

포팅 매뉴얼

1. 개발 환경

1.1. Frontend

1.2. Backend

1.3. Server

1.4. Database

1.5. IDE

1.6. 형상/이슈관리

1.7. 기타 툴

2. 환경 변수 및 설정 파일 관리

2.1. Frontend - .env 파일

2.2. Backend - application.yml

3. EC2 세팅

3.1. EC2 Port

3.2. 방화벽(UFW) 설정

4. Redis 설정

4.1. Redis 실행

5. MySQL 설정

5.1 MySQL 실행

6. 빌드 및 배포 작업

6.1. Nginx 프록시 설정

6.2. 빌드 및 실행 방법

7. 외부 서비스

7.1. 소셜 로그인 - Kakao

7.1.1. 애플리케이션 생성

7.1.2. 애플리케이션 설정

7.1.3. 동의 항목 선택

7.1.4. Client-ID/Secret

7.1.5. 사이트 도메인 설정

7.2. 소셜 로그인 - Naver

7.2.1. 애플리케이션 생성

7.2.2. 애플리케이션 설정

7.2.3. Client-ID/Secret

7.3. SMTP - Google

1. 개발 환경

1.1. Frontend

기술스택	버전	설명
Visual Studio Code	1.97.2	개발 환경 (IDE)
React	18.3.1	UI 라이브러리
Redux	@reduxjs/toolkit (2.5.1) react-redux (9.2.0)	상태 관리
React Router DOM	6.28.2	라우팅
Axios	1.7.9	HTTP 클라이언트
JWT-Decode	4.0.0	JWT 토큰 디코딩
Chart.js	chart.js (4.4.7) react-chartjs-2 (5.3.0)	React용 차트 라이브러리
Vite	6.0.5	빌드 도구 및 로컬 서버

1.2. Backend

기술스택	버전	설명
Java	17	JDK 및 런타임 환경
Spring	6.1.2	코어 프레임워크
Spring Boot	3.4.2	Spring 애플리케이션 부트스트래핑
Spring Security	6.2.2	인증 및 인가
Spring Data JPA	3.4.2	JPA 사용을 위한 Spring 모듈
Spring Data Redis	3.4.2	Redis 데이터베이스 연동
Spring Session	3.4.2	Redis 세션 공유
Spring WebSocket	6.1.2	WebSocket 통신
Spring Validation	6.1.2	입력 값 검증
Spring Web	6.1.2	MVC 및 RESTful 웹 서비스
Spring WebFlux	6.1.2	비동기 및 논블로킹 웹프레임워크
SpringDoc-OpenAPI	2.8.0	Swagger UI 사용
MySQL	8.1.0	mysql-connector-j
Lombok	1.18.28	Getter/Setter 자동 생성
JJWT	0.12.6	JWT 인증 및 인가
Jackson-Databind	2.15.2	JSON 직렬화 및 역직렬화

기술스택	버전	설명
Mail Service	3.4.2	이메일 발송 기능
JUnit	5.10.0	테스트 프레임워크
Reactor-Test	3.5.11	WebFlux 테스트

1.3. Server

소프트웨어	버전	설명
Ubuntu	22.04.5 LTS x86_64	서버 운영체제
Nginx	1.18.0	Reverse Proxy 및 Static 파일 서빙
Docker	27.5.1	컨테이너 가상화 플랫폼
Jenkins	2.479.3	CI/CD 자동화 도구
Coturn	4.5.2	WebRTC TURN 서버

1.4. Database

- MySQL
- Redis

1.5. IDE

- Visual Studio Code (1.97.2)
- IntelliJ IDEA (2024.3.1.1)

