

# Porting Manual

## Index

### 1. **Stacks**

- Issue Management
- SCM (Software Configuration Management)
- Communities
- Development Environment
- Details

### 2. **Builds**

- BackEnd
- FrontEnd

### 3. **AI Learning**

### 4. **Deployment Command**

- Front & Back End Server
- Nginx Web Server

### 5. **MySQL WorkBench Connection**

### 6. **Nginx default**

### 7. **EC2 Setting**

- MySQL
- Nginx

### 8. **Files ignored**

### 9. **etc) Settings or Tips**

- run in Android emulator

# 1. Stacks

## 1.1 Issue Management



## 1.2. SCM



## 1.3. Communities



## 1.4. Development Environment

### 1. OS



- Android 10(Oreo) API : 29

### 2. IDE



- IntelliJ IDEA 2022.1.4 (Ultimate Edition)



- Visual Studio Code 1.70.1



- Android Studio 2021.2

### 3. Database



- MySQL WorkBench 8.0

### 4. Server



Ubuntu 20.04 LTS

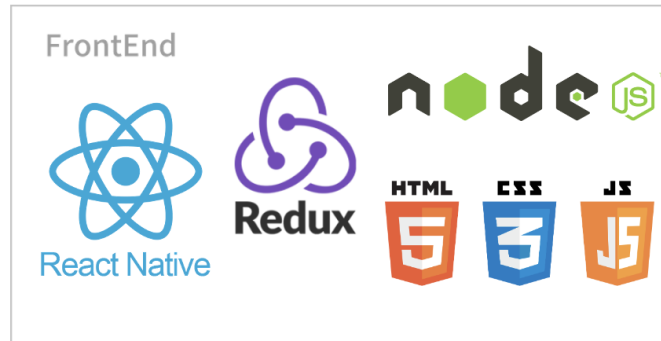
## 1.5. Details

### 1.5.1. Back-End



1. Java (Openjdk 11.0.16.1)
2. Spring Boot Gradle 7.5
3. Lombok 1.18.24
4. Swagger 3.0.0
5. JPA

### 1.5.2. Front-End



1. HTML5, CSS3, JAVASCRIPT(ES6)
2. React Native 0.68.2
3. React-redux 8.0.2
4. Node JS 16.16.0

### 1.5.3. etc



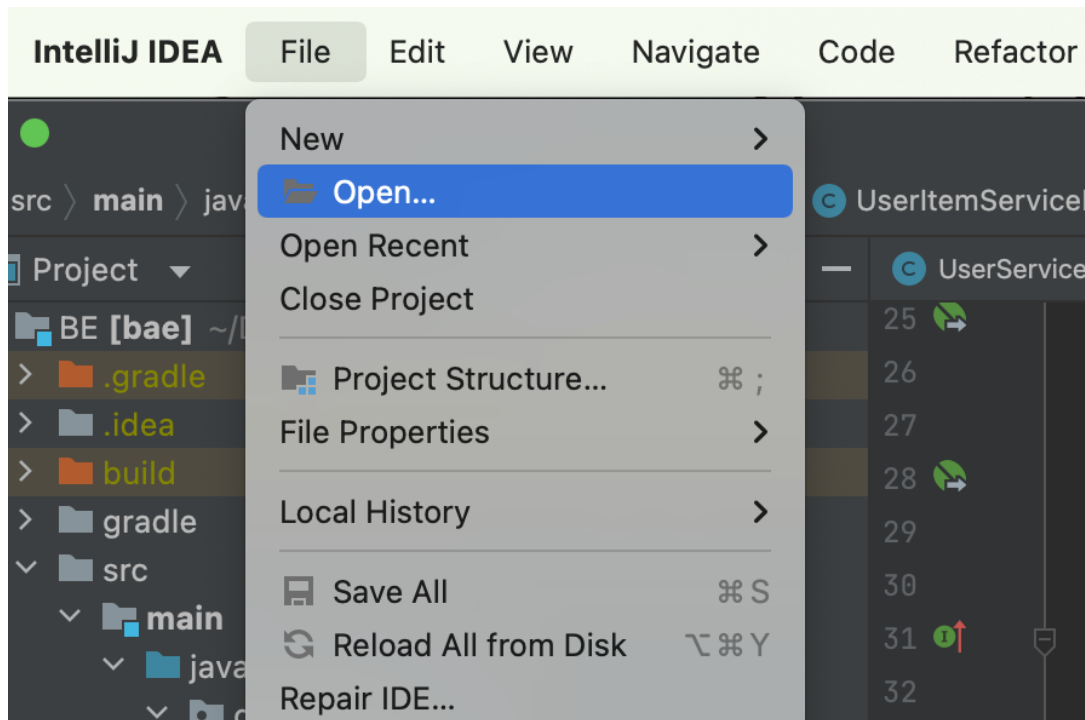
1. AWS EC2
2. Nginx (1.18.0)
3. Android Emulator

