

외부 서비스

이메일 인증

1. [Google 계정 관리] → [보안] → [2단계 인증] 활성화
2. [Google 계정 관리] → [보안] → [앱 비밀번호] 생성 후 복사
3. mail.properties에 정보 기입

```
mail.smtp.auth=true
mail.smtp.starttls.required=true
mail.smtp.starttls.enable=true
mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
mail.smtp.socketFactory.fallback=false
mail.smtp.port=465
mail.smtp.socketFactory.port=465

#admin
AdminMail.id=pecommend@gmail.com
AdminMail.password=
```

4. [Gmail] → [빠른 설정] → [모든 설정 보기] → [전달 및 POP/IMAP]

설정

기본설정 라벨 받은편지함 계정 및 가져오기 필터 및 차단된 주소 전달 및 POP/IMAP 부가기능 채팅 및 Meet 고급 오프라인 테마

IMAP 액세스:

(IMAP를 사용하여 다른 클라이언트에서 Gmail에 액세스)
[자세히 알아보기](#)

상태: IMAP를 사용할 수 있습니다.

- ☒ IMAP 사용
☐ IMAP 사용 안함

IMAP에서 메일을 삭제된 것으로 표시하는 경우:

- ☒ 자동 삭제 사용 - 서버를 즉시 업데이트(기본값)
☐ 자동 삭제 사용 안함 - 클라이언트가 서버를 업데이트할 때까지 대기

메일이 삭제된 것으로 표시되고 마지막으로 표시된 IMAP 폴더에서 삭제된 경우:

- ☒ 메일 보관(기본값)
☐ 메일을 휴지통으로 이동
☐ 메일을 즉시 완전삭제

폴더 크기 제한

- ☒ IMAP 폴더에서 메일 수를 제한하지 않습니다(기본값).
☐ 이만큼의 메일만 포함하도록 IMAP 폴더를 제한합니다. 1,000 ▼

이메일 클라이언트 구성(예: Outlook, Thunderbird, iPhone)

[설정 방법](#)

변경사항 저장

취소

5. build.gradle 설정

```
implementation 'org.springframework.boot:spring-boot-starter-mail'
```

Oauth2 로그인

1. build.gradle 설정

```
implementation 'org.springframework.boot:spring-boot-starter-oauth2-client'
```

2. [Google Developer Console] → [프로젝트 생성]
3. [사용자 인증 정보] → [승인된 리디렉션 URI]에 `https://{도메인}/login/oauth2/code/google` 기입
4. 유저 정보를 받아올 `Oauth2Attribute` 작성
5. `Oauth2` 로그인 처리를 담당할 `CustomOauth2Service` 작성
6. `Oauth2` 로그인 후 처리를 담당할 `Oauth2SuccessHandler` 작성
7. `SecurityConfig`에 `Oauth2` 적용

```
@Override
protected void configure(HttpSecurity http) throws Exception{
    http
        //.... .and()
        .oauth2Login()
        .successHandler(successHandler)
        .userInfoEndpoint()
        .userService(customOAuth2Service); }
```

8. 프론트엔드(Login.js)에 `Oauth2` 호출 링크 생성

```
<a className="google-login-btn"
href="https://pecommend.com/oauth2/authorization/google?client_id={발급받은 client_id}" >
```

9. 리다이렉트 후 `oauth2` 로그인 처리하는 페이지 작성(`oauth.js`)

Nginx 설정

Nginx 설정

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;
    # Add index.php to the list if you are using PHP
    server_name i7e205.p.ssafy.io;
    return 301 https://i7e205.p.ssafy.io\$request-uri;
}

server {
    listen 443 ssl;
    listen [::]:443;
    server_name i7e205.p.ssafy.io;
    ssl_certificate /etc/letsencrypt/live/i7e205.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/i7e205.p.ssafy.io/privkey.pem;
    location / {
        root /home/ubuntu/app/peco/S07P12E205/backend/frontend/build;
        try_files $uri /index.html;
        index index.html;
    }
    location /api {
        proxy_pass http://localhost:8081;
        charset utf-8;
        proxy_redirect off;
```

```

proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-NginX-Proxy true;
proxy_set_header X-Forwarded-Proto $scheme;
}
location /oauth2
{
charset utf-8;
proxy_pass http://localhost:8081;
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Scheme $scheme;
proxy_set_header X-Auth-Request-Redirect $request_uri;
}
location /login/oauth2 {
    charset utf-8;
    proxy_pass http://localhost:8081;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Scheme $scheme;
    proxy_set_header X-Auth-Request-Redirect $request_uri;
}
}

```

deploy.sh 설정

```

#!/bin/bash
REPOSITORY=/home/ubuntu/app/peco
PROJECT_NAME=S07P12E205/backend
BUILD_NAME=perfumeService
cd $REPOSITORY/$PROJECT_NAME/

echo "> 프로젝트 Build 시작"
sudo ./gradlew build

echo "> 디렉토리 이동"
cd $REPOSITORY

echo "> Build파일 복사"
cp $REPOSITORY/$PROJECT_NAME/build/libs/*.jar $REPOSITORY/

```

```
echo "> 현재 구동중인 애플리케이션 pid 확인"
CURRENT_PID=$(pgrep -f ${BUILD_NAME}.*.jar)

echo "> 현재 구동중인 애플리케이션 pid: $CURRENT_PID"
if [ -z "$CURRENT_PID" ]; then
    echo "> 현재 구동중인 애플리케이션이 없으므로 종료하지 않습니다."
else
    echo "kill -9 $CURRENT_PID"
    kill -9 $CURRENT_PID
    sleep 5
fi

echo "새 어플리케이션 배포"
JAR_NAME=$(ls -tr $REPOSITORY/ | grep jar | tail -n 1)

echo "> JAR Name: $JAR_NAME"
nohup java -jar $REPOSITORY/$JAR_NAME 2>&1 &

echo "> 프론트 폴더로 이동"
cd $REPOSITORY/$PROJECT_NAME/frontend

echo "> npm 캐시삭제 및 인스톨 수행"
sudo npm cache clean --force
sudo npm install

echo "> 빌드 수행"
npm run build

echo "엔진엑스 재실행"
sudo service nginx restart
```