1.6. 형상/이슈관리

- GitLab
- Jira

1.7. 기타 툴

- Postman
- K6

2. 환경 변수 및 설정 파일 관리

2.1. Frontend - .env 파일

민감 정보는 실제 값 대신 {...} 또는 예시 값으로 대체하여 기록

```
VITE_SPRING_API_URL=/spring
VITE_FLASK_API_URL=/flask
VITE_KAKAO_CLIENT_ID={KAKAO_CLIENT_ID}
VITE_KAKAO_REDIRECT_URI=https://i12e104.p.ssafy.io/auth/kakao/callback
VITE_USERNAME_URL=pororo
VITE_PASSWORD_URL={PASSWORD}
VITE_NAVER_REDIRECT_URI=https://i12e104.p.ssafy.io/auth/naver/callback
VITE_NAVER_CLIENT_ID={NAVER_CLIENT_ID}
VITE_NAVER_STATE_STRING={NAVER_STATE_STRING}
VITE_STATIC_API_URL=https://i12e104.p.ssafy.io/static
OPENAI_API_KEY={OPENAI_API_KEY}
```

주의사항:

- 이 파일은 배포 서버에 별도 저장하며, Git 등의 버전 관리 시스템에는 포함되지 않도록 .gitignore에 추가합니다.

2.2. Backend - application.yml

민감 정보는 실제 값 대신 {...} 또는 예시 값으로 대체하여 기록

```
server:
  port: 8081

password:
  reset:
    base-url: "https://i12e104.p.ssafy.io/reset-password"

spring:
  application:
    name: be

mail:
```

```
host: smtp.gmail.com
port: 587
username: {GMAIL_USERNAME}
password: {GMAIL_APP_PASSWORD}
properties:
  mail:
    smtp:
      auth: true
      starttls:
        enable: true
      connectiontimeout: 5000
      timeout: 5000
      writetimeout: 5000
```

```
session:
  redis:
    namespace: myapp:sessions
    flush-mode: on_save
    cleanup-cron: "0 * * * * *"
```

```
data:
  redis:
    host: redis # Docker Compose나 클러스터 환경의 별칭 사용
    port: 6379
    timeout: 6000ms
```

```
datasource:
  url: jdbc:mysql://mysql:3306/maraeba_db?serverTimezone=Asia/Seoul
  username: pororo
  password: {DB_PASSWORD}
  driver-class-name: com.mysql.cj.jdbc.Driver
```

```
jackson:
  time-zone: Asia/Seoul
  property-naming-strategy: SNAKE_CASE
```

```
mvc:
  format:
    date-time: yyyy-MM-dd'T'HH:mm:ss
```

```
jpa:
  hibernate:
    ddl-auto: update
  properties:
    hibernate:
      format_sql: true
  show-sql: true
  open-in-view: false
```

```
jwt:
  secret:
    key: {JWT_SECRET_KEY}
  expiration:
    access-token: 1800000
    refresh-token: 604800000
```

```
social:
  naver:
    client-id: {NAVER_CLIENT_ID}
    client-secret: {NAVER_CLIENT_SECRET}
    redirect-uri: "https://i12e104.p.ssafy.io/auth/naver/callback"
    token-uri: "https://nid.naver.com/oauth2.0/token"
    user-info-uri: "https://openapi.naver.com/v1/nid/me"
    state-string: {NAVER_STATE_STRING}
  kakao:
    client-id: {KAKAO_CLIENT_ID}
    client-secret: {KAKAO_CLIENT_SECRET}
    redirect-uri: "https://i12e104.p.ssafy.io/auth/kakao/callback"
    token-uri: "https://kauth.kakao.com/oauth/token"
    user-info-uri: "https://kapi.kakao.com/v2/user/me"
```

```
openai:
  api-key: {OPENAI_API_KEY}
```

```
url:
  ai: "http://flask:5000"
```

주의사항:

- 이 파일은 배포 서버에 별도 저장하며, Git 등의 버전 관리 시스템에는 포함되지 않도록 .gitignore에 추가합니다.