## 2. Builds

### 2.1. How to build BE (Spring boot)

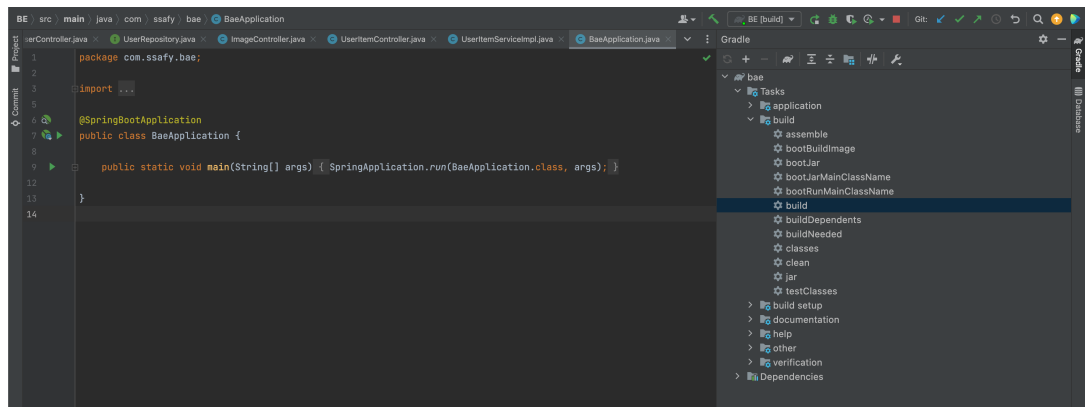
#### 2.1.1. GUI

1. Spring Boot Project Import
  - a. IntelliJ Ultimate 사용
    - i. IntelliJ 실행 후 File - Open 클릭하여 BE 폴더 open

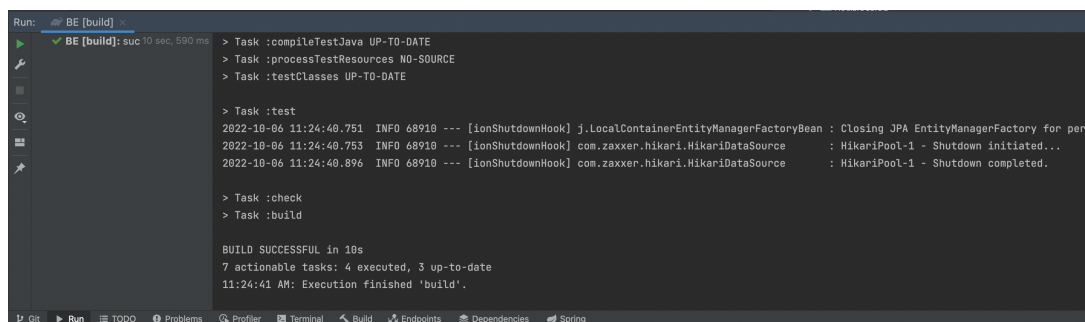


## b. Project Build

### i. 우측 Gradle 클릭 후 Tasks-build-build 클릭



### ii. 좌측 하단에 실행 내역(Run)에서 Successful 확인



iii. 프로젝트 폴더 - build - lib - [Project Name]-0.0.1-SNAPSHOT.jar 파일 확인

## 2.1.2. Command

1. 빌드 희망하는 프로젝트 폴더에서 **gradlew** 파일이 존재하는 위치로 이동

```
2022-08-17 오후 02:13 <DIR> .
2022-08-17 오후 02:13 <DIR> ..
2022-08-11 오후 10:38 507 .gitignore
2022-08-11 오후 10:41 <DIR> .gradle
2022-08-17 오후 07:50 .idea
2022-08-17 오후 07:47 <DIR> build
2022-08-17 오후 02:13 2,905 build.gradle
2022-08-11 오후 10:38 <DIR> gradle
2022-08-11 오후 10:38 8,428 gradlew
2022-08-11 오후 10:38 2,838 gradlew.bat
2022-08-17 오전 11:33 <DIR> octopus
2022-08-11 오후 10:38 0 README.md
2022-08-11 오후 10:38 30 settings.gradle
2022-08-11 오후 10:38 <DIR> src
6개 파일 14,708 바이트
8개 디렉터리 3,785,463,578,624 바이트 남음
```

