



포팅 매뉴얼 : 개미는 오늘도

태그

개발

I. 개발 환경

1. 프로젝트 기술 스택 및 버전 정보
2. 환경 변수 설정
 - Front-end
 - Back-end
 - Data(FastAPI)

II. 프로젝트 실행 가이드

1. Front-end 실행 가이드
2. Back-end 실행 가이드

III. 배포 환경

1. 서버 구성(AWS)
2. 도커 환경 설정
 - 도커 설치
 - 도커 컴포즈 설치
3. 젠킨스
 - 젠킨스 설치 및 도커 컴포즈 설정
 - 젠킨스 도커 이미지 빌드
 - 깃랩 설정
 - 아이템 생성
 - 도커 컴포즈 스프링 등록
 - jenkins 컨테이너 키 등록
 - execute shell 명령어 등록
 - 도메인 설정 및 SSL 인증

nginx 설정

IV. 계정 정보 및 DB Dump 파일

1. 계정 정보
 - 개미는 오늘도 서비스 접속 계정 정보
 - Jenkins 접속 계정 정보
 - 로컬 MySQL 접속 계정 정보
2. DB Dump 파일
 - EC2 MySQL 접속 계정 정보

V. 외부 서비스

1. Kakao OAuth2.0(카카오 로그인)
2. 전자공시 OPENDART API

I. 개발 환경

1. 프로젝트 기술 스택 및 버전 정보



프로젝트에서 사용한 주요 기술 스택 및 프로그램 버전 정보입니다.

Front-end

- React : 18.2.0
- Node.js : 18.16.1
- TypeScript : 5.0.2
- Vite : 4.4.5
- yarn : 1.22.19

Back-end

- Java : 17
- Spring Boot : 2.7.15
- Gradle : 8.2.1
- ORM : JPA(Hibernate)
- Spring Security

- Recoil : 0.7.7
- React-query : 3.39.3
- Recharts : 2.8.0

- Swagger : 2.9.2
- java-jwt : 4.2.1
- jsoup : 1.14.3

Data

- Python
- BeautifulSoup4
- financeDataReader
- FastAPI
- KoNLPy

IDE

- Visual Studio Code : 1.18.1
- IntelliJ : 2023.1.3

UI/UX

- Figma

Database

- MySQL : 8.1.0

Server

- Nginx : 1.18.0
- NginxProxyManager
- AWS

CI/CD

- Docker : 24.0.6
- Jenkins : 2.423
- Ubuntu : Ubuntu 20.04 LTS

이슈 관리

- Jira

형상 관리

- Gitlab

Communication

- Notion
- MatterMost
- Discord

2. 환경 변수 설정

Front-end

- 환경 변수 설정 위치

```
antoday_FE
└── .env
```

- .env 파일 내용

```
# 배포 버전
VITE_BACK_API_URL = "https://antoday.site:8080"
VITE_DATA_API_URL = "https://antoday.site/info"
VITE_FRONT_URL = "https://antoday.site:5173"
VITE_CLIENT_ID = "ef1dbd44e7ef6634e3fec53570d8fd11"
```

```
# local 버전
VITE_BACK_API_URL = "http://localhost:8080"
VITE_DATA_API_URL = "https://antoday.site/info"
VITE_FRONT_URL = "http://localhost:5173"
VITE_CLIENT_ID = "ef1dbd44e7ef6634e3fec53570d8fd11"
```

Back-end

- 환경 변수 설정 위치

```
antoday
├─ src
│   └─ main
│       └─ resources
│           ├── application.yml
│           ├── application-dev.yml
│           └─ application-prod.yml
```

