



## **B103 : ECO2**

삼성청년 SW 아카데미 7기 대전 1반 3팀

공동프로젝트 22/07/11 ~ 22/08/19

## **포팅 매뉴얼**

담당 컨설턴트: 김신일

정세미[팀장], 김우원, 김은경, 김은혜, 신영제, 조유진

## 목차

1. 프로젝트 기술 스택.....	1
2. 빌드 상세 내용.....	5
3. MySQL Workbench .....	9
4. 배포 .....	10
5. Nginx Default 값 .....	11
6. EC2 설정 .....	12
7. https 설정 .....	15
8. Redis 설치 .....	16
9. Kakao Dev 설정 .....	19
10. Firebase 설정 .....	22
11. 공공데이터 API key 설정 .....	30

## 1. 프로젝트 기술 스택

가. 이슈관리: Jira

나. 형상관리: GitLab

다. 커뮤니케이션: MatterMost, Notion, Webex

라. 개발환경

1) OS: Window 10

2) IDE

- IntelliJ
- Visual Studio Code 1.70.0

3) Deploy

- AWS EC2
- NGINX

4) Frontend

- HTML, CSS, JavaScript
- React 18.2.0
- node.js 16.13.2
- redux-toolkit 1.8.3

5) Backend

- java 11
- Springboot: 2.7.1
- Spring Data JPA
- Spring Mail
- Spring Web

- Swagger: 2.9.2
- Spring Security
- Spring Validation
- jdk: zulu-11 11.56.19
- Springboot Websocket
- Stomp WebSocket: 2.3.3-1
- JWT: 0.11.5
- Springboot Redis
- Springboot Gradle
- Firebase-admin: 9.0.0

6) Database

- MySQL Workbench 8.0.29

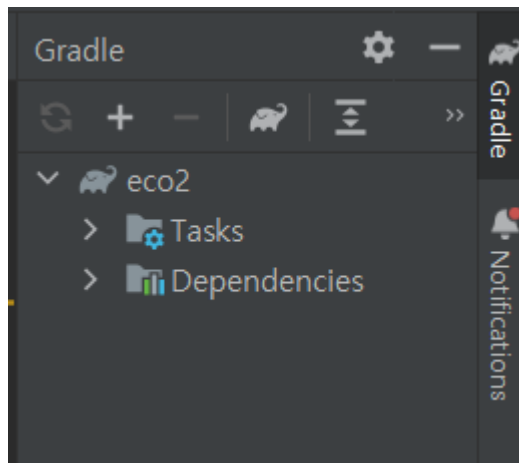
## 2. 빌드 상세 내용

### 가. Backend 빌드 방법

- 1) eco2 의 backend/eco2 를 gradle 로 import 합니다.

이름	수정한 날짜	유형
.idea	2022-08-15 오후 6:22	파일 폴더
eco2	2022-08-17 오전 11:15	파일 폴더
.gitkeep	2022-08-13 오후 3:33	GITKEEP 파일
REST API Docs	2022-08-13 오후 3:33	파일

- 2) import 시에 자동으로 gradle 빌드가 이루어집니다. 만약 build 가 진행되지 않는다면 우측 gradle 의 Reload All Gradle Projects 를 클릭합니다.



- 3) src/main/resources/application-api-key.properties 을 추가합니다.

Database 에 생성한 계정, 비밀번호를 입력합니다. 인증메일을 보낼 google User 의 email 과 password 값을 입력합니다. 발급받은 kakao client id, secret 과 API key 를 입력합니다.

[illegible]

4) Eco2Application 에서 main 을 실행합니다.

```
1 usage  👤 eunkyoung
5 @SpringBootApplication
7 ▶ public class Eco2Application {
   👤 eunkyoung
8 ▶ ⊕ public static void main(String[] args) { SpringApp
1
2
```

나. Frontend 빌드 방법

- 1) eco2 의 frontend/eco2 를 import 합니다.
- 2) frontend/eco2 위치에 .env, .env.development, .env.prodeuction 파일을 추가합니다.
  - .env → Firebase, Google, Kakao 에서 발급받은 key 를 입력합니다.

```
REACT_APP_FIREBASE_API_KEY=  
REACT_APP_FIREBASE_AUTH_DOMAIN=  
REACT_APP_FIREBASE_DB_URL=  
REACT_APP_FIREBASE_PROJECT_ID=  
REACT_APP_FIREBASE_STORAGE_BUCKET=  
REACT_APP_FIREBASE_MESSAGING_SENDER_ID=  
REACT_APP_FIREBASE_APP_ID=  
REACT_APP_GOOGLE_CLIENT_ID=  
REACT_APP_KAKAO_REST_API_KEY=  
REACT_APP_KAKAO_JAVASCRIPT_KEY=  
api.weather.key=
```

- .env.development

```
REACT_APP_BE_HOST='http://localhost:8002/'  
REACT_APP_KAKAO_REDIRECT='http://localhost:3000/kakao'
```

- .env.production

```
REACT_APP_BE_HOST='https://i7b103.p.ssafy.io:8002/'  
REACT_APP_KAKAO_REDIRECT='https://i7b103.p.ssafy.io:3000/kakao'
```