3. EC2 세팅

3.1. EC2 Port

Port 번호	내용
8080	Jenkins
8081	Spring
80	http
443	https
8478	coturn

3.2. 방화벽(UFW) 설정

22	ALLOW	Anywhere
80	ALLOW	Anywhere
8989	ALLOW	Anywhere
8080/tcp	ALLOW	Anywhere
443	ALLOW	Anywhere
3478/udp	ALLOW	Anywhere
3478/tcp	ALLOW	Anywhere
49152:65535/udp	ALLOW	Anywhere
5349/tcp	ALLOW	Anywhere
587	ALLOW	Anywhere
8081	ALLOW	Anywhere
80 (v6)	ALLOW	Anywhere (v6)
8080/tcp (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)
22 (v6)	ALLOW	Anywhere (v6)
8989 (v6)	ALLOW	Anywhere (v6)
3478/udp (v6)	ALLOW	Anywhere (v6)
3478/tcp (v6)	ALLOW	Anywhere (v6)

49152:65535/udp (v6)	ALLOW	Anywhere (v6)
5349/tcp (v6)	ALLOW	Anywhere (v6)
587 (v6)	ALLOW	Anywhere (v6)
8081 (v6)	ALLOW	Anywhere (v6)

4. Redis 설정

4.1. Redis 실행

```
docker run --name redis --network my_network -d redis
```

5. MySQL 설정

5.1 MySQL 실행

```
docker run -d --name mysql \
  --network my_network \
  -e MYSQL_ROOT_PASSWORD=pororo \
  -e MYSQL_DATABASE=maraeba_db \
  -e MYSQL_USER=pororo \
  -e MYSQL_PASSWORD={MYSQL_PASSWORD} \
  -e TZ=Asia/Seoul \
  -p 3306:3306 \
  -v mysql-data:/var/lib/mysql \
  mysql:8.0
```

6. 빌드 및 배포 작업

6.1. Nginx 프록시 설정

- Nginx는 다음과 같이 포트를 프록시합니다.
 - **80, 443 → 8082 (React 프론트 엔드)**

- 8081 → 8083 (Spring 백엔드 API)

6.2. 빌드 및 실행 방법

1. 도커 네트워크 생성

```
docker network create my_network
```

2. Flask 도커 이미지 빌드

```
cd ai
cp {환경변수 파일 경로} .env
docker build -t flask .
```

3. Flask 도커 이미지 실행

```
docker run -d --name flask --network my_network flask
```

4. Spring 도커 이미지 빌드

```
cd be
docker build -t spring .
```

5. Spring 도커 이미지 실행

```
docker run -d --name spring \\\n    -p 8083:8081 \\\n        --network my_network \\\n        -v {yml 파일 경로}:/app/config/application.yml \\\n        -e SPRING_CONFIG_LOCATION=file:/app/config/applicatio\nn.yml \\\n        spring
```

6. React 도커 이미지 빌드

```
cd fe
cp ${환경변수 파일 경로} .env
docker build -t react .
```

7. React 도커 이미지 실행

```
docker run -d --name react \\  
  --network my_network \\  
  --env-file {환경변수 파일 경로} \\  
  -p 8082:80 \\  
  react
```