- `application.yml` 파일 내용

```
spring:
  profiles:
    active: dev
  jpa:
    database: mysql
    show-sql: true
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        show_sql: true
        format_sql: true
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher

logging:
  level:
    org:
      hibernate:
        type:
          descriptor:
            sql: trace

jwt:
  secretKey: c2lsdmVybmluZS10ZWNoLXNwcm1uZy1ib290LWp3dC10dXRvcmlhbmC1zZWNyZXQtc2lsdmVybmluZS10ZWNoLXNwcm1uZy1ib290LWp3dC10dXRvcmlhbmC1zZ

data:
  apiKey: e361bdf14cb2ff30cbd64501d5b51b502734199d
  path: C:/Users/SSAFY/Desktop/
```

- `application-dev.yml` 파일 내용 (Local 환경용)

```
server:
  address: localhost
  port: 8080
  servlet:
    context-path: /api
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/antoday?createDatabaseIfNotExist=true&useUnicode=true&characterEncoding=UTF-8&characterSetResults=
    username: root
    password: root
kakao:
  client_id: ef1dbd44e7ef6634e3fec53570d8fd11
  redirect_uri: http://localhost:5173/oauth/callback/kakao
```

- `application-prod.yml` 파일 내용 (배포 환경용)

```
server:
  address: 0.0.0.0
  port: 8080
  servlet:
    context-path: /api
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://3.35.54.104:3306/antoday?createDatabaseIfNotExist=true&useUnicode=true&characterEncoding=UTF-8&characterSetResult=UTF-8
    username: root
    password: omfg
  kakao:
    client_id: ef1dbd44e7ef6634e3fec53570d8fd11
    redirect_uri: https://antoday.site/oauth/callback/kakao
```

Data(FastAPI)

- 환경변수, 도커파일 위치

```
antoday_data
├── app
│   └── secret.json
├── requirements.txt
└── Dockerfile
```

- `secrets.json` 파일 내용

```
{
  "DB" : {
    "user" : "root",
    "password" : "omfg",
    "host" : "3.35.54.104",
    "port" : 3306,
    "database" : "antoday"
  },
  "Dart" : {
    "api_key" : "ac4b2a472edbe5d41500b370c8f35b4c9fc6677c"
  },
  "OPENAI" : {
    "api_key" : "sk-BDTg9YxJDSC80ueG9EvnT3B1bkFJjyYVybX6DgxIx8gJSpq"
  }
}
```

- `Dockerfile` 내용

```
FROM ubuntu:latest

ENV LANG=C.UTF-8
ENV DEBIAN_FRONTEND=noninteractive

# 기본 시스템 패키지 설치
RUN apt-get update && \
  apt-get install -y --no-install-recommends tzdata g++ curl

# OpenJDK 8 설치
RUN apt-get install -y openjdk-8-jdk
ENV JAVA_HOME="/usr/lib/jvm/java-1.8-openjdk"

# Python 3.11 설치
RUN apt-get install -y python3.11 python3.11-dev python3.11-distutils

# Python 3.11을 기본으로 설정
RUN update-alternatives --install /usr/bin/python python /usr/bin/python3.11 1
RUN apt-get install -y pkg-config
RUN apt-get install -y libmysqlclient-dev

# pip 설치 및 설정
RUN curl https://bootstrap.pypa.io/get-pip.py | python -
RUN apt-get clean && \
  rm -rf /var/lib/apt/lists/*
```

```

WORKDIR /src
COPY . .

# Python 패키지 설치
RUN pip install -r requirements.txt

EXPOSE 8000

CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]

```

II. 프로젝트 실행 가이드

1. Front-end 실행 가이드

▼ 사전 필수 설정 사항

- (1) **Visual Studio Code** 1.18.1 설치 - [설치 링크](#)
- (2) **Node.js** 18.16.1 설치 - [설치 링크](#)
- (3) **yarn** 1.22.19 설치

```
> npm install -g yarn
```

(4) Project Git Clone

```
> git clone https://lab.ssafy.com/s09-fintech-finance-sub2/S09P22E107.git
```

- 1) **antoday_FE** 프로젝트 열기
- 2) **yarn install** 실행
- 3) **환경 변수 설정**
- 4) **yarn start** 실행