3) node\_modules 를 위한 기본 install

```
npm install
```

4) 현재 상태로 시작하기

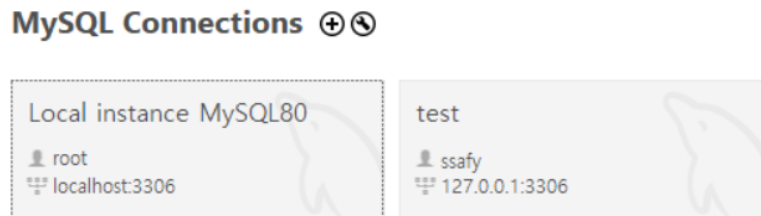
```
npm start
```



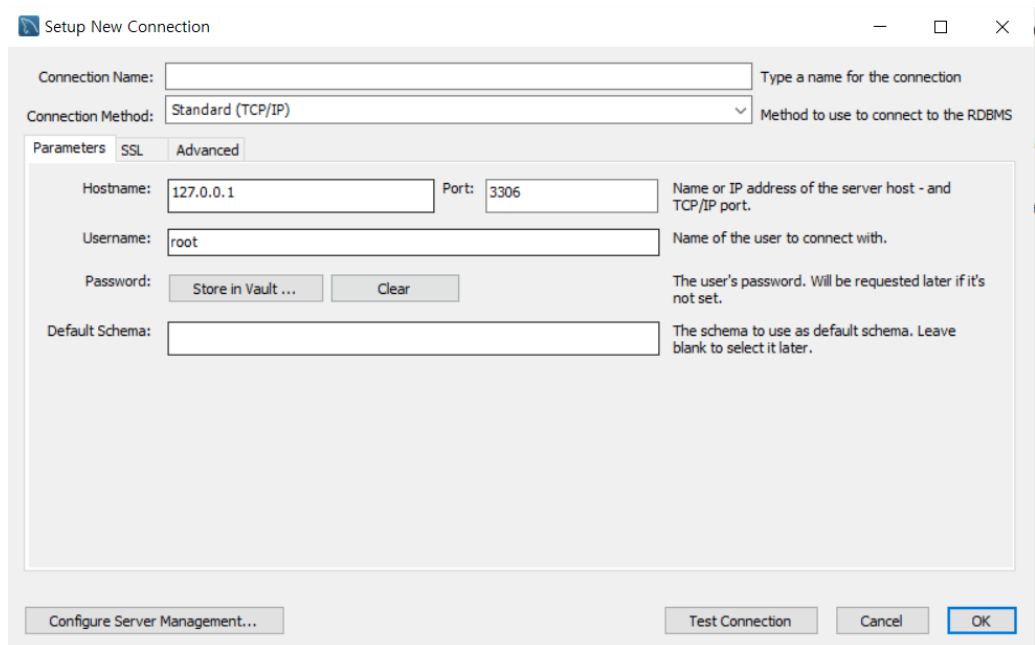
### 3. MySQL Workbench

가. MySQL 설정

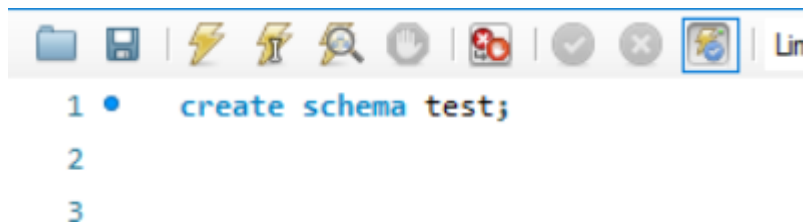
1) MySQL Workbench 추가하기



2) 기존 root 계정이 아닌 별도의 계정을 생성합니다.



3) 스키마를 생성합니다.



## 4. 배포

가. FE directory 이동

터미널에 npm run build 실행

생성된 build 폴더를 ec2 인스턴스 ~ 경로에 업로드

나. BE 폴더를 gradle 로 import 후 gradle bootjar 실행 후 jar 파일 ec2 ~ 경로에 추가

다. ~ 경로에 path 폴더 생성 후, path 에 to 폴더 생성, to 폴더에 serviceAccount.json 업로드

라. ~ 경로에 레포지토리에 올라가 있는 img 폴더 업로드

//jar 파일 실행하는 커맨드

java -jar <jar-name>.jar

//백그라운드 실행 명령어

nohup java -jar <jar-name>.jar &

//현재 jar PID 확인

ps -ef | grep <jar-name>.jar

//PID 종료

kill -9 <PID>

## 5. Nginx default 값

```
server {  
    server_name i7b103.p.ssafy.io;  
    location / {  
        root /home/ubuntu/build;  
        index index.html index.htm;  
        try_files $uri $uri/ /index.html;  
    }  
    listen [::]:443 ssl ipv6only=on; # managed by Certbot  
    listen 443 ssl; # managed by Certbot  
    ssl_certificate /etc/letsencrypt/live/i7b103.p.ssafy.io/fullchain.pem; # managed by  
    Certbot  
    ssl_certificate_key /etc/letsencrypt/live/i7b103.p.ssafy.io/privkey.pem; # managed  
    by Certbot  
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot  
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot  
}  
server {  
    if ($host = i7b103.p.ssafy.io) {  
        return 301 https://\$host\$request\_uri;  
    } # managed by Certbot  
    listen 80;  
    listen [::]:80;  
    server_name i7b103.p.ssafy.io;  
    return 404; # managed by Certbot  
}
```

## 6. EC2 설치

### 가. open jdk 11 설치

#### 1) yum 설치

```
apt-get update  
  
//업데이트 완료 후  
  
apt install yum
```

#### 2) Java 11 jdk 설치

```
apt-get install openjdk-11-jdk  
  
//설치 후 버전 확인  
  
java -version
```

### 나. mySql 설치

#### 1) apt 를 이용해 mySQL 설치

```
sudo apt install mysql-server  
  
//netstat 으로 mysql 실행여부 확인  
  
sudo netstat -tap | grep mysql  
  
(netstat: command not found 발생 시  
=> sudo apt isntall net-tools)
```

#### 2) Database 설정

```
mysql -u root -p  
  
mysql> create user 'root'@ '%' identified by '비밀번호';  
  
mysql> create database test;
```

### 3) Database 접근 권한 주기

```
mysql> grant all privileges on test.* to 'root'@'%';  
mysql> flush privileges;  
mysql> show grants for 'root'@'%';
```