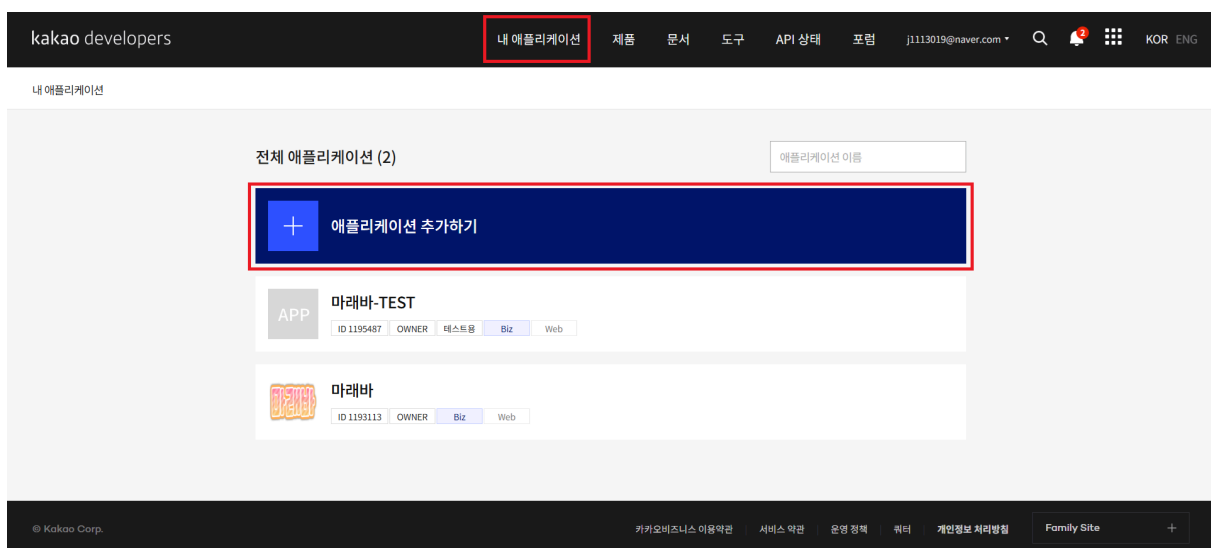
7. 외부 서비스

7.1. 소셜 로그인 - Kakao

Kakao Developers REST API (카카오 로그인)

- <https://developers.kakao.com/docs/latest/ko/kakaologin/rest-api>

7.1.1. 애플리케이션 생성



- 내 애플리케이션 > 애플리케이션 추가하기
- 적절한 내용 기입 후 만들어진 애플리케이션 클릭

7.1.2. 애플리케이션 설정

설정

항목	상태	상세 정보
카카오 로그인	ON	설정하기 가이드 확인하기 →
동의항목	설정함	설정하기 가이드 확인하기 →
간편가입	OFF	설정하기 가이드 확인하기 →
Redirect URI	설정함	설정하기 가이드 확인하기 →
연결 끊기	설정 안 함	설정하기 가이드 확인하기 →
계정 상태 변경 웹훅	설정 안 함	설정하기 가이드 확인하기 →

- 설정의 표시된 영역 설정하기

카카오 로그인
ON
동의 화면 미리보기

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.

OpenID Connect 활성화 설정

상태
ON

카카오 로그인의 확장 기능인 OpenID Connect를 활성화합니다.
이 설정을 활성화하면 카카오 로그인 시 사용자 인증 정보가 담긴 ID 토큰을 액세스 토큰과 함께 발급받을 수 있습니다.

Redirect URI

삭제
수정

Redirect URI	
https://[redacted]/auth/kakao/callback	
https://[redacted]/auth/kakao/callback	

- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)
- REST API로 개발하는 경우 필수로 설정해야 합니다.

- 활성화 설정에서 ON으로 변경
- Redirect URI에 주소 설정(Front/Back의 환경 변수에 존재)

7.1.3. 동의 항목 선택

동의항목

카카오 로그인으로 서비스를 시작할 때 동의받는 항목을 설정합니다. 미리보기를 통해 사용자에게 보여질 화면을 확인할 수 있습니다.
비즈니스 인증을 완료하면, 권한이 필요한 동의항목에 대한 심사 신청을 할 수 있습니다.

개인정보 동의항목 심사 신청

개인정보

항목 이름	ID	상태
닉네임	profile_nickname	● 필수 동의 설정
프로필 사진	profile_image	● 필수 동의 설정
카카오계정(이메일)	account_email	● 필수 동의 설정
이름	name	○ 권한 없음
성별	gender	○ 권한 없음
연령대	age_range	○ 권한 없음
생일	birthday	○ 권한 없음