2. Back-end 실행 가이드

▼ 사전 필수 설정 사항

- (1) **IntelliJ** 2023.1.3 설치 - [설치 링크](#)
- (2) **Java 17** 설치 및 환경 변수 설정 - [설치 링크](#)
- (3) **MySQL** 8.0.34 설치 - [설치 링크](#)
- (4) **Project Git Clone**

```
> git clone https://lab.ssafy.com/s09-fintech-finance-sub2/S09P22E107.git
```

- 1) **antoday** 프로젝트 열기
- 2) **JDK 17** 버전 확인
- 3) **환경 변수 설정**
- 4) **src/main/java/ AntodayApplication** class 실행

III. 배포 환경



서버 구성에 사용된 코드입니다.

1. 서버 구성(AWS)

[참고 포팅 매뉴얼](#)

2. 도커 환경 설정

도커 설치

```
$ sudo apt-get update
$ sudo apt-get -y upgrade
$ curl -fsSL https://get.docker.com/ | sudo sh

...

$ docker --version
# 권한설정
$ sudo usermod -aG docker $USER
$ sudo service docker restart

# 아래는 재로그인(연결을 종료했다 다시 연결해도 된다)
$ sudo su
$ sudo su ubuntu

$ docker ps
```

도커 컴포즈 설치

```
$ sudo curl -L https://github.com/docker/compose/releases/download/1.26.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose

$ sudo chmod +x /usr/local/bin/docker-compose

$ docker-compose --version
```

3. 젠킨스

젠킨스 설치 및 도커 컴포즈 설정

```
$ mkdir compose && cd compose
$ mkdir jenkins-dockerfile && cd jenkins-dockerfile
$ vim Dockerfile

...
FROM jenkins/jenkins:lts

USER root
RUN apt-get update &&\
    apt-get upgrade -y &&\
    apt-get install -y openssh-client
...

$ cd ..
$ vim docker-compose.yml

...
version: "3"
services:
  jenkins:
    container_name: jenkins-compose
    build:
      context: jenkins-dockerfile
      dockerfile: Dockerfile
    user: root
    ports:
      - 8080:8080
      - 8888:50000
    volumes:
      - /home/ubuntu/compose/jenkins:/var/jenkins_home
      - /home/ubuntu/compose/.ssh:/root/.ssh
```

```
...  
$ mkdir jenkins  
$ mkdir .ssh
```

젠킨스 도커 이미지 빌드

```
docker-compose up --build -d  
$ docker logs jenkins-compose # 비밀번호 출력
```