//설정파일 변경

```
cd /etc/mysql/mysql.conf.d
```

```
sudo vi my.cnf
```

//bind-address 를 127.0.0.1 에서 0.0.0.0 으로 변경

//mysql inbound 규칙 설정

```
sudo service mysql restart
```

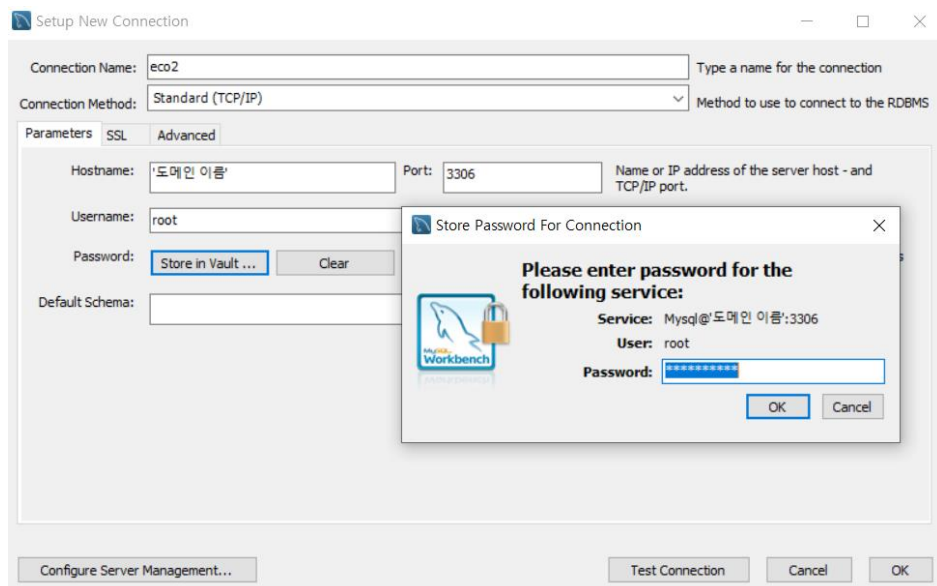
```
sudo ufw allow out 3306/tcp
```

```
sudo ufw allow in 3306/tcp
```

//mySql 재시작

```
sudo service mysql restart
```

### 4) Workbench Connection 확인



```
//workbench 에서 아래 sql 문 실행  
select @@sql_mode;  
SET global sql_mode='STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION'
```

다. redis 설치

```
sudo apt-get install redis-server
```

라. font 설치

1) Fontconfig Package 설치

```
sudo apt install fontconfig
```

2) Font 파일 /usr/share/fonts/에 등록

```
//권한 거부 오류 시  
sudo vi /etc/ssh/sshd_config  
  
//기존 ' SubSystem sftp /usr/lib/openssh/sftp-server ' 를 주석처리 후 '  
SubSystem sftp internal-sftp ' 를 추가 Match 부분이 주석처리가 되어 있을  
경우 주석 해제 후 ' ChrootDirectory /usr/share/fonts/ ' 을 추가  
  
//저장 후  
sudo chown ubuntu:ubuntu /var/www/ -R
```

3) Font 캐시 갱신

```
sudo fc-cache -f -v
```

## 7. https 설정

### 가. CertBot 설치

```
sudo add-apt-repository ppa:certbot/certbot  
$ sudo apt-get install python-certbot-nginx
```

### 나. SSL 설치

```
sudo certbot --nginx -d '도메인 이름'  
//http 연결을 https 로 리다이렉트 하도록 설정
```

### 다. Spring Boot 인증서 설정

- 1) /etc/letsencrypt/live/ 안에 내 도메인 이름으로 되어있는 폴더로 이동
- 2) 내부의 pem 키들을 복사 후, 복사한 디렉토리에서 pkcs12 로 변환

```
openssl pkcs12 -export -in cert.pem -inkey privkey.pem -out  
cert_and_key.p12 -name ttp -CAfile chain.pem -caname root
```





- 3) 생성된 p12 파일 이름을 keystore 로 변경 후, boot 프로젝트 resources 에 추가
- 4) boot 프로젝트 application properties 에 ssl 설정 추가

```
##https  
server.ssl.key-store=classpath:keystore.p12  
server.ssl.key-store-type=PKCS12  
server.ssl.key-store-password=
```

## 8. Redis 설정

### 가. Redis 설치

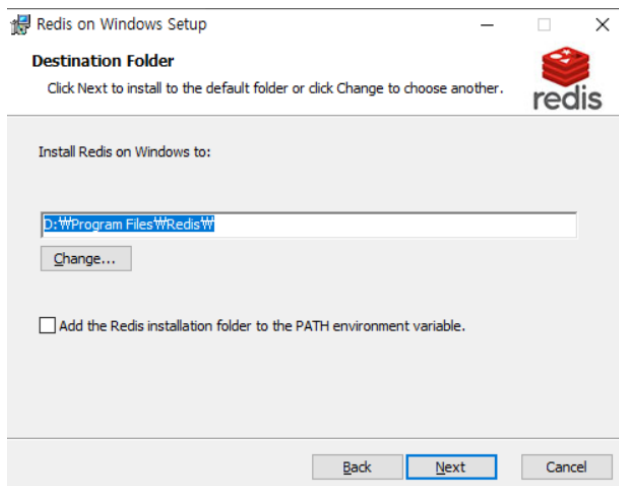
- 1) <https://github.com/microsoftarchive/redis/releases> 에서 msi 파일을 다운로드한다.

▼ Assets 4		
 <a href="#">Redis-x64-3.0.504.msi</a>	6.42 MB	01 Jul 2016
 <a href="#">Redis-x64-3.0.504.zip</a>	5.6 MB	01 Jul 2016
 <a href="#">Source code</a> (zip)		01 Jul 2016
 <a href="#">Source code</a> (tar.gz)		01 Jul 2016

- 2) 설치를 진행한다.



- 3) 설치할 경로를 지정한다.

