



## 수줍(SOOJOOB)

지구와 나의 건강을 지키는

플로깅 어플리케이션

## 포팅 메뉴얼

팀명 | 프리지아

D210

# 목 차

I. 프로젝트 기술 스택.....	2
II. 백엔드 빌드 방법.....	4
III. 프론트엔드 빌드 방법.....	6
IV. 배포 명령어 정리.....	9
V. MySQL 워크벤치 사용방법.....	12
VI. Google console 설정.....	13
- Firebase auth	
- Google map	
VII. 환경 변수 설정.....	20

## I . 프로젝트 기술 스택

### • Backend

- IntelliJ 2022.1.3
- spring boot 2.6.1
- Gradle 7.5
- spring-boot-jpa
- spring security
- JDK 1.8
- AWS EC2
- MySQL 8.0.28

### • Frontend

- Android studio 2021.2.1
- JDK 1.8
- retrofit 2.9.0
- Gradle Plugin Version 7.2.1
- Gradle Version 7.3.3
- TTS

### • 배포

- aws ec2
- Nginx

## 표. 백엔드 빌드 방법

### 1. Command 버전

먼저 스프링 부트 프로젝트 폴더로 이동

```
./gradlew build
```

빌드(gradlew build) 명령을 실행하면

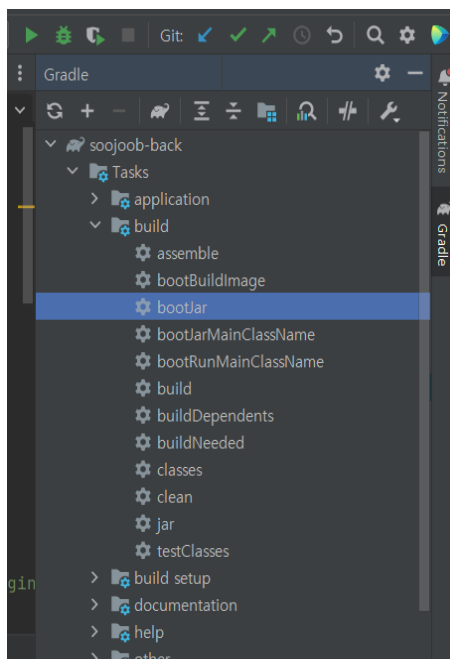
build/libs 폴더에 빌드된 파일(jar)을 확인할 수 있습니다.

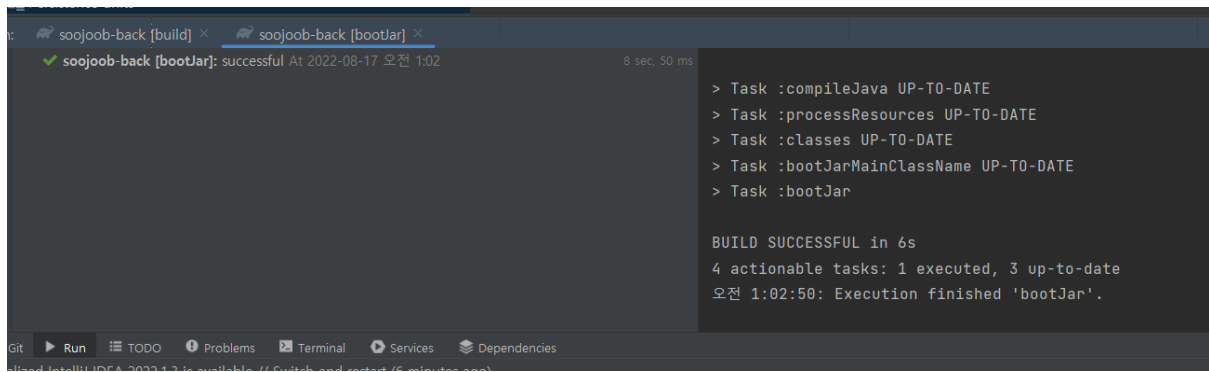
Build/libs 경로로 이동하여 .jar 파일 실행

```
Java -jar soojooob-back-1.0-SNAPSHOT.jar
```