2. 아래의 명령어를 통해서 빌드

- a. Windows

```
$ gradlew build
```

- b. Linux

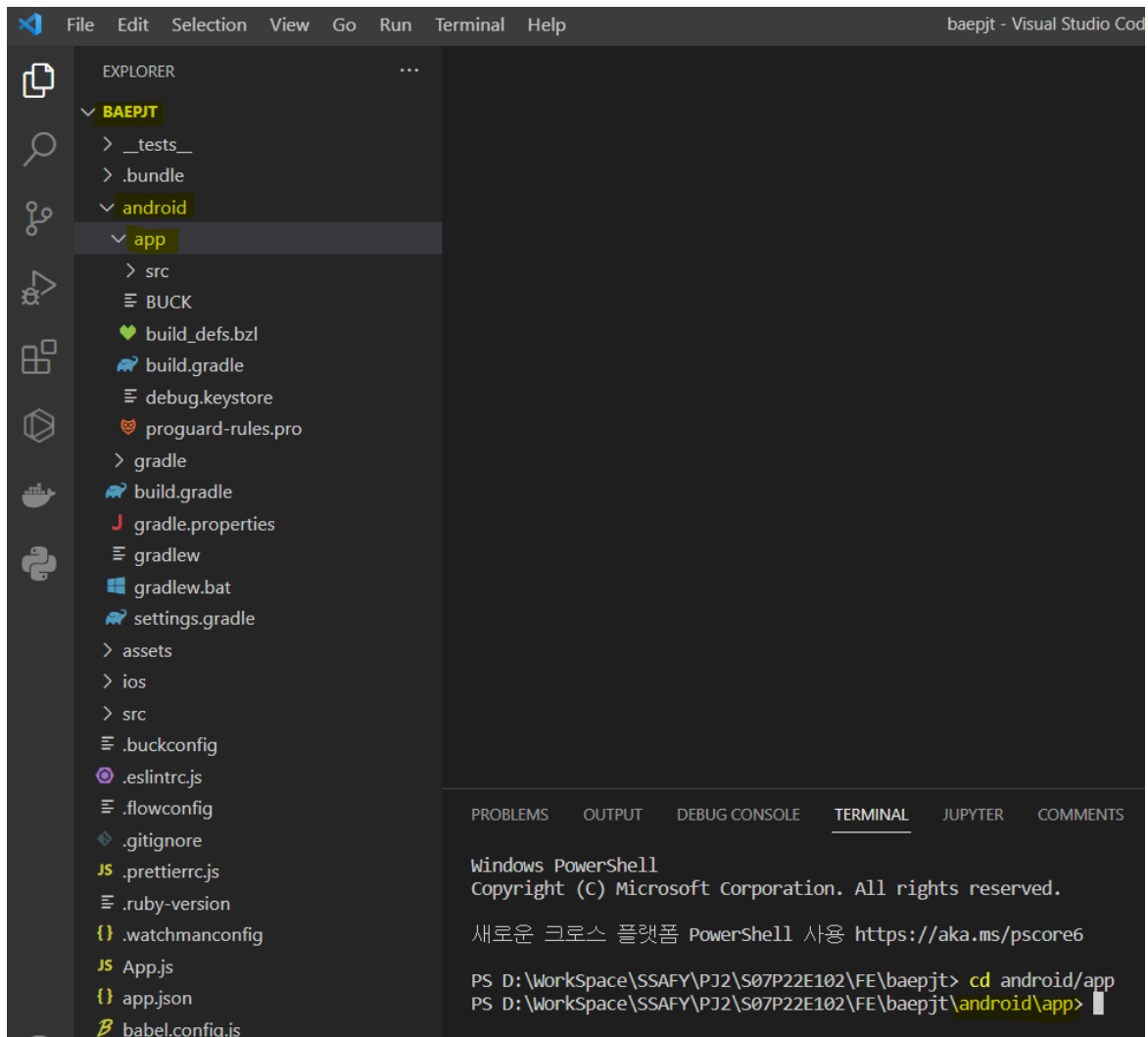
```
$ ./gradlew build
```

## 2.2. How to build FE (React Native)

### 2.2.1. 서명 키 생성

- 서명 키 : 해당 앱이 복제되거나 위변조 되지 않았는지 확인 등을 위해 Android는 인증서를 사용해 디지털 방식으로 서명된 APK만 기기에 설치하거나 업데이트할 수 있다.

1. 아래 위치로 이동



프로젝트 기준 FE/baepjt/android/app 입니다.

## 2. 다음 명령어 작성

```
keytool -genkeypair -v -keystore my-upload-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000
```

## 3. 질문에 대한 답변 작성 (아래는 질문 리스트)

- a. 이름, 성
- b. 조직(소그룹, 조직 이름)
- c. 국가 코드
- d. my-key-alias에 대한(서명 키) 키 비밀번호 입력  
※ 키 저장소와 동일한 비번 쓰고 싶다면 그냥 엔터키 입력

## 4. 키 생성 확인

- a. 생성된 키 파일 이름들
  - i. custom-key-name.keystore (키 저장소)
  - ii. my-key-alias.key (키)
  - iii. debug.keystore (디버그용 키 저장소)

## 2.2.2. Gradle 설정 변경

### 1. gradle.properties 설정

- a. 파일 위치 : /android/gradle.properties
  - i. 프로젝트 기준 : FE/baepjt/android/gradle.properties
- b. 추가할 내용

```
MYAPP_RELEASE_STORE_FILE=my-upload-key.keystore
MYAPP_RELEASE_KEY_ALIAS=my-key-alias
MYAPP_RELEASE_STORE_PASSWORD=위에서 지정한 비밀번호
MYAPP_RELEASE_KEY_PASSWORD=위에서 지정한 비밀번호
```

- c. 추가 위치
  - i. gradle.properties 아무대나
- d. 유의 사항
  - i. 경로 주의.

### 2. build.gradle 설정

- a. 파일 위치 : /android/app/build.gradle
  - i. 프로젝트 기준 : FE/baepjt/android/app/build.gradle
- b. 추가할 내용
  - i. signingConfig 설정 내용

```
signingConfigs {
    release {
        if (project.hasProperty('MYAPP_RELEASE_STORE_FILE')) {
            storeFile file(MYAPP_RELEASE_STORE_FILE)
            storePassword MYAPP_RELEASE_STORE_PASSWORD
            keyAlias MYAPP_RELEASE_KEY_ALIAS
            keyPassword MYAPP_RELEASE_KEY_PASSWORD
        }
    }
}
```

#### ii. signingConfig 빌드 타입 설정

```
signingConfig signingConfigs.release
```

- c. 추가할 위치
  - i. signingConfig 설정 내용
    1. 파일 내 android - defaultConfig 아래 작성(내부 작성X)
    2. 예시 코드

```
android {
    ...
    defaultConfig { ... }
    splits { ... }
    signingConfigs {
        release {
            if (project.hasProperty('MYAPP_RELEASE_STORE_FILE')) {
                storeFile file(MYAPP_RELEASE_STORE_FILE)
            }
        }
    }
}
```



```

        storePassword MYAPP_RELEASE_STORE_PASSWORD
        keyAlias MYAPP_RELEASE_KEY_ALIAS
        keyPassword MYAPP_RELEASE_KEY_PASSWORD
    }
}
buildTypes {
    release {
        ...
        signingConfig signingConfigs.release
    }
}
}

```

#### ii. signingConfig 빌드 타입 설정

1. 파일 내 android - buildTypes - release 내부에 작성

2. 예시 코드

a. 위 코드 참조

#### d. 유의 사항

i. build.gradle이란 파일은 여러 경로에 많이 존재 하므로 파일 혼동없게 조심.

ii. 코드 추가 할때 위치 잘 확인

### 2.2.3. 어플리케이션 빌드

1. /android 로 터미널 이동

2. 아래 명령어로 빌드 실행

```
./gradlew assembleRelease
```

3. 아래 경로에서 확인

```
/android/app/build/outputs/apk/app-release.apk
```

## 3. AI Learning

### 3.1. 준비사항

1. 원하는 위치에 폴더 생성

2. 아래 명령어로 소스 다운

```
git clone https://github.com/ultralytics/yolov5.git
```

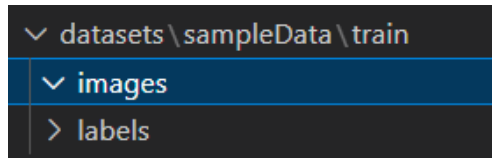
3. 원하는 데이터 구하기

예시 링크) <https://www.kaggle.com/datasets> 에서 yolo v5 검색 후 dataset 설정

### 3.2. 학습 전 설정

### 3.2.1. 데이터 설정

1. yolo 폴더에 datasets 폴더 생성
2. datasets 하위 폴더로 구분하기 쉬운 폴더와 그 하위로 images, labels 폴더 생성