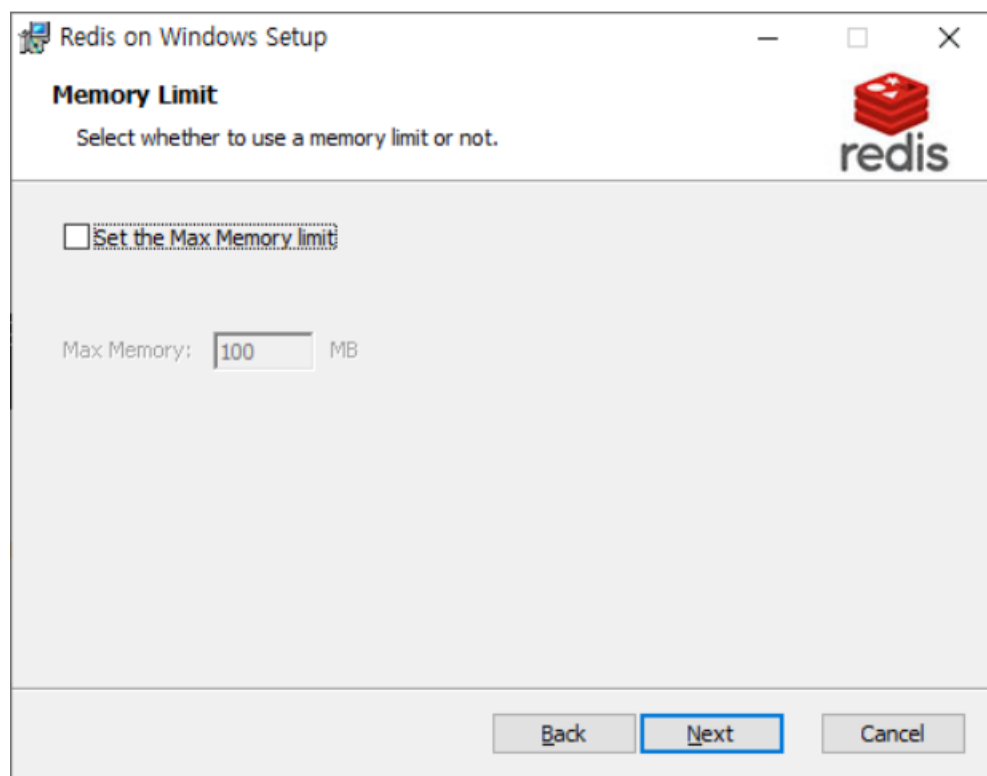




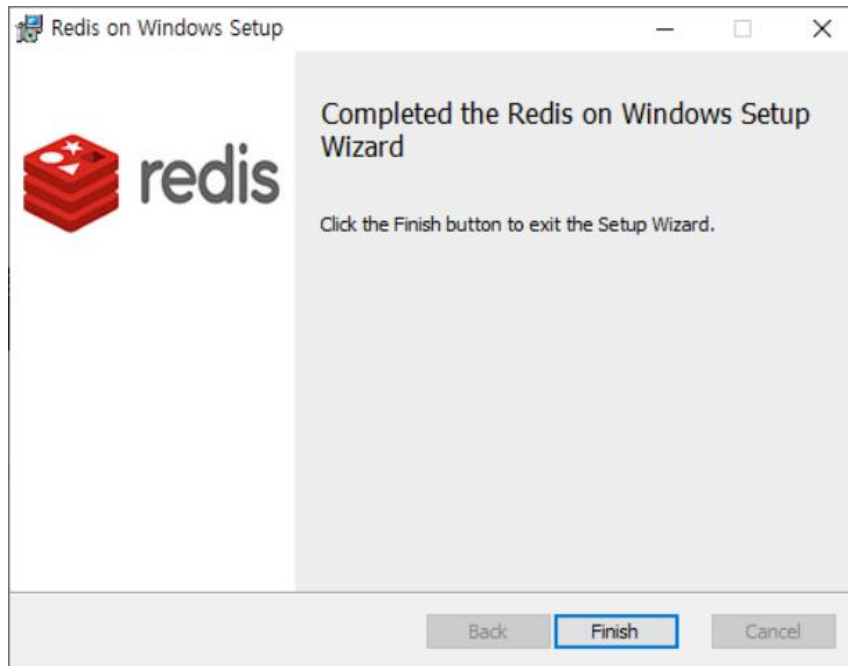
- 4) 포트를 기본포트(6379)로 설정한다.



- 5) Memory Limit 옵션을 지정한 후 설치를 진행한다.



6) 설치를 완료한다.



7) 설치 경로로 이동하여 redis-cli 를 실행한다.

redis-check-aof.pdb	2016-07-01 오후 3:55	Program Debug Datab...	3,436KB
redis-check-dump	2016-07-01 오후 3:55	응용 프로그램	262KB
redis-check-dump.pdb	2016-07-01 오후 3:55	Program Debug Datab...	3,404KB
redis-cli	2016-07-01 오후 3:55	응용 프로그램	471KB
redis-cli.pdb	2016-07-01 오후 3:55	Program Debug Datab...	4,412KB
redis-server	2016-07-01 오후 3:55	응용 프로그램	1,517KB
redis-server.pdb	2016-07-01 오후 3:55	Program Debug Datab...	6,740KB

8) 명령어를 통해 확인한다.

```
C:\Program Files\Redis> redis-cli.exe
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> set test test-value
OK
127.0.0.1:6379> get test
"test-value"
127.0.0.1:6379> _
```

## 9. Kakao Dev 설정

가. 카카오 developers 설정(Oauth, 인가코드 수신)

- 1) <https://developers.kakao.com/console/app>
- 2) 애플리케이션 추가

### 애플리케이션 추가하기

앱 아이콘

이미지 업로드

파일 선택

JPG, GIF, PNG  
권장 사이즈 128px, 최대 250KB

앱 이름

teset

사업자명

ssafy

• 입력된 정보는 사용자가 카카오 로그인 할 때 표시됩니다.