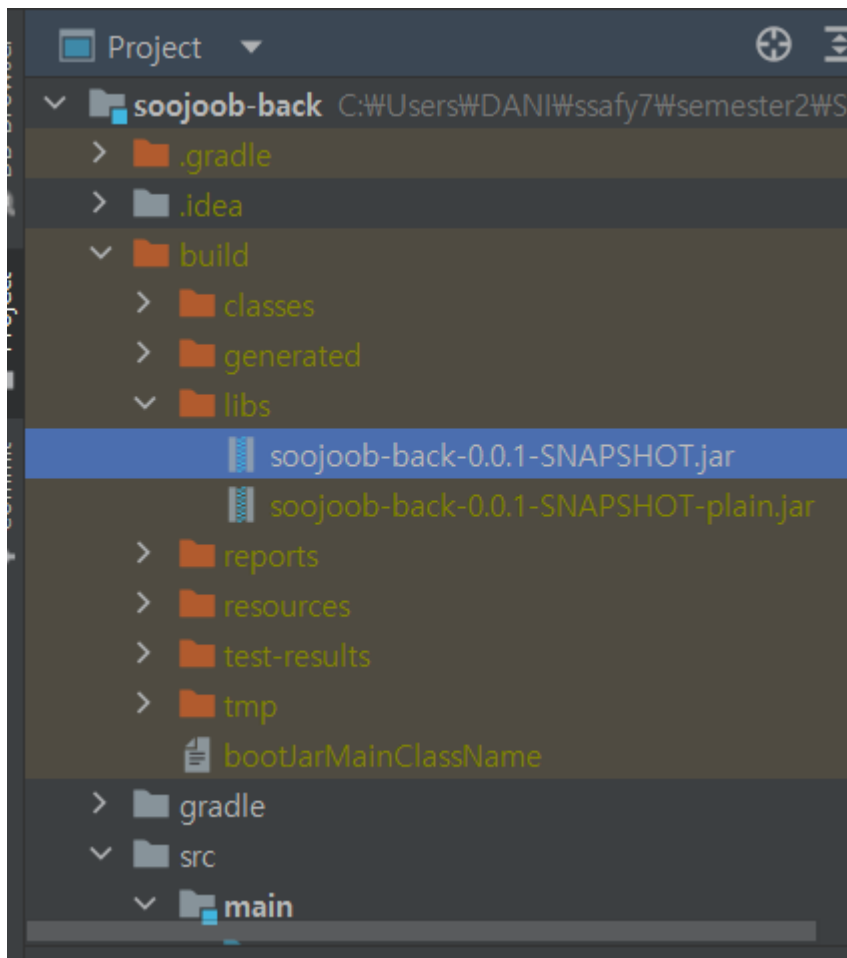
### 2. GUI 버전

IntelliJ 오른쪽 Gradle 을 열어주고 soojooob-back/Tasks/build/bootJar 실행





정상적으로 실행된 모습



Soojoob-back/build/libs 에 jar 파일 생성 완료

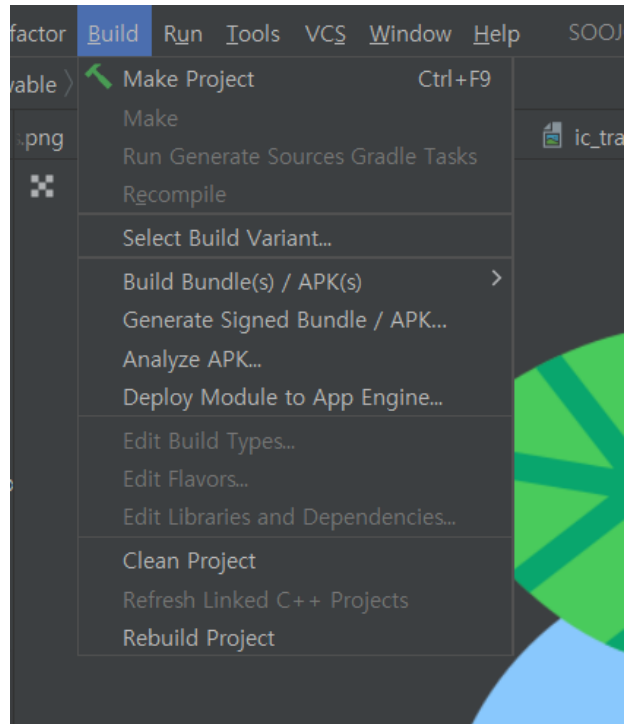
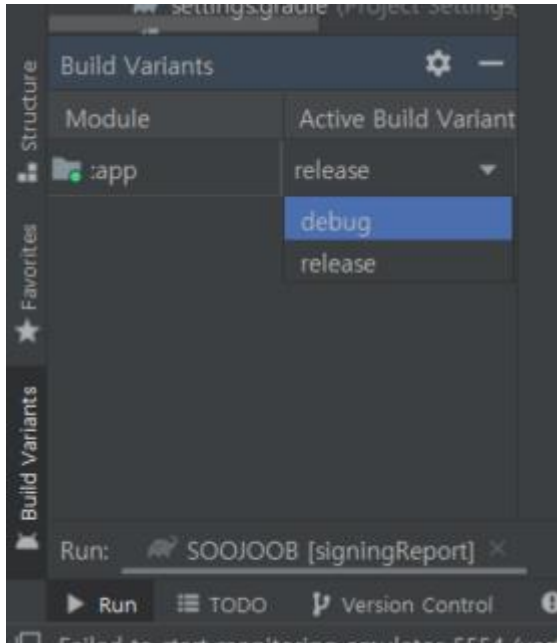
파일이 위치한 곳으로 경로 이동 후 .jar 파일 실행

```
Java -jar soojoob-back-1.0-SNAPSHOT.jar
```

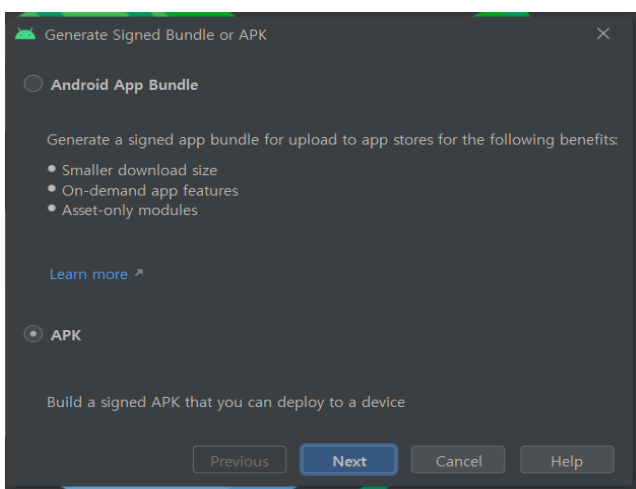
## Ⅲ. 프론트엔드 빌드 방법

### 1. Release 버전

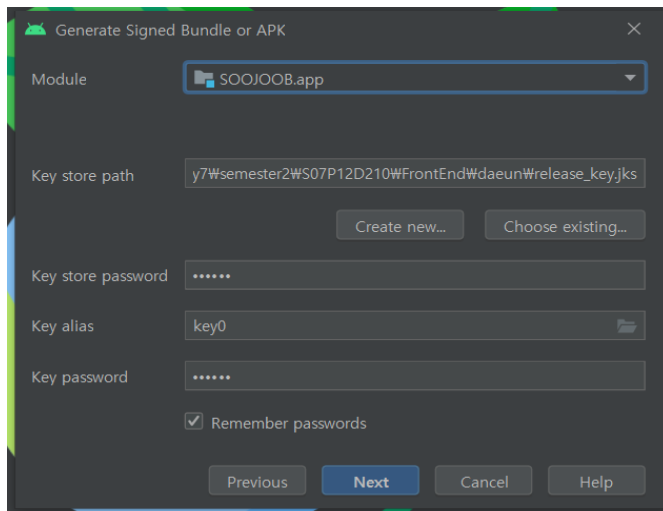
debug 버전을 release 버전으로 바꿔주기(proguard 코드 난독화 적용)