SampleData - train 폴더 생성 후 필수 폴더 생성

3. 다운로드 받은 데이터를 각 폴더에 맞게 이동
  - a. 이미지 - images
  - b. 이미지 설명 txt 파일 - labels

### 3.2.2. 학습 데이터 관련 설정

1. yolo 폴더 안에 data 로 이동
2. sample.yaml 생성
3. 아래 내용 추가 및 다운로드 받은 데이터 에 있는 yaml 데이터 추가

```
path: ../datasets/sampleData# dataset root dir
train: train/images# train images (relative to 'path') 128 images
val: train/images# val images (relative to 'path') 128 images
test: # test images (optional)

nc: 7
names: ['fish', 'jellyfish', 'penguin', 'puffin', 'shark', 'starfish', 'stingray']
```

- path : 데이터가 존재하는 폴더의 루트 주소 입니다.
  - train : 이미지 데이터가 존재하는 폴더 주소 입니다.
  - nc : 구별할 사물의 갯수 입니다.
  - names : 구별할 사물의 이름들 리스트입니다.
    - nc, names는 보통 받은 데이터 안에 data.yaml 로 있습니다.
4. yolo 폴더에 train.py 파일을 아래와 같이 수정
    - a. 기존 데이터 경로 및 정보 파일(yaml)을 변경

```
parser.add_argument('--data', type=str, default=ROOT / 'data/coco128.yaml', help='dataset.yaml path')
```

```
parser.add_argument('--data', type=str, default=ROOT / 'data/sample.yaml', help='dataset.yaml path')
```

- b. worker 수를 8명에서 0명으로 변경(에러날 가능성이 높다고 함)

```
parser.add_argument('--workers', type=int, default=8, help='max dataloader workers (per RANK in DDP mode)')
```

```
parser.add_argument('--workers', type=int, default=0, help='max dataloader workers (per RANK in DDP mode)')
```

### 3.2.3. 필요한 라이브러리 및 드라이버

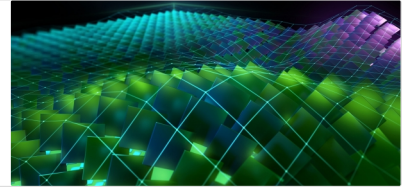
#### 1. 그래픽 드라이버(Nvidia 제품 한정)

본인 컴퓨터 적용 대상인지는 아래 링크에서 확인

##### CUDA GPUs - Compute Capability

Explore your GPU compute capability and learn more about CUDA-enabled desktops, notebooks, workstations, and supercomputers.

<https://developer.nvidia.com/cuda-gpus>



#### 2. 필요 라이브러리

##### a. PyTorch - GPU 사용 설정

- 아래 링크에서 확인

##### PyTorch

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported,

<https://pytorch.org/get-started/locally/>



- 링크 접속 후 스크롤 후 아래 사진 파트에서 선택 후 설치

PyTorch Build	Stable (1.12.1)		Preview (Nightly)		LTS (1.8.2)
Your OS	Linux		Mac		Windows
Package	Conda	Pip	LibTorch		Source
Language	Python			C++ / Java	
Compute Platform	CUDA 10.2	CUDA 11.1	CUDA 11.6	ROCm 5.1+	CPU
Run this Command:	<pre>pip3 install torch==1.8.2 torchvision==0.9.2 torchaudio==0.8.2 --extra-index-url https://download.pytorch.org/whl/lts/1.8/cu111</pre> <p><b>Note:</b> Additional support for these binaries may be provided by <b>PyTorch Enterprise Support Program Participants</b>.</p>				

- PyTorch Build : PyTorch 버전
- Your OS : 사용중인 운영체제
- Package : 사용중인 python 라이브러리 관리 체계
- Compute Platform : 1번(그래픽 드라이버 파트)에서 선택한 버전

##### b. Yolo V5 라이브러리 버전 요구사항

```
# Base -----
matplotlib>=3.2.2
numpy>=1.18.5
```

```

opencv-python>=4.1.1
Pillow>=7.1.2
PyYAML>=5.3.1
requests>=2.23.0
scipy>=1.4.1
torch>=1.7.0
torchvision>=0.8.1
tqdm>=4.64.0
# protobuf<=3.20.1 # https://github.com/ultralytics/yolov5/issues/8012

# Logging -----
tensorboard>=2.4.1
# wandb
# clearml

# Plotting -----
pandas>=1.1.4
seaborn>=0.11.0

# Extras -----
ipython # interactive notebook
psutil # system utilization
thop>=0.1.1 # FLOPs computation

```

### 3.3. 실행

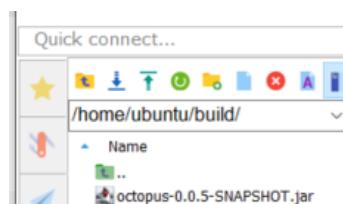
1. yolo 폴더로 이동(trade.py 있는 폴더)
2. 아래 명령어로 실행

```
python trade.py
```

## 4. Deployment Command

### 4.1. Front & Back End Server

1. 빌드 파일 위치한 폴더 이동 (/home/ubuntu/build)



2. 아래 명령어로 실행

```
java -jar [Server File Name].jar
```

### 4.2. Nginx Web Server

#### 1. 상태 확인

```
sudo systemctl status nginx
```

#### 2. 프로세스 시작

```
sudo systemctl start nginx
```

#### 3. 프로세스 종료

```
sudo systemctl stop nginx
```

#### 4. 프로세스 재시작

```
sudo systemctl restart nginx
```

## 5. How to use the MySQL workbench

### 5.1. Standard TCP/IP 연결

1. 홈 화면에서 MySQL Connections 에서 + 추가
2. 아래 사진처럼 설정.

Setup New Connection

Connection Name: SSAFY EC2 Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: j7e102.p.ssafy.io Port: 3306 Name or IP address of the server host - and TCP/IP port.