• 정보가 정확하지 않은 경우 서비스 이용이 제한될 수 있습니다.

취소

저장

- 3) 도메인 등록

### Web 플랫폼 등록

#### 사이트 도메인

JavaScript SDK, 카카오톡 공유, 카카오맵, 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>

<http://localhost>  
<http://localhost:3000>  
<https://>

#### 기본 도메인

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

<http://localhost>

취소

저장

#### 4) 로그인 활성화

- 카카오 로그인 활성화

**카카오 로그인** ON

**활성화 설정**

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.

- redirect URL 설정(https 사용 시 빈 포트 사용)

#### 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

```
http://localhost:3000/kakao  
https://localhost/kakad
```

- 동의 항목 설정

#### 동의 항목 설정

항목  
카카오계정(이메일) / account\_email

#### 동의 단계

- ☒ 필수 동의 (검수 필요)  
카카오 로그인 시 사용자가 필수로 동의해야 합니다.
- ☐ 선택 동의  
사용자가 동의하지 않아도 카카오 로그인을 완료할 수 있습니다.

- .env 파일에 kakao client key, secret

### 앱 키

플랫폼	앱 키		재발급
네이티브 앱 키		복사	재발급
REST API 키		복사	재발급
JavaScript 키		복사	재발급
Admin 키		복사	재발급

- REST API 키와 JavaScript 키를 사용합니다.

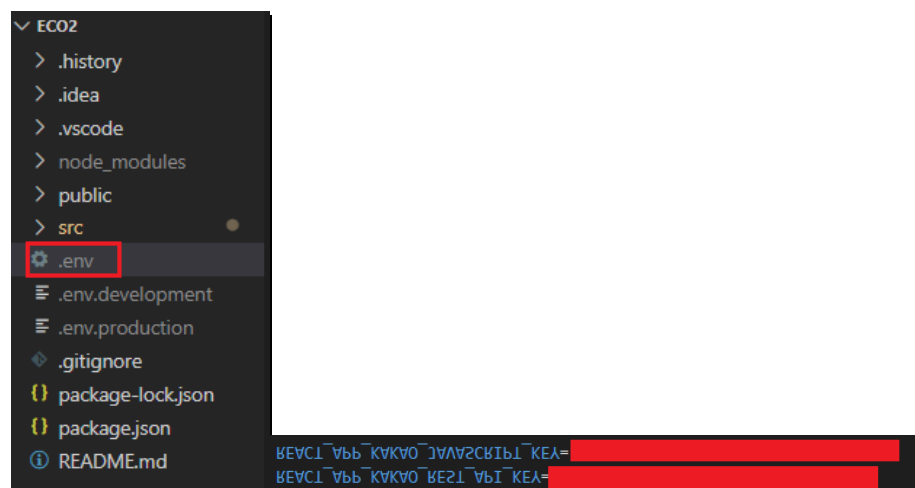
카카오 로그인
ON

Client Secret
삭제

토큰 발급 시, 보안을 강화하기 위해 Client Secret을 사용할 수 있습니다. (REST API인 경우에 해당)

코드		재발급
활성화 상태	사용안함	설정

- 프론트엔드 <프로젝트폴더>/.env 파일



## 10. Firebase 설정

가. 버전 : 9.0

나. 구글 소셜 로그인을 firebase authentication 을 활용하여 구현

다. 실시간 알림 기능을 firestore 를 활용하여 구현

1) Firebase 프로젝트 추가

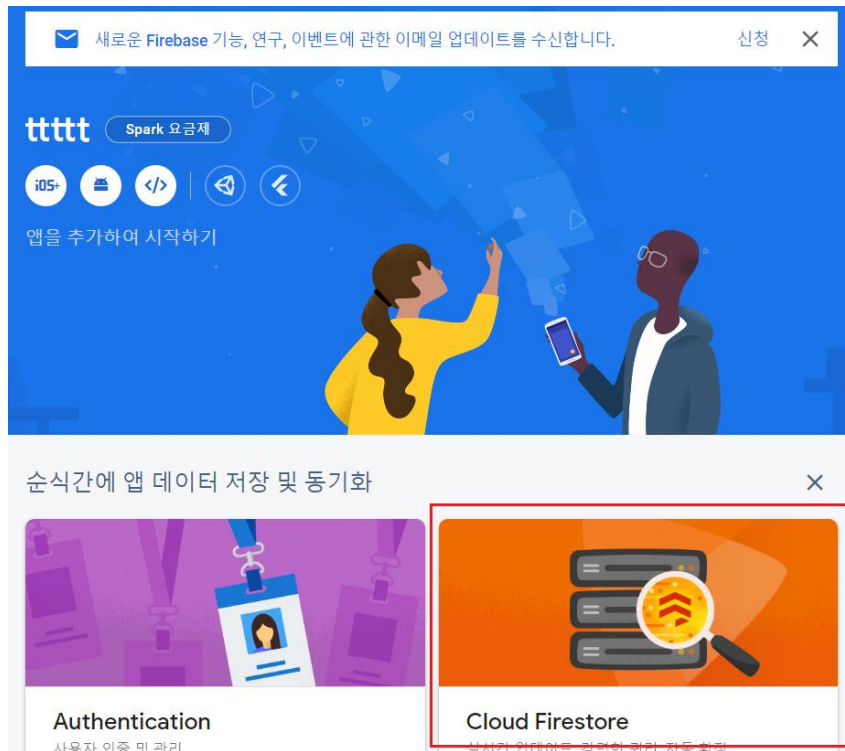
- <https://console.firebase.google.com/>에 접속
- 프로젝트 추가