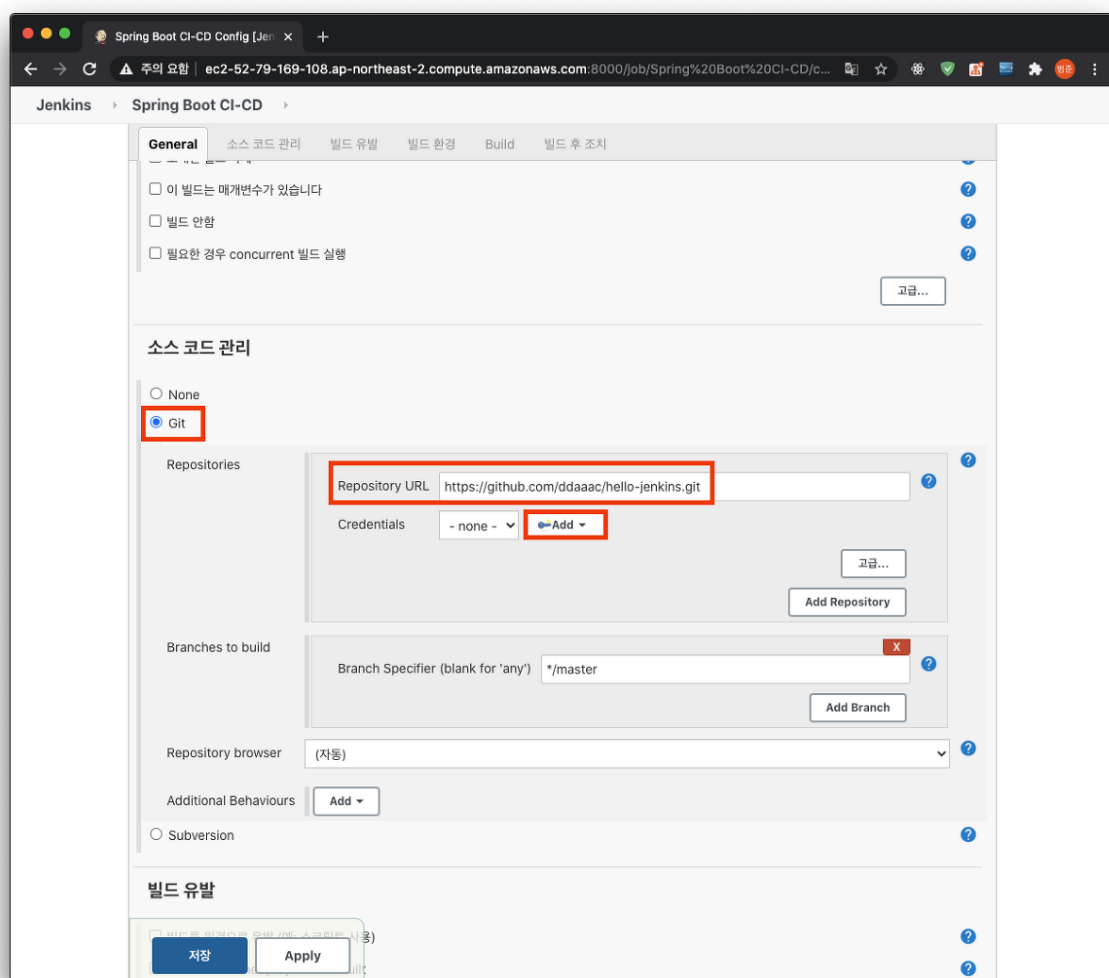
깃랩 설정

젠킨스에 gitlab 플러그인 설치

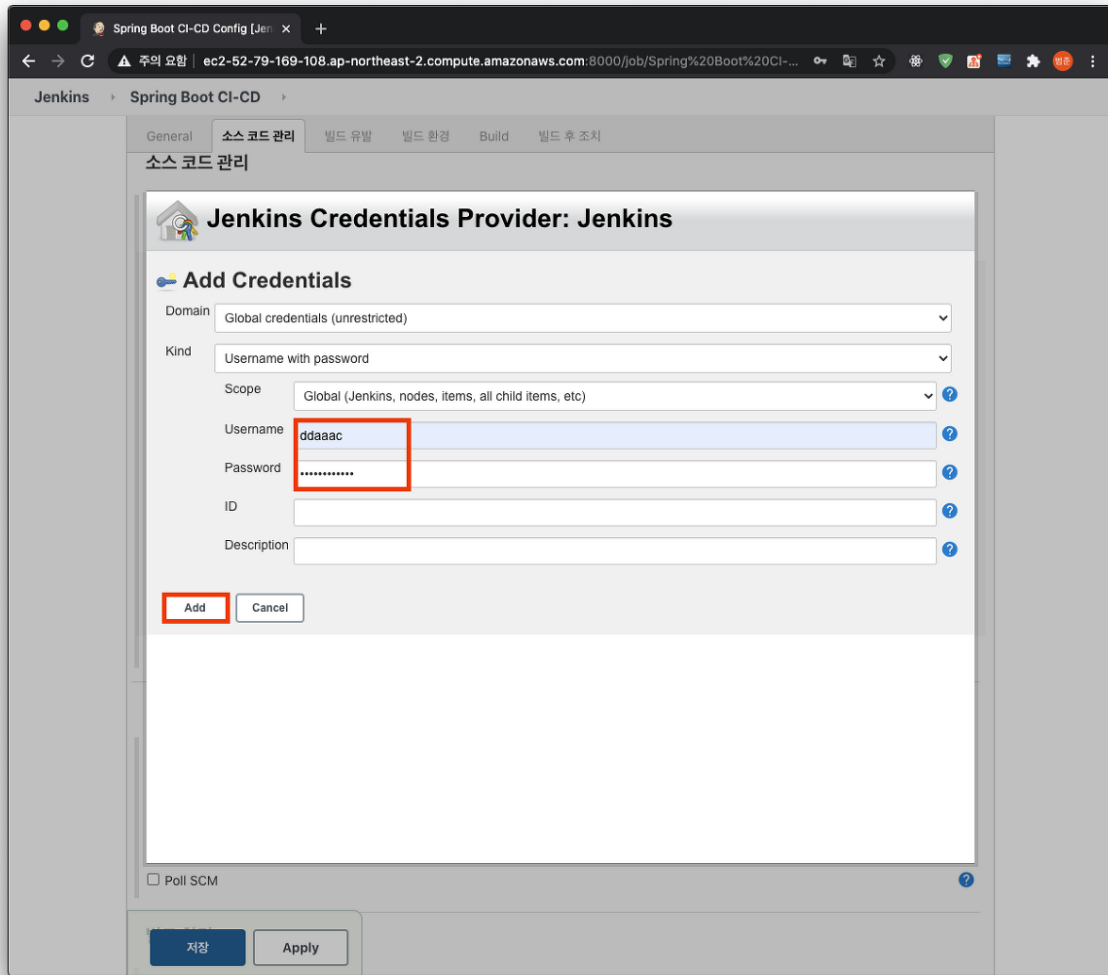
gitlab credential 추가

아이템 생성

프리스타일 생성



깃랩 저장소 등록



계정 등록

깃랩 토큰 등록

깃랩 웹훅 등록

도커 컴포즈 스프링 등록

```
version: "3"
services:
  jenkins:
    container_name: jenkins-compose
    build:
      context: jenkins-dockerfile
      dockerfile: Dockerfile
    user: root
    ports:
      - 8000:8080
      - 8888:50000
    volumes:
      - /home/ubuntu/compose/jenkins:/var/jenkins_home
      - /home/ubuntu/compose/.ssh:/root/.ssh
  spring:
    container_name: spring-compose
    build:
      context: spring-dockerfile
      dockerfile: Dockerfile
    ports:
      - 8080:8080
    volumes:
      - /home/ubuntu/compose/jenkins/workspace/antoday_back/antoday/build/libs:/deploy
```



```
FROM openjdk:17-jdk