Username: ssafy Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

Store in Keychain ... Clear

Default Schema: The schema to use as default schema. Leave blank to select it later.

Configure Server Management... Test Connection Cancel OK

- Connection Name : 본인이 해당 커넥션이 어떤건지 알아보기 쉽게 설정
- Connection Method : Standard TCP/IP 그대로 사용
- Hostname : j7e102.p.ssafy.io 으로 설정 (도메인 이름)
- Username : 계정 ID 으로 설정
- Password : 계정 PW 으로 설정

3. Test Connection으로 연결 확인

4. 연결 테스트가 성공하면 OK 누르고 사용

## 5.2. (공통)Spring Boot에서 연결

- **application.properties**에서 DB 연결 구문을 아래와 같이 수정

```
spring.datasource.url=jdbc:mysql://[서버도메인]/[스키마 명]?serverTimezone=Asia/Seoul
spring.datasource.username=[계정 ID]
spring.datasource.password=[계정 PW]
```

## 6. Nginx default

```
server {
    root /home/ubuntu/fe/sample/sampleUI/build;
    index index.html index.htm index.nginx-debian.html;
```

```

server_name j7e102.p.ssafy.io;

location / {
    try_files $uri $uri/ =404;
}

listen [::]:443 ssl ipv6only=on; # managed by Certbot
listen 443 ssl; # managed by Certbot
ssl_certificate /etc/letsencrypt/live/j7e102.p.ssafy.io/fullchain.pem; # managed by Certbot
ssl_certificate_key /etc/letsencrypt/live/j7e102.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 ($host = j7e102.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80 default_server;
    listen [::]:80 default_server;

    server_name j7e102.p.ssafy.io;
    return 404; # managed by Certbot
}

```

## 7. EC2 Settings

### 7.1. MySQL

#### 7.3.1. Install

아래 명령어를 통해 MySQL 서버 설치

```

$ sudo apt-get update
$ sudo apt-get install mysql-server #접속
$ sudo mysql

```

#### 7.1.2. 계정 생성 및 권한 부여

```
mysql> use mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select Host, User, authentication_string from user;
+-----+-----+-----+
| Host      | User           | authentication_string |
+-----+-----+-----+
| localhost | debian-sys-maint | $A$005$QG HqYoA |
| localhost | mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| localhost | mysql.session   | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| localhost | mysql.sys       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED |
| localhost | root           |                      |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> CREATE USER 'name'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'name'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> EXIT
Bye
```

```
CREATE USER '사용할 유저이름'@'%' IDENTIFIED BY '사용할 유저 비밀번호';
GRANT ALL PRIVILEGES ON . TO '사용할유저이름'@'%' WITH GRANT OPTION => 권한 부여
```

### 7.1.3. 외부 접속 허용

1. 아래 명령어로 스크립트 실행

```
$ cd /etc/mysql/mysql.conf.d
sudo nano mysqld.cnf
```

```
# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address = 0.0.0.0
mysqlx-bind-address = 127.0.0.1
```

2. 해당 부분을 변경해준다.  
또한 MySQL의 사용포트인 3306포트를 방화벽에서 제한 해제해준다.

```
$ sudo ufw allow 3306
$ sudo systemctl restart mysql.service
```

## 7.2. Nginx

### 7.2.1. 설치

1. 아래 명령어로 설치



```
sudo apt-get install nginx
```

## 7.2.2. SSL 인증서 발급

1. 아래 명령어로 nginx certbot 툴 설치

```
sudo add-apt-repository ppa:certbot/certbot
```

```
sudo apt-get install python3-certbot-nginx
```

### 2. CertBot 실행 - SSL 인증서 발급

- a. 아래 명령어 실행으로 CertBot 실행

```
sudo certbot --nginx -d 자신의도메인
```

- b. 이후 이메일과 약관 동의 하면 1,2 중 선택 하라고 한다. 각 내용은 다음과 같다.
  - i. 1번 선택 : http 요청을 https로 리다이렉션 하지 않는다
  - ii. 2번 선택 : http 요청을 https로 리다이렉션 한다.
- c. 선택 후 nginx가 재시작 되면서 https(SSL 인증서) 적용이 완료된다.