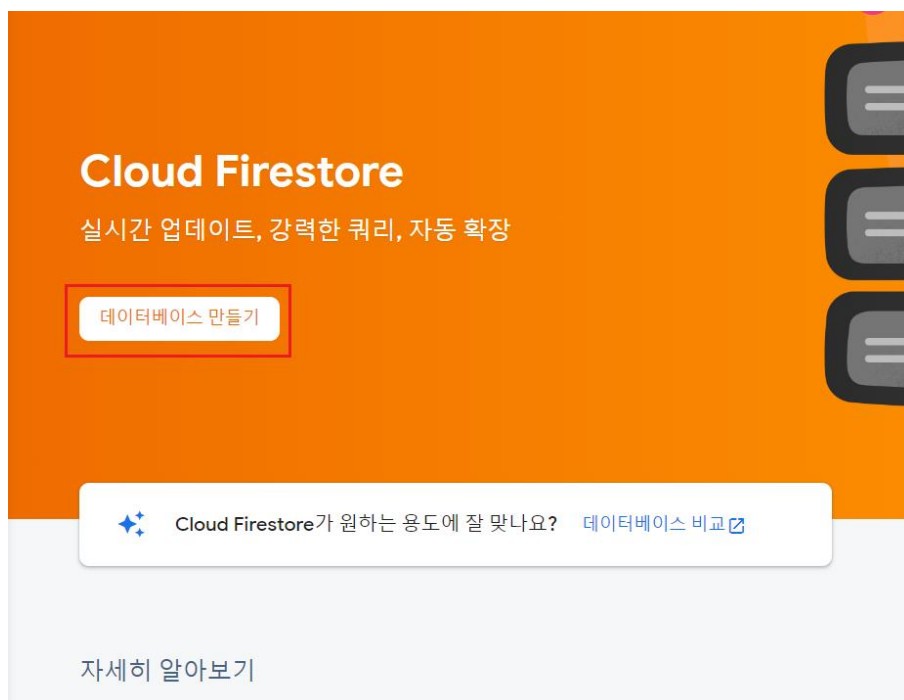
- 프로젝트 이름 설정(원하는 이름으로 설정)
- 애널리틱스 설정(비활성화)
- 프로젝트 만들기 버튼 클릭

## 2) Firestore 생성 (알림 기능)

- 프로젝트에 Cloud Firestore 클릭



- Database 만들기



- 보안 규칙 설정

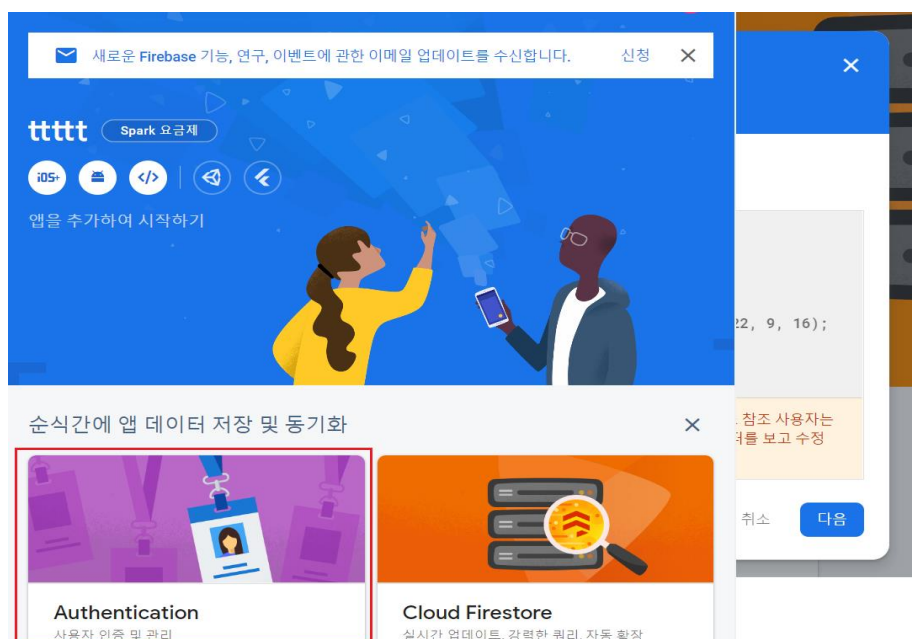
- 30 일 이후로도 활용할 경우, 보안 규칙을 변경해야 합니다.



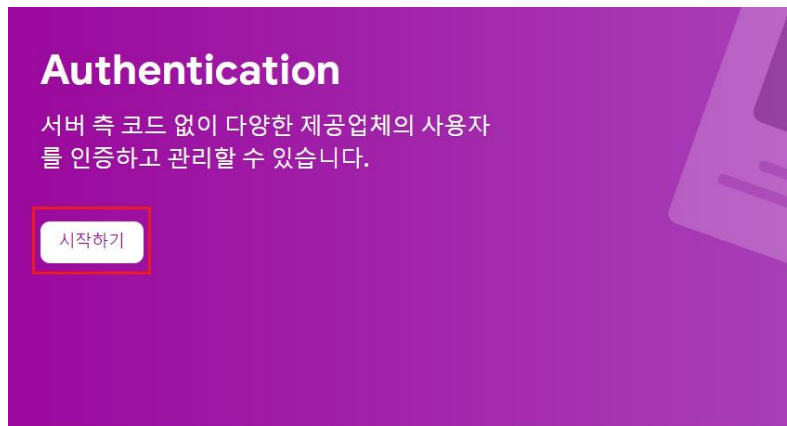
- 위치 설정

- 위치가 중요하진 않으나 빠른 속도를 위해 서버의 위치랑 가장 가까운 곳을 선택합니다.

3) Auth 생성 (소셜로그인)







**Authentication**

Users Sign-in method Templates Usage Settings

로그인 제공업체

첫 번째 로그인 방법을 추가하여 Firebase 인증 시작하기

기본 제공업체	추가 제공업체	커스텀 제공업체
이메일/비밀번호	<b>Google</b>	OpenID Connect
전화	Facebook	SAML
익명	Play 게임즈	
	게임 센터	
	...	

