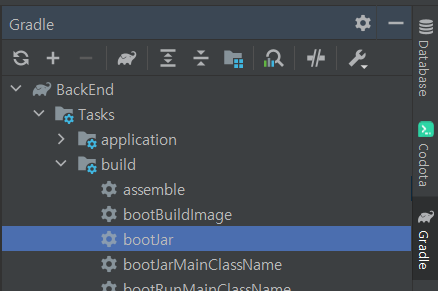
**2. 빌드 상세 과정**

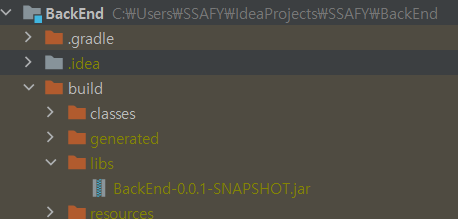
**2-1. 백엔드 빌드 상세 과정**

Git Repository에서 BackEnd 폴더를 clone 해옵니다.

git clone [git@lab.ssafy.com:s07-webmobile1-sub2/S07P12A303.git](mailto:git@lab.ssafy.com:s07-webmobile1-sub2/S07P12A303.git)

이후 IntelliJ 에서 (로컬 호스트로 구동 시)

(AWS EC2에 올리려면 Gradle > Tasks > build > bootJar 과정을 통해 



프로젝트 폴더 > build > libs 에 생성된 BackEnd-0.0.1-SNAPSHOT.jar 파일이

최종 백엔드 빌드 파일입니다.

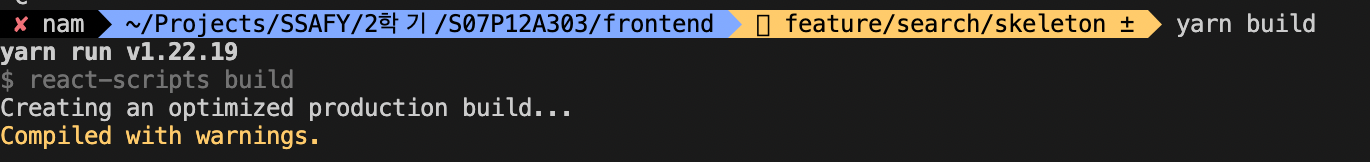
**2-2. 프론트엔드 빌드 상세 과정**

Git Repository에서 BackEnd 폴더를 clone 해옵니다.

git clone [git@lab.ssafy.com:s07-webmobile1-sub2/S07P12A303.git](mailto:git@lab.ssafy.com:s07-webmobile1-sub2/S07P12A303.git)

frontend 폴더 내부 터미널에서 명령어를 입력합니다

* yarn build



build폴더가 zip파일로 나오는데 이것이 최종 프론트엔드 빌드 파일입니다. 텍스트이(가) 표시된 사진

자동 생성된 설명

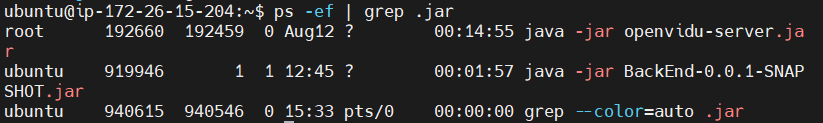
**2-3. 서버 배포 상세 과정**

위의 과정에서 나온 .jar 파일과 build 폴더를 AWS EC2 서버에 업로드 한 뒤 jar 파일을 실행시키면 된다.