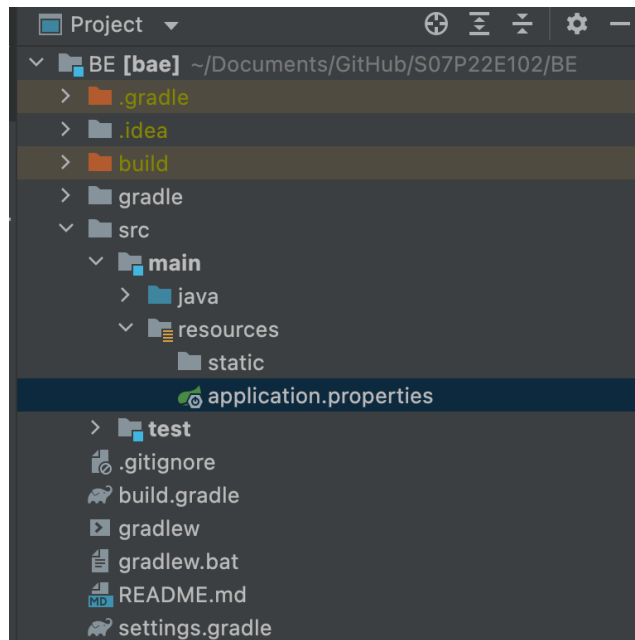
## 8. Files ignored

### 1. BackEnd - application.properties

※ DB 계정 등 보안 관련 정보 때문에 별도 관리

- a. 파일 위치

- [BackEnd Folder] - src - main - resources - application.properties



b. 파일 내용

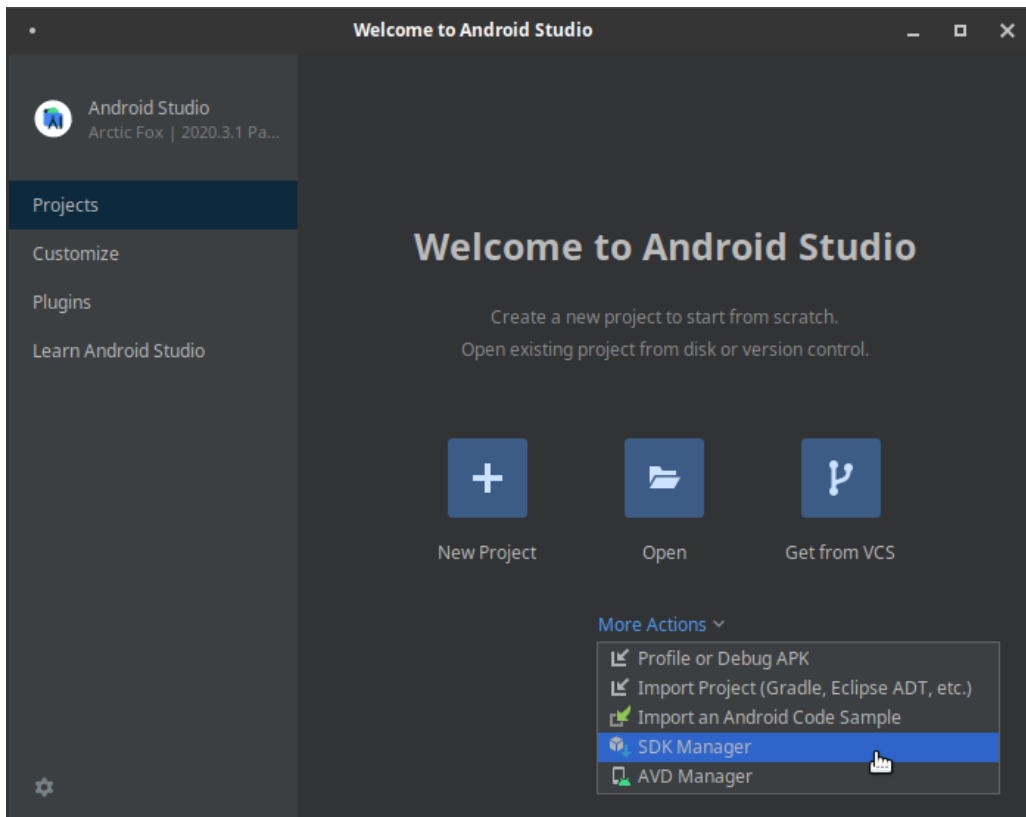
- DB(Schema) Auth Info
- JPA Setting
- Swagger
- Logging

## 9. etc) Tips

※ Android Emulator 에서 해당 프로젝트를 실행하기

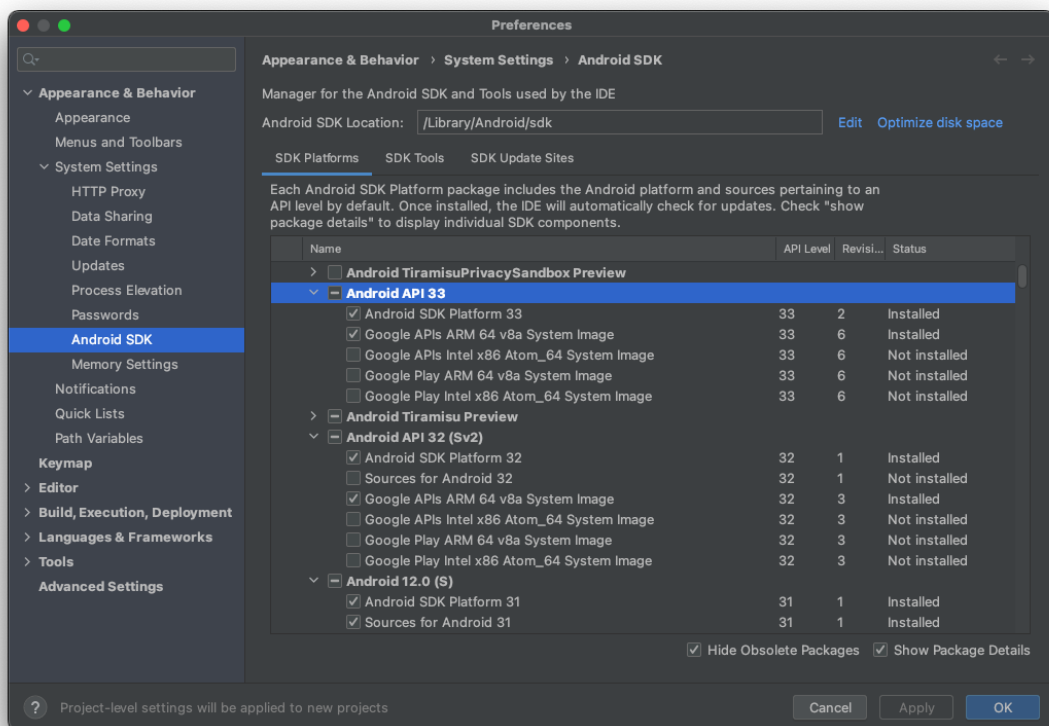
### 9.1. Android Studio 설정 (SDK, ADB)