Google ☒ 사용 설정

연결된 Apple 및 웹 앱에서는 Google 로그인이 자동으로 구성됩니다. Android 앱에 대한 Google 로그인을 설정하려면 [프로젝트 설정](#)에서 각 앱에 대해 [SHA-1 디지털 지문](#)을 추가해야 합니다.

⚙️ 계속하려면 아래에서 [프로젝트 수준 설정](#)을 업데이트하세요.

프로젝트의 공개용 이름 ⓘ

project-1095021031872

프로젝트 지원 이메일 ⓘ

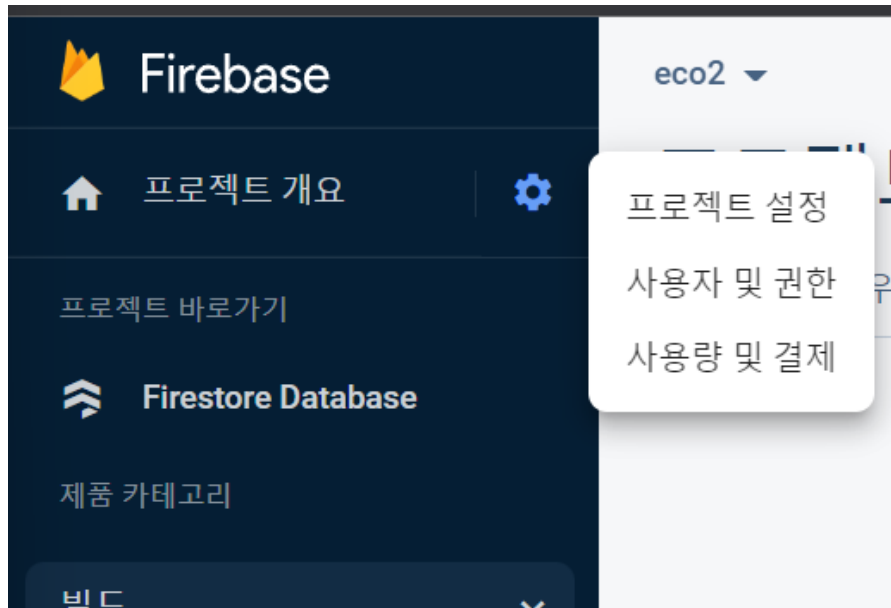
외부 프로젝트의 클라이언트 ID 허용 목록에 추가(선택사항) ⓘ

웹 SDK 구성 ⓘ

취소 **저장**

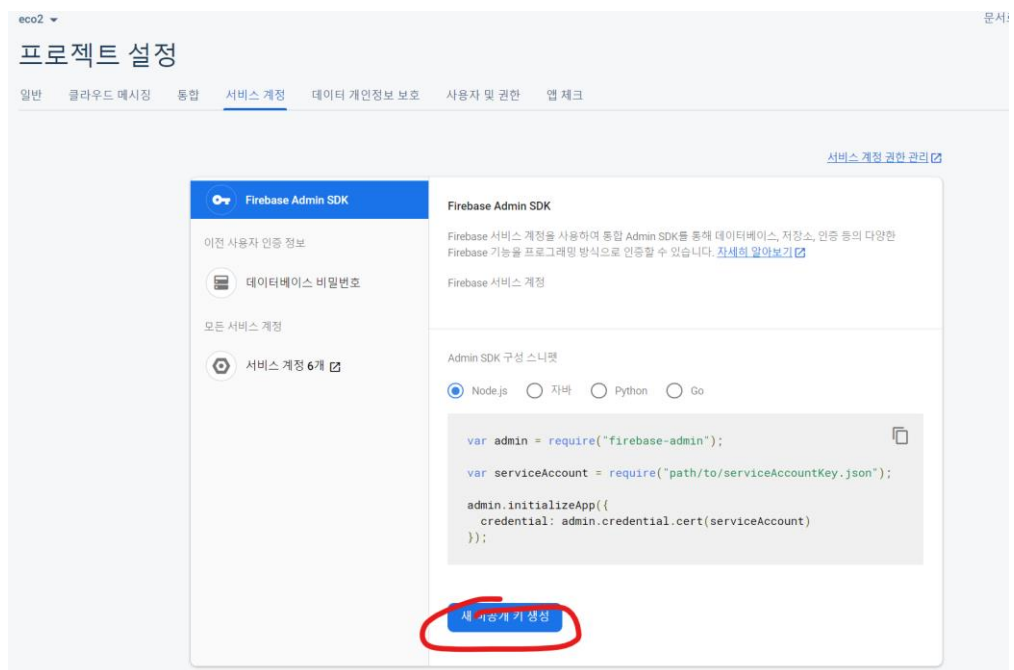
#### 4) Backend 비공개 키 저장

- 네비게이션 바에 **프로젝트 개요** 옆의 설정 아이콘 선택 후, **프로젝트 설정**을 클릭합니다.

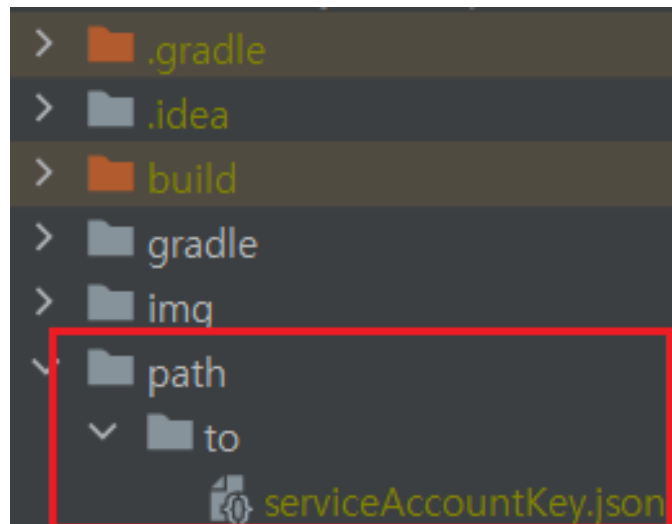


- 서비스 계정 > Firebase Admin SDK > 새 비공개 키 생성

■ json 파일을 저장합니다.