이후 상단바에서 Build -> Generate Signed Bundle / APK.... 클릭

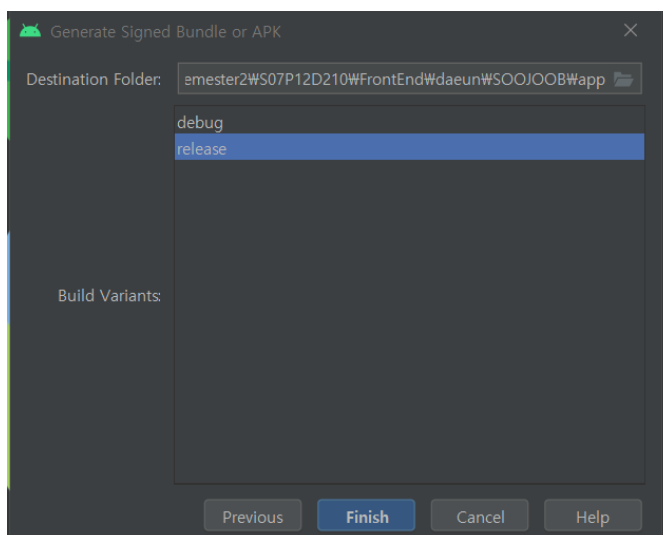


APK 선택 후 Next

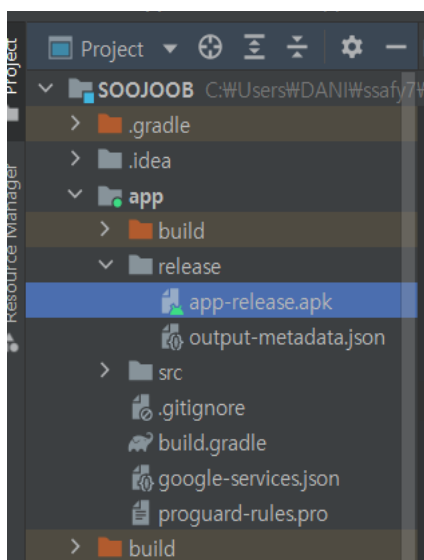


Key store path 에서 Create new... 선택 후 키 저장할 경로와 Key 정보를 설정

다음 업데이트에 password 를 다시 입력하지 않으려면 Remember password 체크 후 Next



release 버전 선택 후 Finish



SOOJOOB/app/release 경로에 app-release.apk 생성

## 2. SHA-1

Gradle 의 Tasks 에서 수행하는 signingReport 는 디버깅용 SHA-1 key 이므로

release 용 SHA-1 을 firebase 및 google console 에 등록해야 함

### i) Release 모드 SHA-1 발급 방법

- cmd 창에서 자바 경로로 들어가기

# 예시

```
cd C:\Program Files\Java\jdk-11.0.15\bin
```

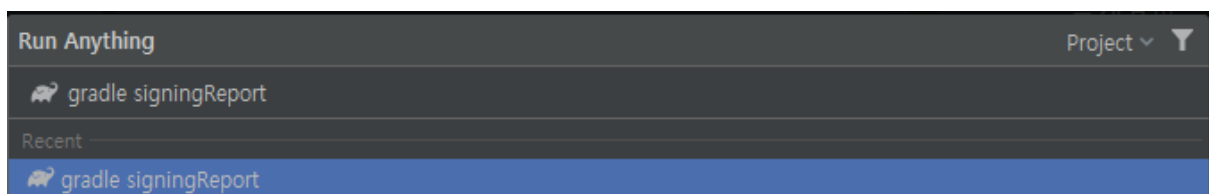
# 예시

```
keytool -list -v -keystore <key store path>
```

- key store path 로 SHA-1 생성 및 key 비밀번호 입력

### ii) Debug 모드 SHA-1 발급 방법

- Android Studio 에서 Ctrl 버튼 2 번 눌러 Run Anything 창 출력
- Run Anything 창에 "gradle signingReport" 명령어 입력 후 SHA1 Key 확인



- 원상복구 : Ctrl 버튼 2 번 누른 후 "App" 입력 후 엔터





## IV. 배포 명령어 정리

### # EC2 접속

\* ssh -i <pem 키> ubuntu@<public ip>

\* remote host : ubuntu@<public ip>

\* private key : pem 키 사용

\* ubuntu 계정 비밀번호 초기화

```
# bash  
  
$ sudo passwd root
```

\* root 계정 비밀번호 초기화

```
# bash  
  
$ sudo su -  
  
$ password ubuntu
```

\* home/ubuntu 디렉토리에서 작업

\* # Nginx 설치 관련 명령어

```
$ sudo apt-get update  
$ sudo apt install nginx -y  
$ nginx -v  
$ sudo service nginx status  
$ sudo service nginx start // nginx 서비스 실행  
  
$ cd /etc/nginx/site-available/default  
$ sudo vi default //Key 및 server 정보 확인  
$ sudu service nginx restart // 서비스 재실행
```