1. 안드로이드 스튜디오 설치
2. 설치 후 실행시 아래 화면이 보일텐데 여기서 SDK Manager 선택



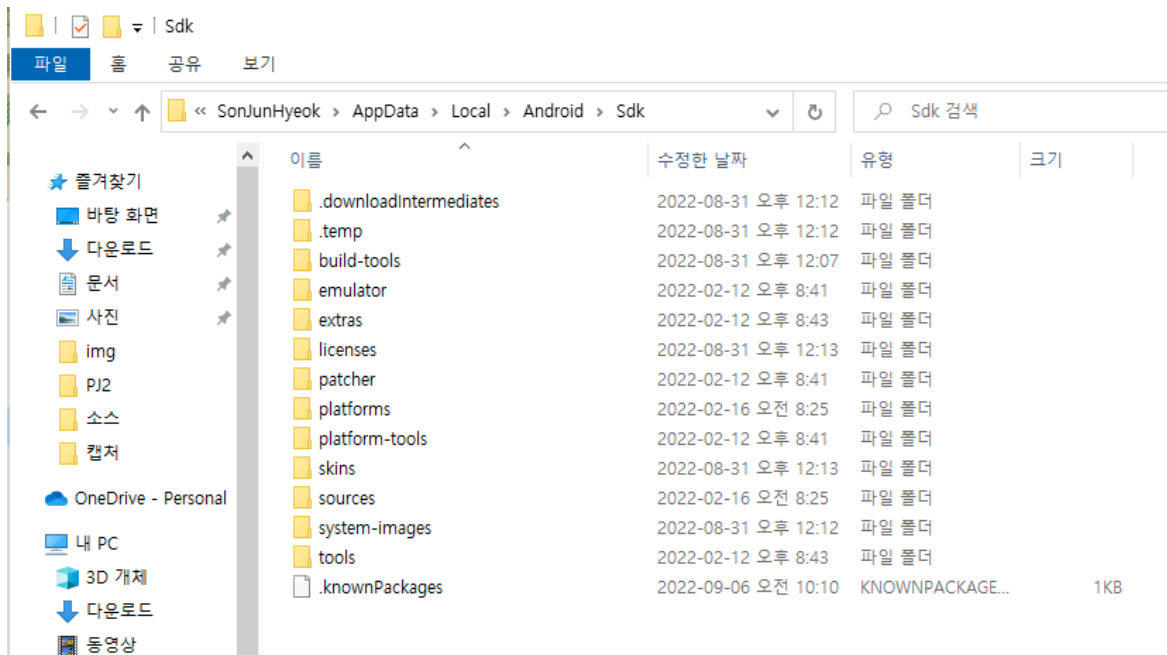
Android Studio 버전마다 조금씩 다르게 보일 수 있음

### 3. SDK platforms에서 원하는 API 버전(OS 버전) 선택 이후 설치 - 여러개도 선택 가능



Android Studio 버전마다 조금씩 다르게 보일 수 있음 (위 사진은 Mac OS 버전)

#### 4. SDK 설치된 장소 확인 하여 정상적으로 설치 되었는지 확인

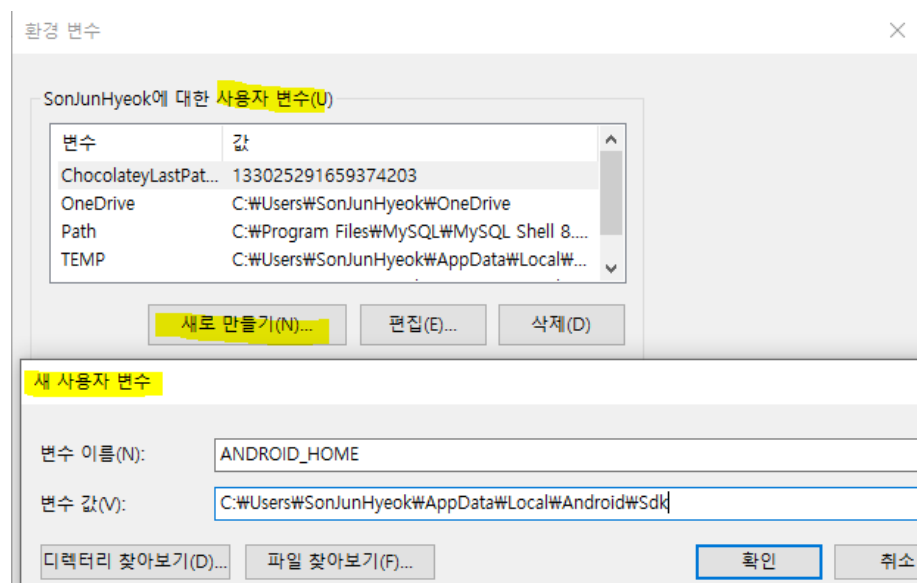


AppData 폴더는 숨겨진 폴더이기에 직접 링크를 쳐야 접속 가능

- 별도의 설치과정에 수정을 하지 않았다면 아래처럼 구성되어 있음
  - C:\Users\사용자\AppData\Local\Android\Sdk
  - 싸트북 기준은 : C:\Users\SSAFY\AppData\Local\Android\Sdk

#### 5. 환경 변수 설정

- 설정 - 정보 - 고급 시스템 설정 - 시스템 속성 - 고급 - 환경 변수 이동 (윈도우 10 기준)
- 사용자 변수에서 새로 만들기 이후 아래처럼 설정