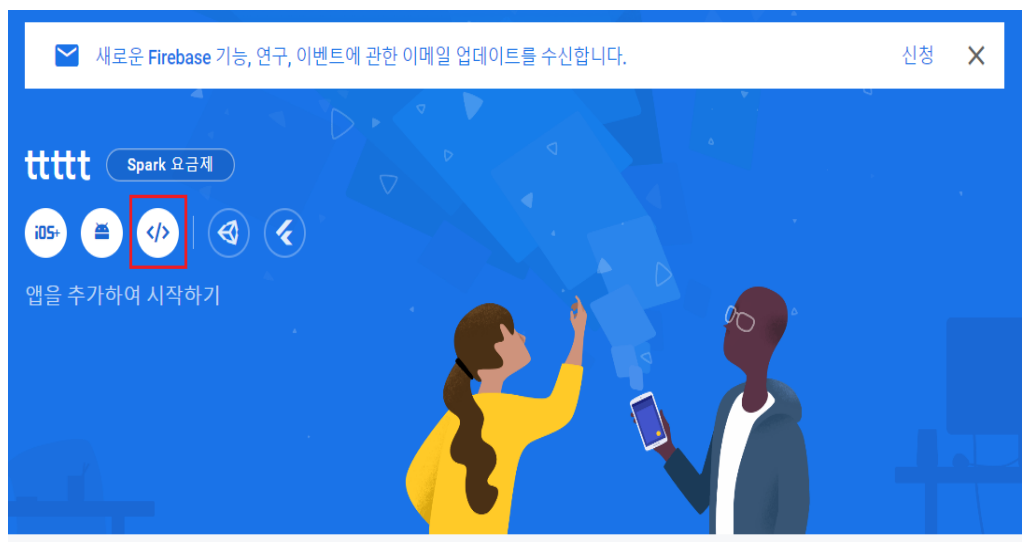


- 저장한 키 이름을 'serviceAccountKey.json'으로 변경한 후 프로젝트 폴더에 넣기
  - 위치: <프로젝트폴더>/path/to
  - 배포 시에는 백엔드 .jar 파일과 같은 디렉토리에서 /path/to 디렉토리를 생성합니다.



#### 5) Frontend API Key

- 앱 추가
  - 프로젝트 메인(개요)에서 웹 앱 추가 버튼 클릭하여 웹 플랫폼 앱을 생성합니다.



- Firebase 추가

- npm 사용 선택 후 콘솔로 이동 버튼 클릭합니다.

×

웹 앱에 Firebase 추가

1

앱 등록

앱 닉네임 ⓘ

ttttt

☐

또한 이 앱의 **Firebase 호스팅**을 설정하세요. [자세히 알아보기](#) ⓘ

나중에 호스팅을 설정할 수도 있습니다. 언제든지 시작하는 비용은 무료입니다.

앱 등록

2

Firebase SDK 추가

2

Firebase SDK 추가

☒ npm 사용 ⓘ

☐ <script> 태그 사용 ⓘ

이미 [npm](#) 및 모듈 번들러(예: [webpack](#) 또는 [Rollup](#))를 사용 중인 경우 다음 명령어를 실행하면 최신 SDK를 설치할 수 있습니다.

\$ npm install firebase

그런 다음 Firebase를 초기화하여 사용하려는 제품의 SDK를 사용하세요.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey:
  authDomain: "ttttt-74b5d.firebaseio.com",
  projectId: "ttttt-74b5d",
  storageBucket: "ttttt-74b5d.appspot.com",
  messagingSenderId: "1095021031872",
  appId:
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

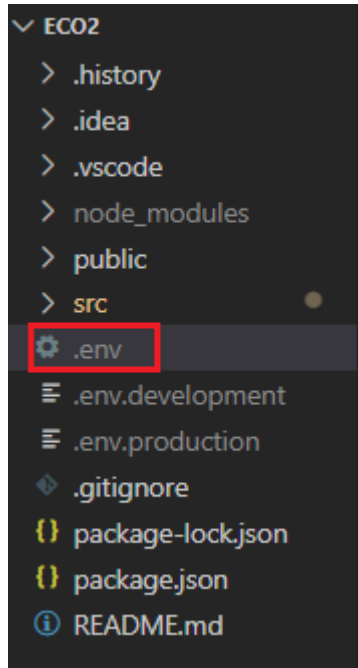
신청

×

28

6) API Key

- 프론트엔드 <프로젝트폴더>/env 파일에 작성
  - 각 항목에 맞게 작성합니다.



```
REACT_APP_FIREBASE_API_KEY=[REDACTED]  
REACT_APP_FIREBASE_AUTH_DOMAIN=[REDACTED]  
REACT_APP_FIREBASE_DB_URL=[REDACTED]  
REACT_APP_FIREBASE_PROJECT_ID=[REDACTED]  
REACT_APP_FIREBASE_STORAGE_BUCKET=[REDACTED]  
REACT_APP_FIREBASE_MESSAGING_SENDER_ID=[REDACTED]  
REACT_APP_FIREBASE_APP_ID=[REDACTED]
```

## 11. 공공데이터 API key 설정

가. 기상청 단기예보 조회 서비스

- 1) 미션 추천을 위해 현재 위치의 날씨 정보를 받아옵니다.
- 2) <https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15084084>
- 3) application-api-key.properties

`api.weather.key= <인코딩된 인증키>`