\* # EC2 에 MySQL 설치

```
$ sudo apt-get update
$ sudo apt-get install mysql-server

// 접속
$ sudo mysql -u root -p
// DB 생성
$ create database <DB 이름>;

// 사용자 생성 및 권한 부여
$ create user '{username}'@'{host}' identified by
```

\* # EC2 서버에 jar 파일 업로드

\* .jar 파일 /home/ubuntu 디렉토리에 업로드

\* .jar 파일 배포

```
// Spring 프로젝트 실행
$ java -jar 빌드파일.jar

// 백그라운드에서 세션연결 유지하며 실행
$ nohup java -jar 빌드파일.jar&
```

\* 프로세스 확인 및 종료 명령어

```
// Spring 프로젝트 실행
$ java -jar 빌드파일.jar $ ps -ef | grep <jar-name>.jar // <jar-name>.jar 파일 프로세스 정보 확인

$ kill -9 <PID> //해당 프로세스 종료
```

\* # HTTPS 설정

\* ssl 인증서 발급

```
# letsencrypt 설치하기
sudo apt-get update
sudo apt-get install letsencrypt

# 인증서 발급
# sudo letsencrypt certonly --standalone -d 도메인[팀 ID.p.ssafy.io]
sudo letsencrypt certonly --standalone -d i6f101.p.ssafy.io

# 이메일 입력 및 안내 사항에 동의 후 진행

# root 계정 로그인
sudo su

# 인증서 위치 폴더 이동
cd /etc/letsencrypt/live/도메인

# pem 을 PKCS12 형식으로 변경
# key 파일 비밀번호 입력
```

\* Nginx 를 통해 Cerbot 으로 발급받기

```
$ certbot --nginx -d <public ip>
```

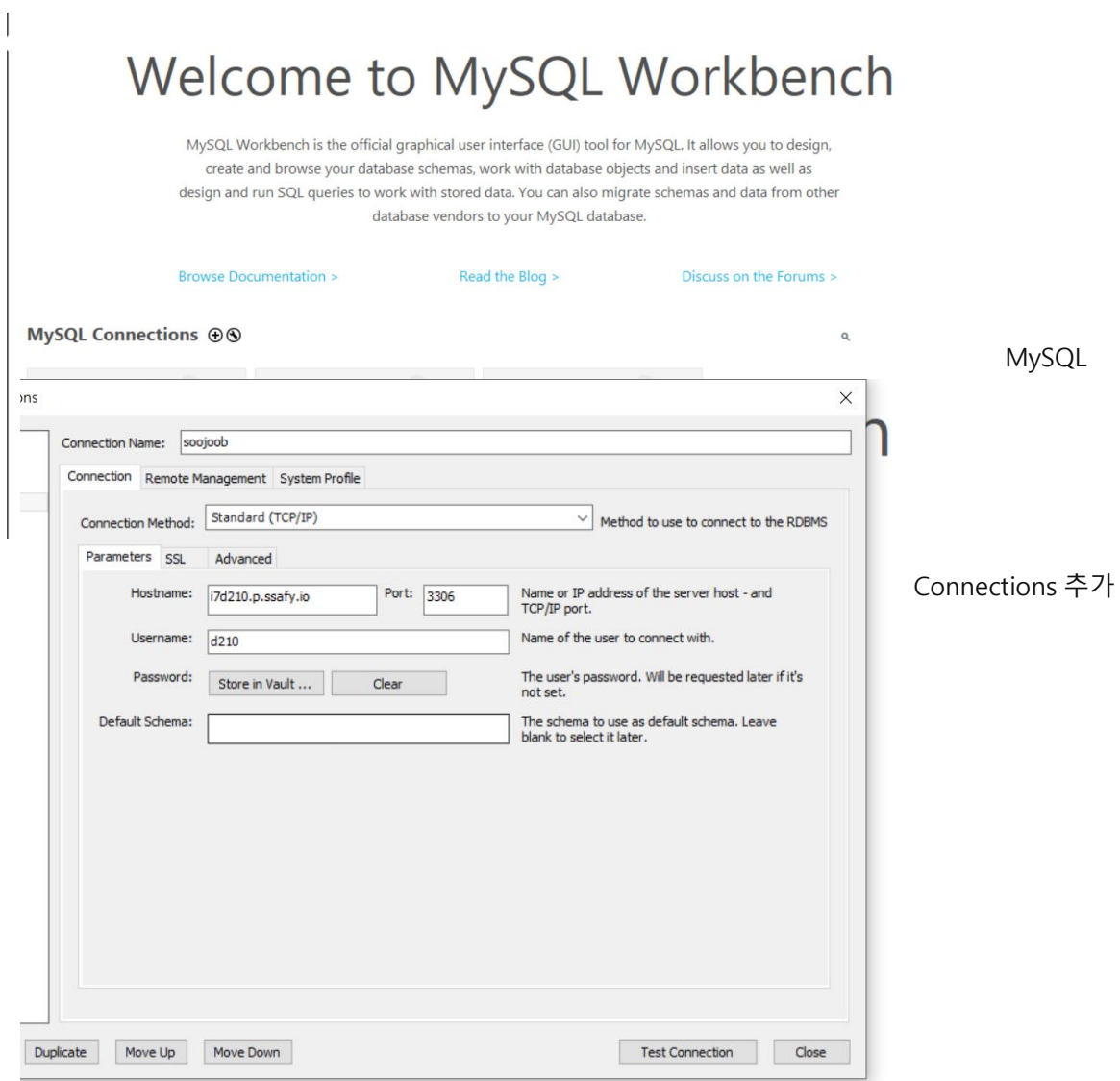
\* /etc/nginx/sites-enabled/default 에서 certificate 세팅 설정 관련 내용 확인

```
$ sudo certbot renew --dry-run // 인증서 자동 갱신
```