- 닉네임, 프로필 사진, 카카오 계정(이메일)을 필수 동의 항목으로 설정

7.1.4. Client-ID/Secret

앱 설정

대시보드

일반

비즈니스

앱 키

플랫폼

앱 권한 신청

팀 관리

유료 설정

비즈윌렛 관리

유료 사용량

제품 설정

카카오 로그인

동의항목

마래바

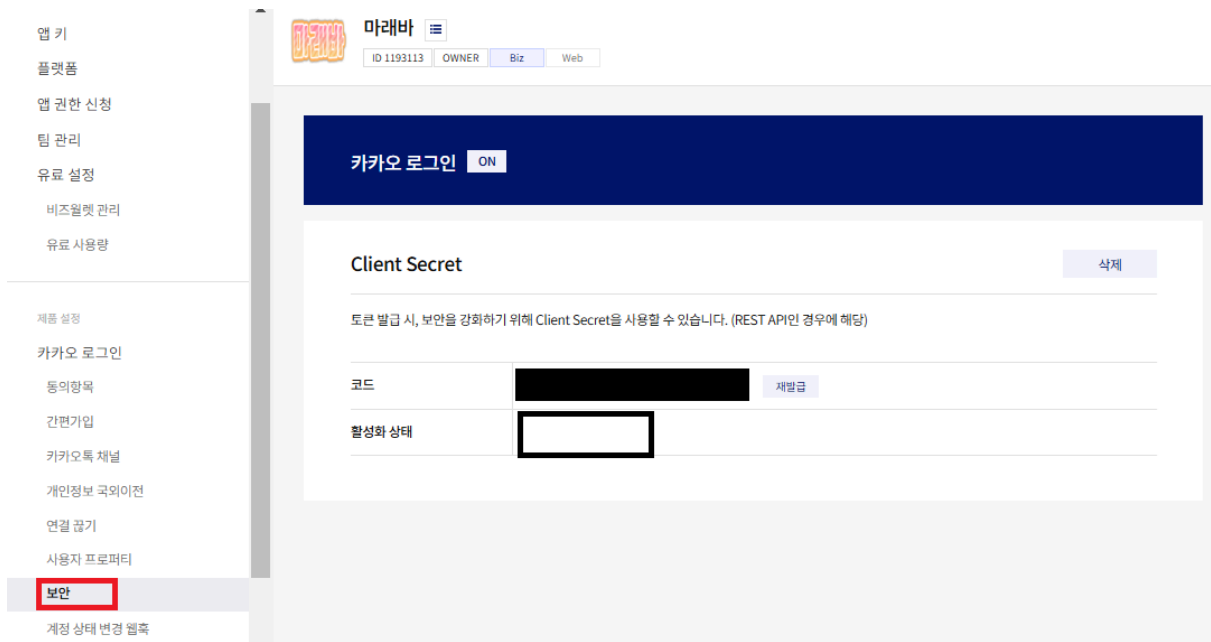
ID 1193113 OWNER Biz Web

앱 키

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

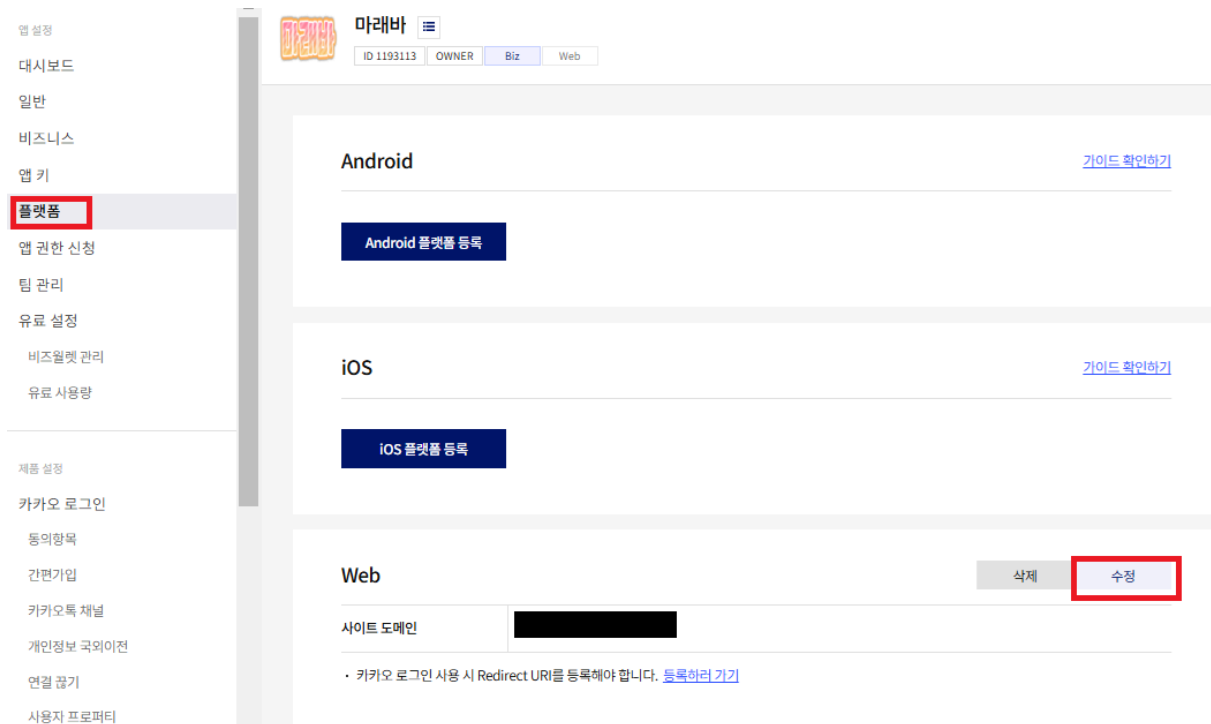
- 네이티브 앱 키: Android, iOS SDK에서 API를 호출할 때 사용합니다.
- JavaScript 키: JavaScript SDK에서 API를 호출할 때 사용합니다.
- REST API 키: REST API를 호출할 때 사용합니다.
- Admin 키: 모든 권한을 갖고 있는 키입니다. 노출이 되지 않도록 주의가 필요합니다.