ENTRYPOINT java -jar /deploy/antoday-0.0.1-SNAPSHOT.jar
ENV SPRING_PROFILES_ACTIVE=prod // 환경변수 등록
EXPOSE 8080
```

jenkins 컨테이너 키 등록

```
$ ssh-keygen -t rsa

$ cat /root/.ssh/id_rsa.pub

$ exit
```

```
$ vim ~/.ssh/authorized_keys // 이후 위에서 얻은 키 붙여넣기

$ docker exec -it jenkins-compose bash

$ ssh ubuntu@우분투주소 // 접속 확인
```

execute shell 명령어 등록

```
GRADLE_BUILD_DIR=/var/jenkins_home/workspace/antoday_back/antoday
cd $GRADLE_BUILD_DIR // 프로젝트 디렉토리로 이동
chmod +x ./gradlew // 권한부여
./gradlew clean bootjar -Pspring.profiles.active=prod // 빌드
ssh -t -t ubuntu@j9e107.p.ssafy.io <<EOF // ec2서버 접속 후 명령어 실행
cd /home/ubuntu/compose
docker-compose build --no-cache spring // 캐시 없이 새로 spring만 빌드
docker-compose up -d // 컴포즈 실행
echo "y" | docker container prune // 빈 컨테이너 삭제
echo "y" | docker image prune // 빈 이미지 삭제
exit
EOF
```