## V. MySQL 워크벤치 사용방법

### 1. 데이터베이스 접속 정보

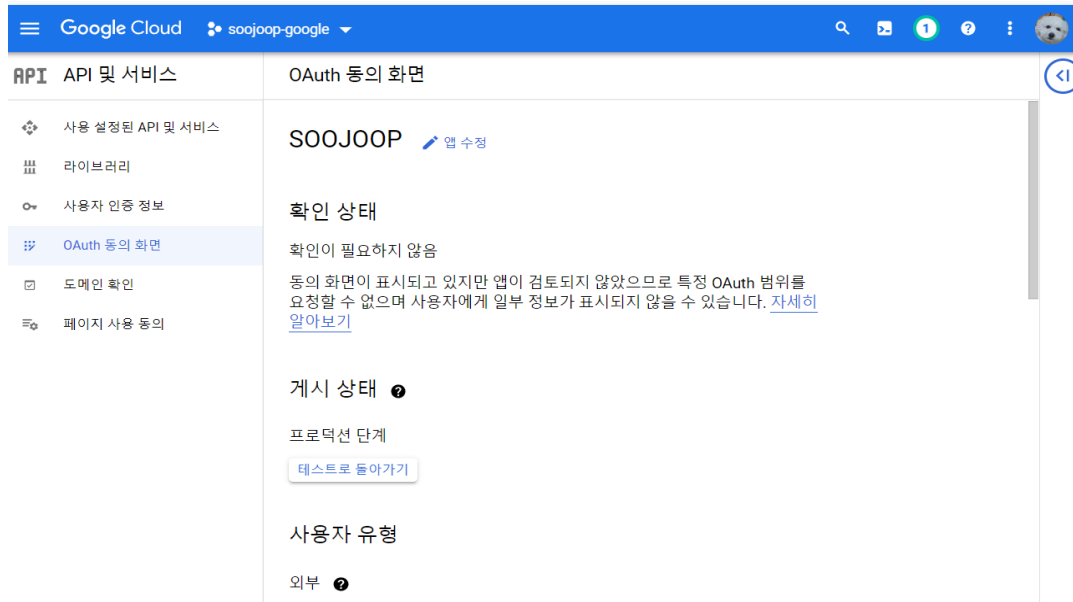
- hostname: i7d210.p.ssafy.io
- port: 3306
- username: d210
- password: tjrgus777!



## VI. Google console 설정

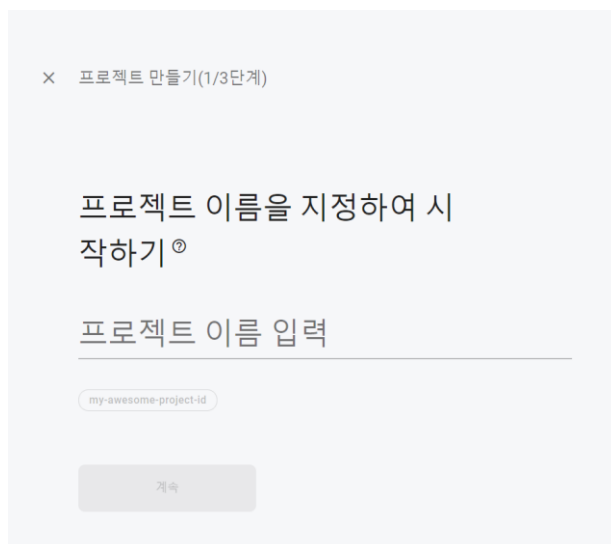
### 1. Firebase auth

Google console developer 에서 프로젝트 생성 후 OAuth 동의 설정



firebase 에서 프로젝트 생성

애널리틱스 구성을 Default Account for Firebase 로 구성



× 프로젝트 만들기(3/3단계)

## Google 애널리틱스 구성

Google 애널리틱스 계정 선택 또는 만들기 ②

 Default Account for Firebase

이 계정에서 자동으로 새 속성 만들기 

프로젝트를 만들면 선택한 Google 애널리틱스 계정에 새 Google 애널리틱스 속성이 생성되고 Firebase 프로젝트에 연결됩니다. 이 연결을 통해 제품 간에 데이터 흐름이 활성화됩니다. Google 애널리틱스 속성에서 Firebase로 내보낸 데이터에는 Firebase 서비스 약관이 적용되지만 Google 애널리틱스로 가져온 Firebase 데이터에는 Google 애널리틱스 서비스 약관이 적용됩니다. [자세히 알아보기](#)

이전

## 프로젝트 만들기

## Android 앱에 Firebase 추가해주기

SHA-1 키는 debug 모드에서 발급받은 키를 임시로 입력

## × Android 앱에 Firebase 추가

## 1 앱 등록

## Android 패키지 이름 ②

com.company.appname

앱 닉네임 (선택사항) ?