- 앱 키 탭에서 REST API 키를 Client-ID로 환경 변수에 기입



- 보안 탭에서 Client Secret을 활성화 한 뒤, 코드를 Backend 환경 변수에 기입

7.1.5. 사이트 도메인 설정



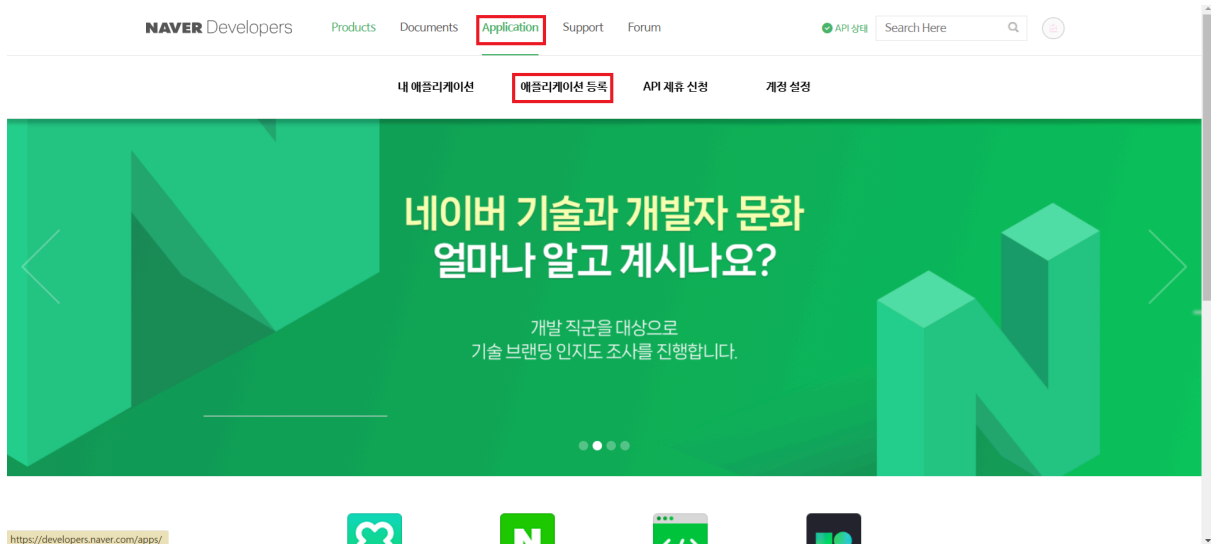
- 운영하는 도메인 주소를 설정

7.2. 소셜 로그인 - Naver

Naver Developers Open API (네이버 로그인)

- <https://developers.naver.com/main/>

7.2.1. 애플리케이션 생성



- Application > 애플리케이션 등록



- 사용하고자 하는 이름 기입 후 사용 API에 네이버 로그인 추가

7.2.2. 애플리케이션 설정

사용 API

선택하세요.