\* 리눅스 명령어

1. 서버에서 실행되고 있는 .jar 파일의 목록 반환

ps – ef | grep .jar



2. 실행 중인 파일이 있다면 kill 하기

sudo kill 해당파일넘버

ex. sudo kill 919946

3. jar 파일 실행

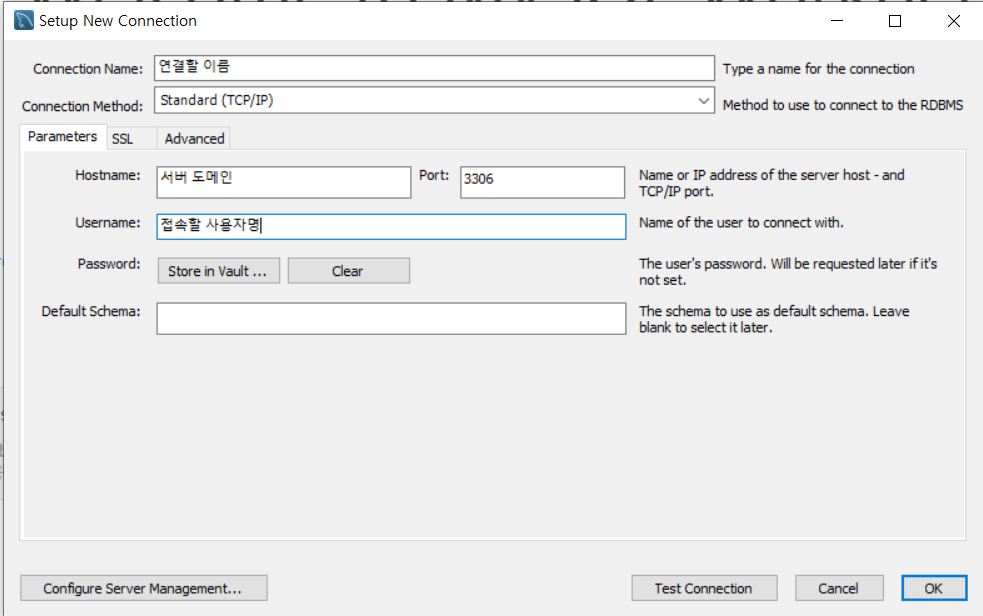
3-1. 해당 터미널이 돌아갈 때만 실행

java –jar 빌드한JAR파일명 &

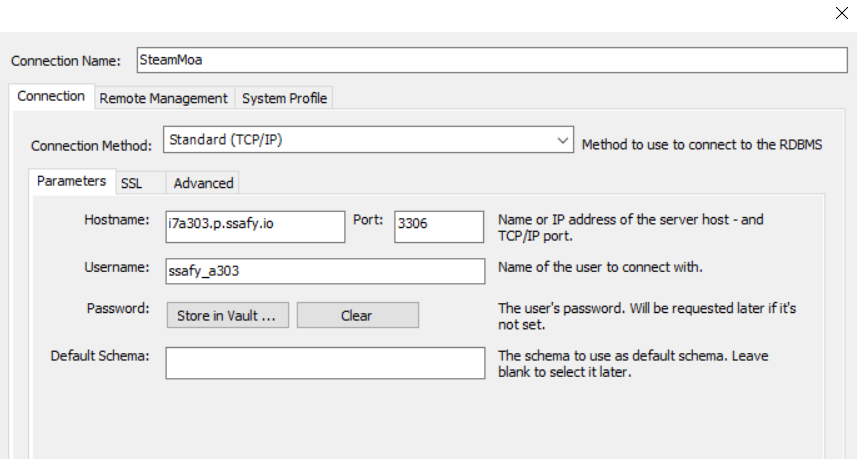
3-2. 백그라운드에서 계속 실행

nohup java –jar 빌드한JAR파일명 &

**3. DB 접속**

****

**(예시)**

****

**4. 프로퍼티 정의**

가. nginx 세팅

1. EC2 에서 세팅 파일로 접근

#nginx config 파일이 있는 위치로 이동

cd /etc/nginx/sites-available

#default 파일(Nginx 설정 파일) 관리자 권한으로 열기

sudo vi default

2. Nginx config 파일 수정

텍스트이(가) 표시된 사진

자동 생성된 설명

# 실제 코드

server {

root /home/ubuntu/build;

index index.html index.htm;

server\_name i7a303.p.ssafy.io;

location / {

try\_files $uri $uri/ /index.html;

}

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;

}

listen [::]:443 ssl ipv6only=on;

listen 443 ssl;

ssl\_certificate /etc/letsencrypt/live/i7a303.p.ssafy.io/fullchain.pem; #pem 키 위치

ssl\_certificate\_key /etc/letsencrypt/live/i7a303.p.ssafy.io/privkey.pem;

include /etc/letsencrypt/options-ssl-nginx.conf;

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem;

}

server {

if ($host = i7a303.p.ssafy.io) {

return 301 https://$host$request\_uri;

}

listen 80 ;

listen [::]:80 ;

server\_name i7a303.p.ssafy.io;

return 404;

}

**5. 외부 서비스**

가) OpenID

<https://partner.steamgames.com/doc/features/auth>

* OAuth 2.0 프로토콜 상위의 간단한 Identity Layer
* 하나의 Open ID를 통해 여러 서비스에 인증을 받을 수 있다.
* Open ID 제공자를 통해 사용자는 Open ID를 발급받을 수 있다.
* 사용자는 Open ID 지원 사이트에서 자신이 발급 받은 Open ID를 이용하여 인증 할 수 있다.
* Open ID를 지원하는 사이트의 사용자 인증은 Open ID제공자가 수행 한다.
* Open ID 제공자는 Open Id를 지원하는 사이트에게 인증된 사용자의 정보를 제공한다.

회원가입시 Steam을 이용하는 사용자들을 대상으로 서비스를 제공하기 위해 steam 계정인증을 받습니다.



텍스트, 스크린샷, 모니터, 검은색이(가) 표시된 사진

자동 생성된 설명

(프론트 구현 코드)