## 내 Android 앱

디버그 서명 인증서 SHA-1(선택사항) ②

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:(

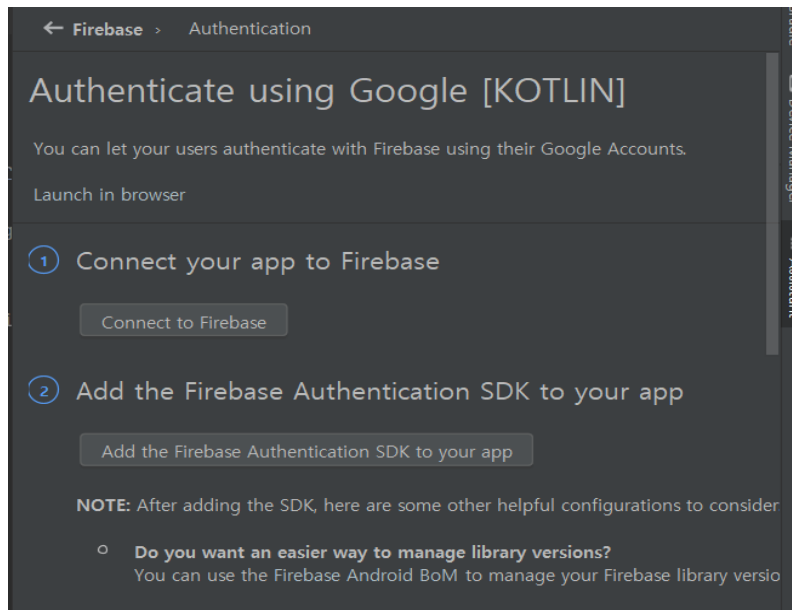
인증에서 동적 링크, Google 로그인, 전화번호를 지원하는 데 필요합니다. 설정에서 SHA-1 을 수정하세요.

google\_services.json 를  
다운로드 후 프로젝트  
app 에 복사

android studio 에서  
firebase 를 연결해줌  
상단바 Tools

➔ firebase

우측메뉴 Authentication -> Authenticate using Google [kotlin]



1 단계와 2 단계 진행 후 완료

debug 모드나 release 모드 등 각 컴퓨터마다 firebase 상에 SHA-1 등록 필요

gso 객체를 생성할때 firebase 프로젝트 client ID 가 필요함

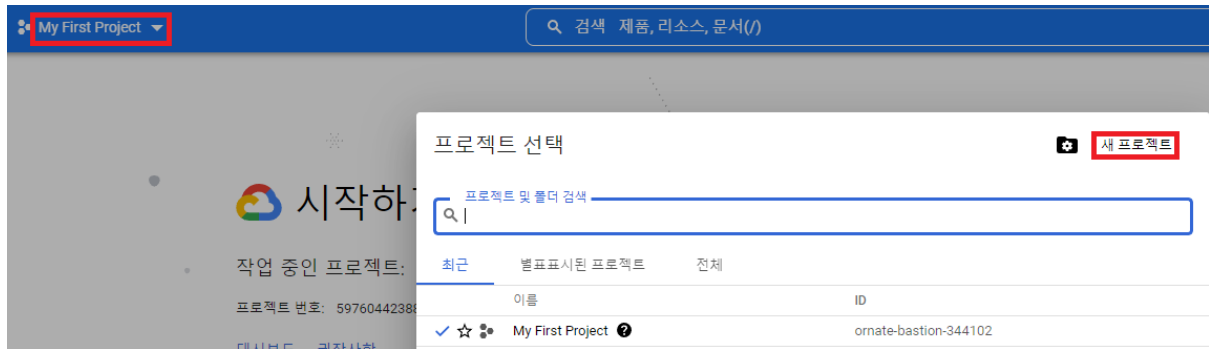
(firebase 등록이 되어서 res(generated)에 values 가 생성되면 정상 작동)

```
// Configure Google Signin
val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.default_web_client_id))
    .requestEmail()
    .build()
```