네이버 로그인

제공 정보 선택(이용자 식별자는 기본 정보로 제공) [?]

필수 항목은 개인정보보호법 제3조 제1항, 제16조 제1항 등에 따라 서비스 제공을 위해 필요한 최소한의 개인정보만을 선택해야 합니다.

권한	필수	추가
회원이름	<input checked="" type="checkbox"/>	<input type="checkbox"/>
연락처 이메일 주소	<input checked="" type="checkbox"/>	<input type="checkbox"/>
별명	<input type="checkbox"/>	<input type="checkbox"/>
프로필 사진	<input type="checkbox"/>	<input type="checkbox"/>
성별	<input type="checkbox"/>	<input type="checkbox"/>
생일	<input type="checkbox"/>	<input type="checkbox"/>
연령대	<input type="checkbox"/>	<input type="checkbox"/>
출생연도	<input type="checkbox"/>	<input type="checkbox"/>
휴대전화번호	<input type="checkbox"/>	<input type="checkbox"/>

- 제공 정보 선택에서 회원이름, 연락처 이메일 주소를 필수로 등록

로그인 오픈 API
서비스 환경 [?]

환경 추가



PC 웹



서비스 URL

[Redacted]

서비스 URL예시: (O) http://naver.com (X) http://www.naver.com

서비스 URL값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다.

불법/음란성 사이트 등 이용약관에 위배되는 사이트의 경우, 이용이 제한될 수 있습니다.

서비스하려는 사이트 URL과 동일한 사이트 URL로 해주셔야 **네이버 로그인 뱃지**가 노출됩니다.

네이버 로그인

Callback URL (최대 5개)

https://[Redacted]/auth/naver/callback



https://[Redacted]/auth/naver/callback



텍스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다.

Callback URL은 네이버 로그인 후 이동할 페이지 URL입니다. Callback URL값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다.

입력한 주소와 다른 Callback URL로 리다이렉트 될 경우, 이용이 제한될 수 있습니다.

- 서비스 URL과 Callback URL 등록 (Callback URL은 환경변수의 Redirect-URI에 똑 같이 기입)

7.2.3. Client-ID/Secret

내 애플리케이션

마래바

개요 API 설정 멤버관리 로그인 통계 API 통계 Playground (Beta)