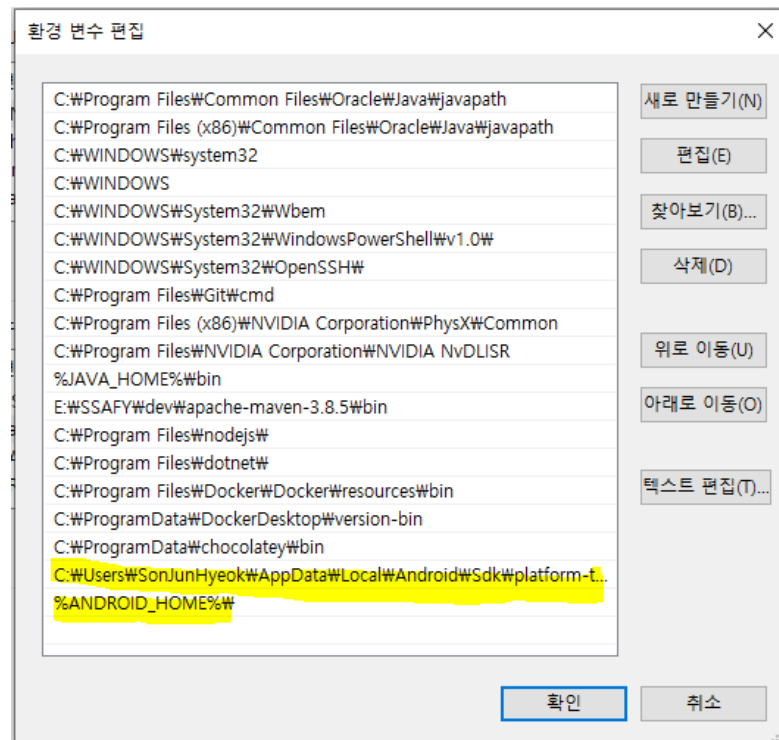


변수 값은 위에 설치한 sdk 폴더값

#### 6. 시스템 변수에서 Path에 아래 두개 추가

- Sdk 폴더 설정한 ANDROID\_HOME

b. Sdk 폴더 아래 있는 platform-tools



7. 저장 후 재부팅 다음에 cmd 창에 adb 실행하여 잘 작동 하는지 확인

## ADB(Android Debug Bridge)

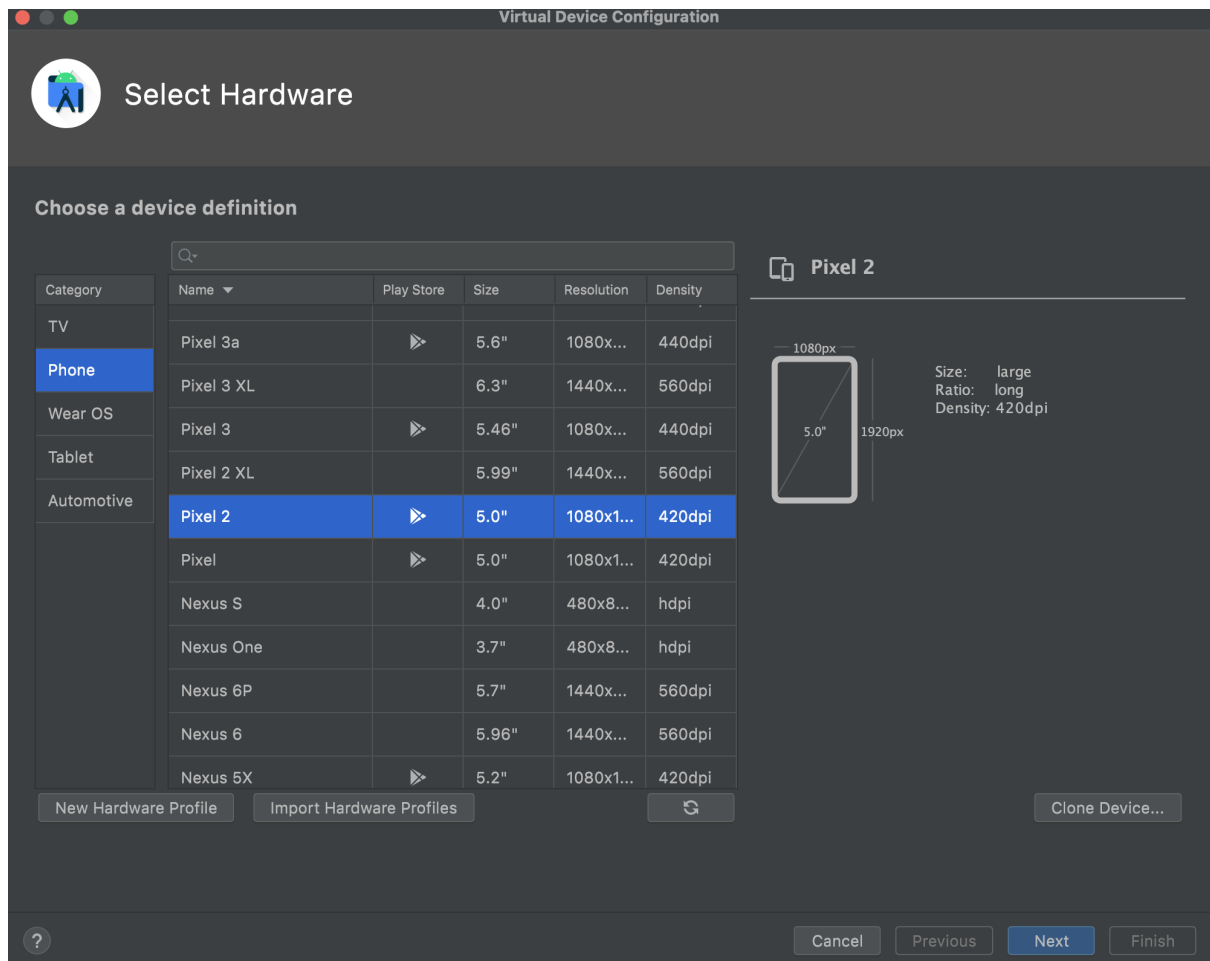
- 아래 링크에서 windows 버전으로 다운로드 및 원하는 위치에 압축해제
  - <https://developer.android.com/studio/releases/platform-tools>
- 해당 폴더 주소를 환경변수에 추가
  - 설정 - 정보 - 고급 시스템 설정 - 시스템 속성 - 고급 - 환경 변수 이동 (윈도우 10 기준)
  - 시스템 변수 - Path에 폴더 주소 추가

## 9.2. Android Emulator 에서 실행

### 9.2.1. Android Emulator 생성

- android studio에서 device manager 에서 create device 클릭
- 아래와 같이 기기 정보를 선택하여 생성 권장 (Pixel 2)

```
- 안드로이드 10(Oreo)
- API : 29
- 기기 정보
  - 해상도 : 1920*1080px
  - 스크린 사이즈 : 5.0 inch
  - RAM : 4GB 이상
  - ROM : 1GB 이상
```



## 9.2.2. Android Emulator 실행

1. cmd를 열어 FE/baepjt 폴더로 이동
2. 필요한 module 설치

```
npm install
```

3. 실행

```
npx react-native run-android
```