## 2. Google Map

준비물 : 구글 계정, 해외 결제 카드

- 1) <https://console.cloud.google.com/> 접속
- 2) 새 프로젝트 만들기



프로젝트 이름 \*

S00J00B

프로젝트 ID: soojoo-359812입니다. 나중에 변경할 수 없습니다. 수정

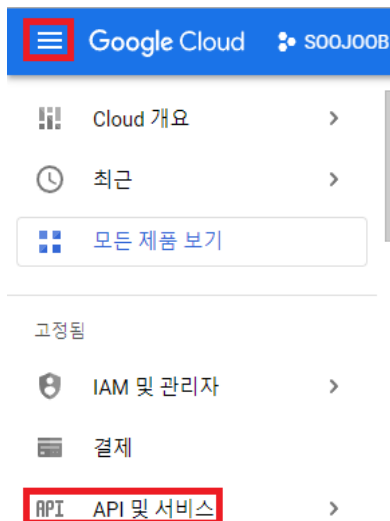
위치 \*

조직 없음

상위 조직 또는 폴더

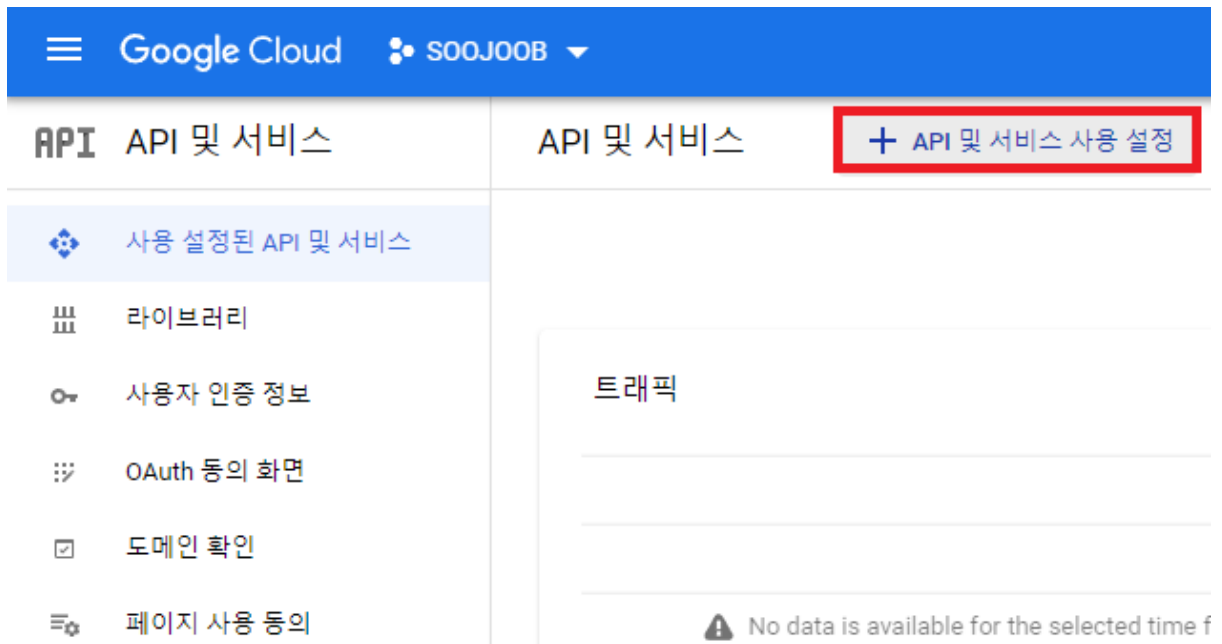
만들기 취소

3) "API 및 서비스 - 사용자 인증 정보" 메뉴 이동

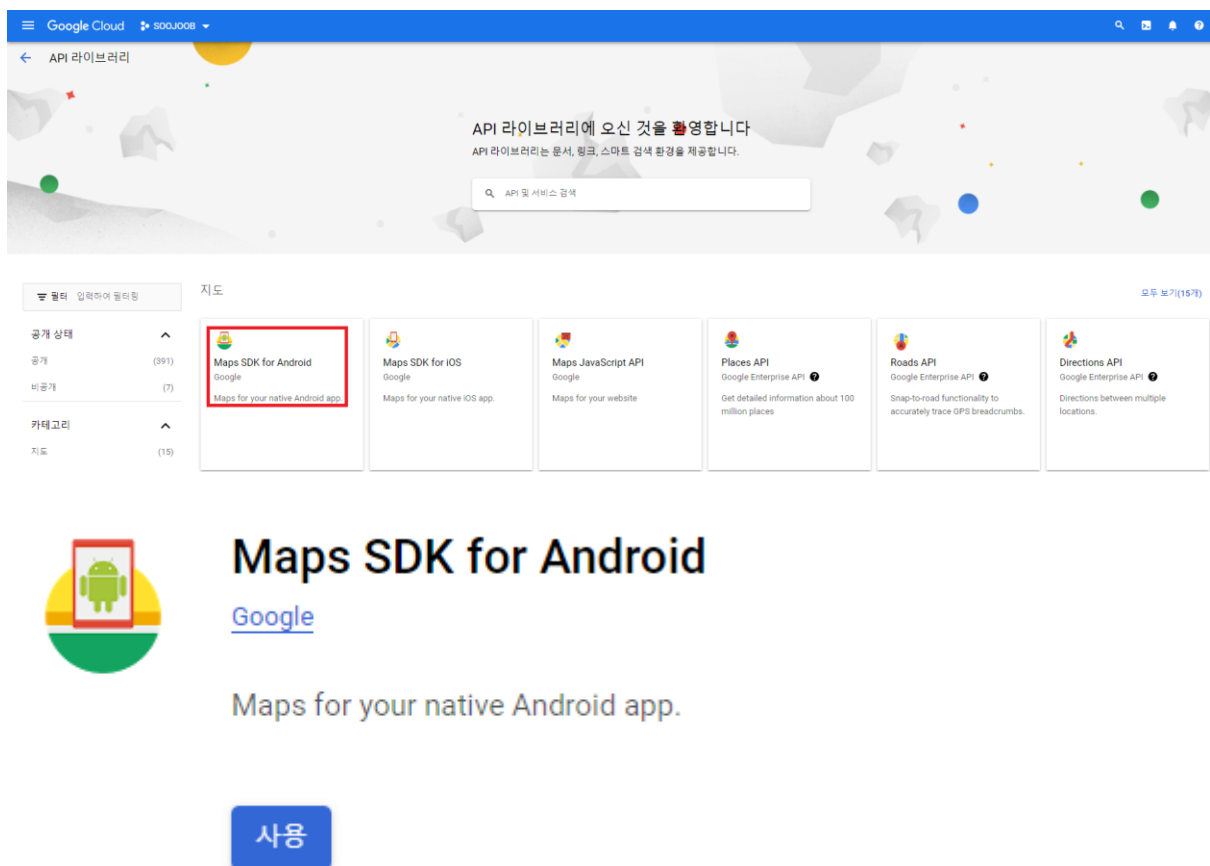




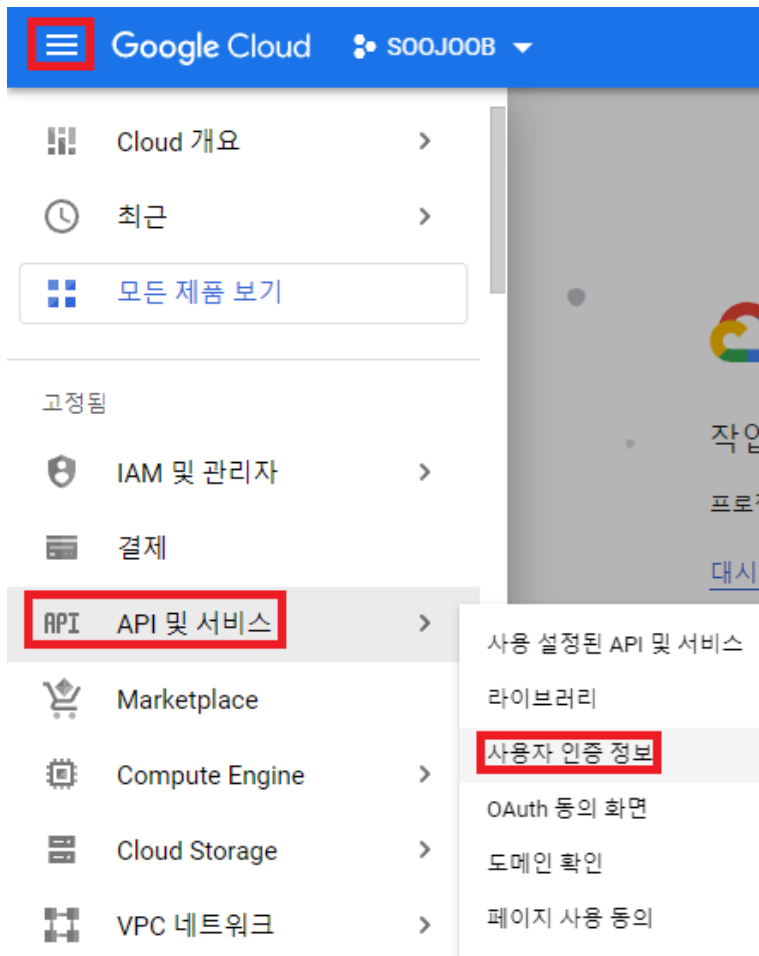
#### 4) API 및 서비스 사용 설정



#### 5) Maps SDK for Android 사용 설정



6) "API 및 서비스 - 사용자 인증 정보" 메뉴 이동



7) 사용자 인증정보 만들기



## 8) API Key 수정 및 설정

API 키

이름	생성일	제한사항 ↑	작업
▲ API 키 1개	2022. 8. 18.	없음	키 표시

OAuth 2.0 클라이언트 ID

이름	생성일 ↓	유형	클라이언트 ID
이름 *			

API 키 수정  
API 키 삭제

이름 \*  
API 키 1개

API Key  
key=API\_KEY 매개변수로 키를 전달하여 애플리케이션에서 이 키를 사용하세요.

생성일  
2022년 8월 18일 PM 10시 3분 40초 GMT+9

키 제한사항

이 키를 사용할 수 있는 위치와 대상을 제한하면 무단 사용을 방지하는 데 도움이 됩니다. [자세히 알아보기](#)

애플리케이션 제한사항

애플리케이션 제한사항은 API 키를 사용할 수 있는 웹사이트, IP 주소 또는 애플리케이션을 제어합니다. 키별로 애플리케이션 제한사항 1개를 설정할 수 있습니다.

☐ 없음  
☐ HTTP 리퍼러(웹사이트)  
☐ IP 주소(웹 서버, 크론 작업 등)  
☒ Android 앱  
☐ iOS 앱

Android 앱의 사용량 제한

Android 앱의 사용량을 제한하려면 패키지 이름과 SHA-1 서명 인증서 디지털 지문을 추가하세요.

새 항목

패키지 이름 \*  
com.example

SHA-1 인증서 디지털 지문 \*

취소 완료

항목 추가

API 제한사항

API 제한사항은 이 키를 호출할 수 있는 사용 설정된 API를 지정합니다.

☐ 키 제한 안함  
이 키는 모든 API를 호출할 수 있습니다.

☒ 키 제한

API 1개

선택한 API:  
Maps SDK for Android

참고: 설정이 적용되는 데 최대 5분이 걸릴 수 있습니다.

저장 취소

출시 인증서 디지털 지문

\$ keytool -list -v -keystore your\_keystore\_name -alias your\_alias

your\_keystore\_name은 .keystore 확장자를 포함하여 키 저장소의 정규화된 경로와 이름으로 바꿉니다. your\_alias\_name은 인증서 생성 시 할당된 별칭으로 바꿉니다.