애플리케이션 등록
API 제휴 신청
계정 설정

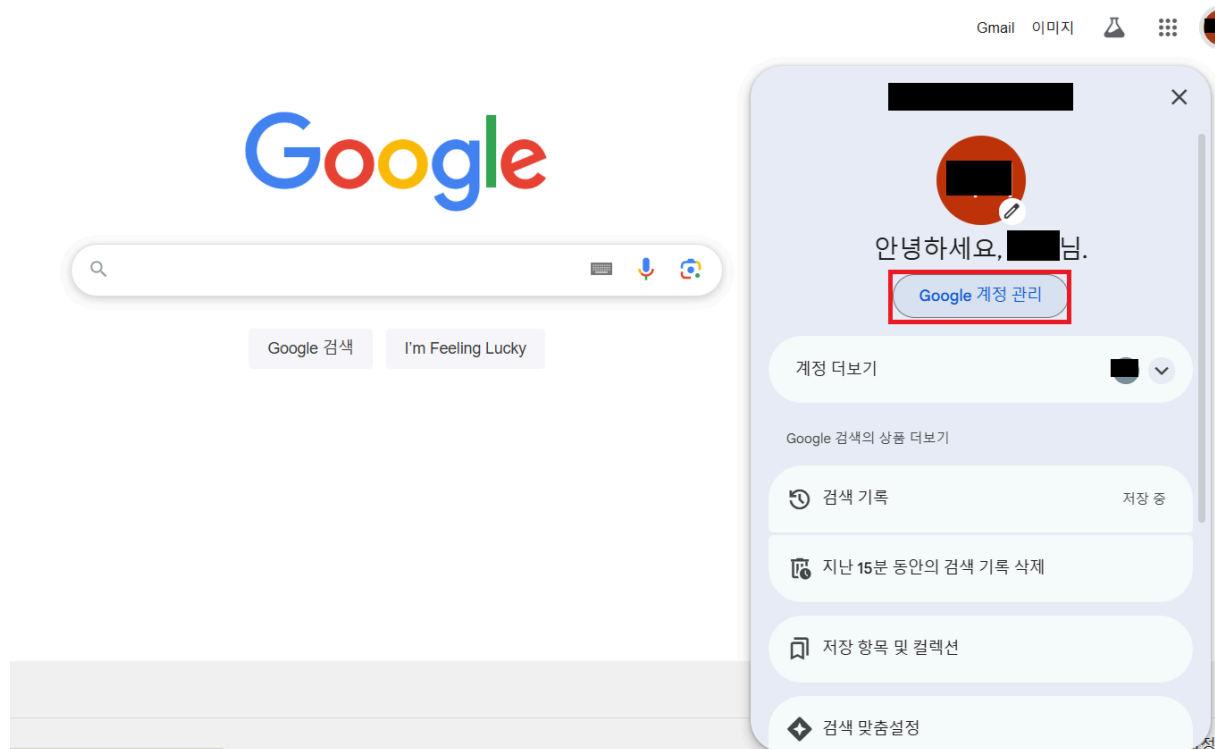
애플리케이션 정보

Client ID	[Redacted]
Client Secret	[Redacted] 보기

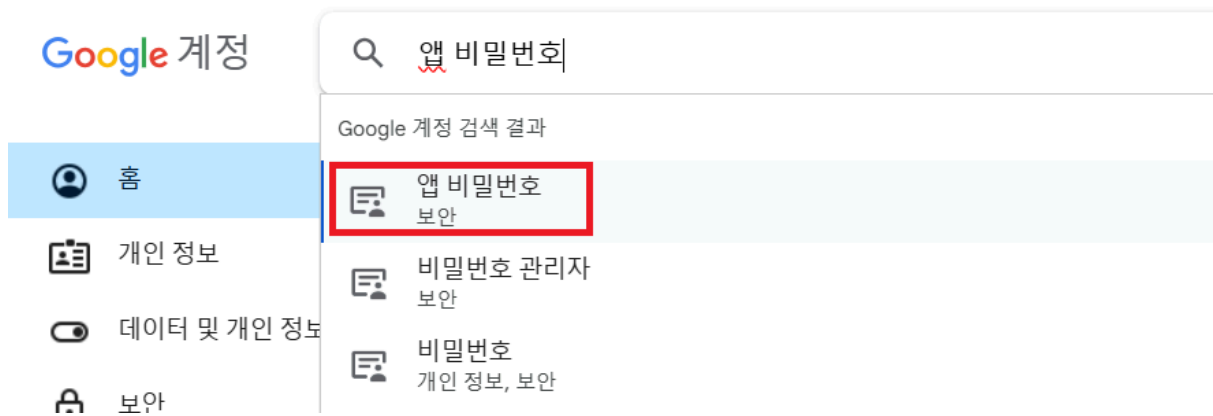
- 내 애플리케이션 탭에서 생성된 Client-ID와 Client-Secret 코드 확인 가능

7.3. SMTP - Google

- 비밀번호 재설정 링크를 이메일로 보내는 기능에서 사용



- 이메일을 전송할 구글 계정으로 로그인 하고 Google 계정 관리 접속



- 검색창에 앱 비밀번호 입력 후 접속

앱 전용 비밀번호를 새로 만들려면 아래에 앱 이름을 입력하세요...

- 앱 이름 기입 후 생성된 앱 비밀번호를 Back-end의 application.yml에 기입