## 9) AndroidManifest.xml 파일에 Key 등록

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="{google_map_key}"/>
```

10) local.properties 에 코드 1 줄 추가

```
google_map_key="API Key 입력"
```

## VII. 환경 변수 설정

### Backend

```
# src/main/resources/application.properties
```

```
#token secret key
```

```
jwt.token.key = skldjficmnwl123kclknmcnflsdkskmxxfofohue;
```

```
# src/main/resources/application.yml
```

```
spring:
```

```
  datasource:
```

```
    driver-class-name: com.mysql.cj.jdbc.Driver
```

```
    url: jdbc:mysql://i7d210.p.ssafy.io:3306/d210?serverTimezone=Asia/Seoul
```

```
    username: d210
```

```
    password: tjrgus777!
```

```
jpa:
```

```
  hibernate:
```

```
    ddl-auto: none #create update none
```

## Frontend

app/res/generated/values/values.xml

```
<?xml version="1.0" encoding="utf-8"?>

<resources>

    <string name="default_web_client_id" translatable="false">639304711953-
b2abpkijh5f29plt1rk9mo71umepsnmf.apps.googleusercontent.com</string>

    <string name="gcm_defaultSenderId" translatable="false">639304711953</string>

    <string name="google_api_key"
translatable="false">AlzaSyAskQngFQ_t4uNjp5UVk1hwcNa9b8qs6sM</string>

    <string name="google_app_id"
translatable="false">1:639304711953:android:fb12d54c2d6faecd3496b0</string>

    <string name="google_crash_reporting_api_key"
translatable="false">AlzaSyAskQngFQ_t4uNjp5UVk1hwcNa9b8qs6sM</string>

    <string name="google_storage_bucket" translatable="false">soojoop.appspot.com</string>
```

Gradle Scripts/local.properties

```
google_map_key="AlzaSyAS9cn9ixRM5oW2ODVy9UY5aYJbQBe273g"
```