도메인 설정 및 SSL 인증

SSLForFree, CertBot 등의 서비스를 사용해 도메인에 SSL 인증

nginx 설정

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name antoday.site www.antoday.site;

    ssl_certificate /home/ubuntu/certificate.pem;
    ssl_certificate_key /home/ubuntu/private.key;
    ssl_prefer_server_ciphers on;

    location / {
        proxy_pass http://localhost:5173;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
    }

    location /api {
        proxy_pass http://localhost:8080;
```

```

        proxy_set_header X-Real_IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
    }

    location /info {
        proxy_pass http://localhost:8000;
        proxy_set_header X-Real_IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
    }
}

server {
    listen 80;
    server_name antoday.site j9e107.p.ssafy.io;
    return 301 https://antoday.site$request_uri;
}

```

IV. 계정 정보 및 DB Dump 파일

1. 계정 정보



프로젝트 관련 DB, Jenkins, 서비스 접속 계정 정보입니다.

개미는 오늘도 서비스 접속 계정 정보

- ID : antoday107@gmail.com
- Password : antoday123
- Swagger
 - 접속 URL : <https://antoday.site/api/swagger-ui.html>
 - 사용자 정보 필수인 API 테스트를 위한 토큰
 - Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJzdWliOiLQsJzrr7jripQg7Jik64qY64-ESldUVG9rZW4iLCJzb2NpYWxJZCI6MzAxNTcwMDE2MSwiZXhwljoxNjk3MDE2MTE3fQ.CnOYgtqwzGJ6g7dEsDdEmzklRqBNjKKURNA

Jenkins 접속 계정 정보

- Jenkins URL : <http://j9e107.p.ssafy.io:9000/login?from=%2F>
- ID : omfg
- Password: omfg

로컬 MySQL 접속 계정 정보

- ID : root
- Password : root

EC2 MySQL 접속 계정 정보

- ID : root
- Password : omfg

2. DB Dump 파일

개미는 오늘도 Dump_Data.zip

V. 외부 서비스

1. Kakao OAuth2.0(카카오 로그인)

1) <https://developers.kakao.com/console/app> 접속 후 로그인

2) 내 애플리케이션 > 애플리케이션 추가하기

kakao developers

내 애플리케이션

제품

더 보기

애플리케이션 추가하기

내 애플리케이션

전체 애플리케이션 (1)

애플리케이션 추가하기

앱 아이콘

이미지 업로드

파일 선택

JPG, GIF, PNG

권장 사이즈 128px, 최대 250KB

앱 이름

내 애플리케이션 이름

사업자명

사업자 정보와 동일한 이름

· 입력된 정보는 사용자가 카카오 로그인을 할 때 표시됩니다.
· 정보가 정확하지 않은 경우 서비스 이용이 제한될 수 있습니다.

서비스 이용이 제한되는 카테고리, 금지된 내용, 금지된 행동 관련 운영정책을 위반하지 않는 앱입니다.

취소

저장

- 앱 이름 : 서비스 이름
- 사업자명 : 팀명(혹은 임의 설정)
- ☒ 서비스 이용이 제한되는 카테고리, 금지된 내용, 금지된 행동 관련 운영 정책을 위반하지 않는 앱입니다.” 체크하기

3) 내 애플리케이션 > 앱 설정 > 플랫폼 > Web

내 애플리케이션 > 앱 설정 > 플랫폼

앱 설정

요약 정보

일반

비즈니스

앱 키

플랫폼

팀 관리

제품 설정

카카오 로그인

동의항목

간편가입

카카오톡 채널

개인정보 국외이전

Android 플랫폼 등록

iOS

iOS 플랫폼 등록

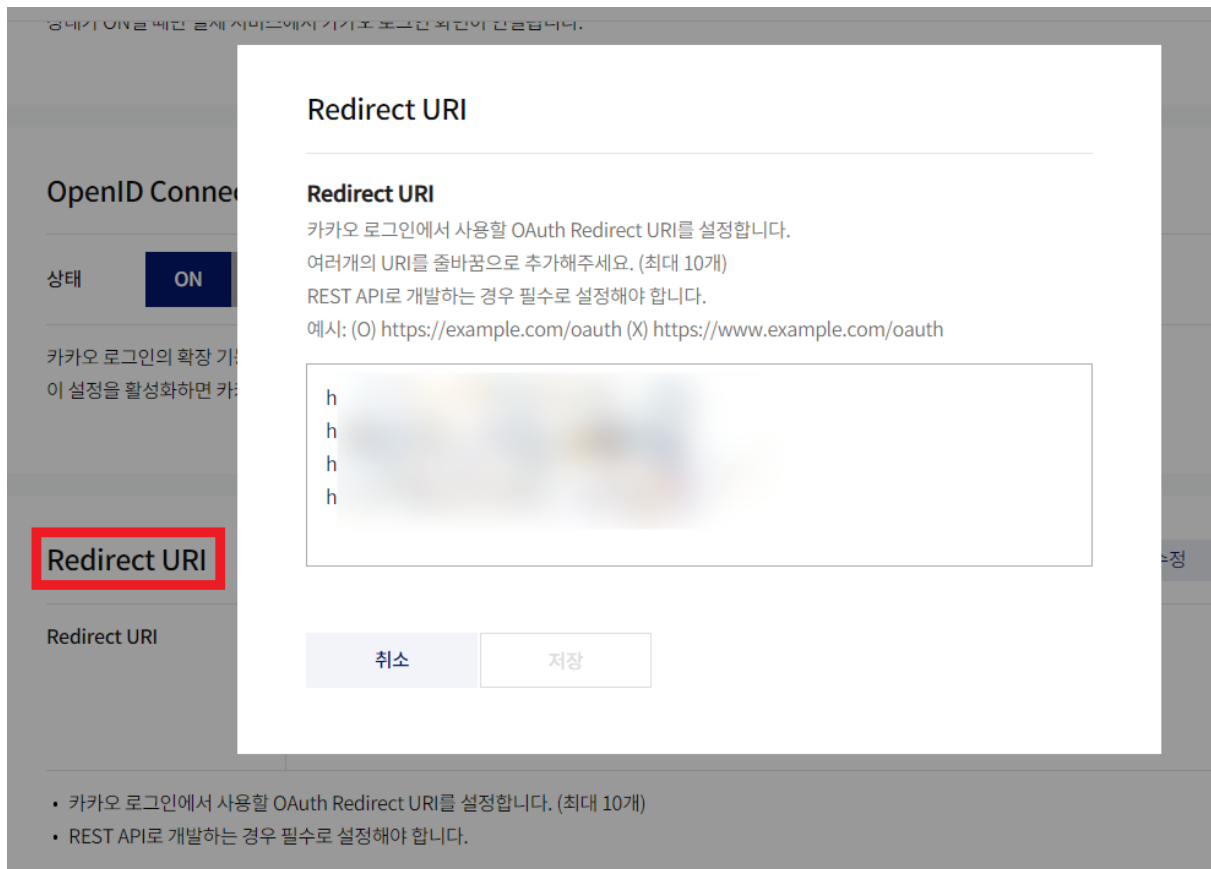
Web

사이트 도메인

htt
htt
htt• 카카오 로그인 사용 시 Redirect URI를 등록해야 합니다. [등록하러 가기](#)

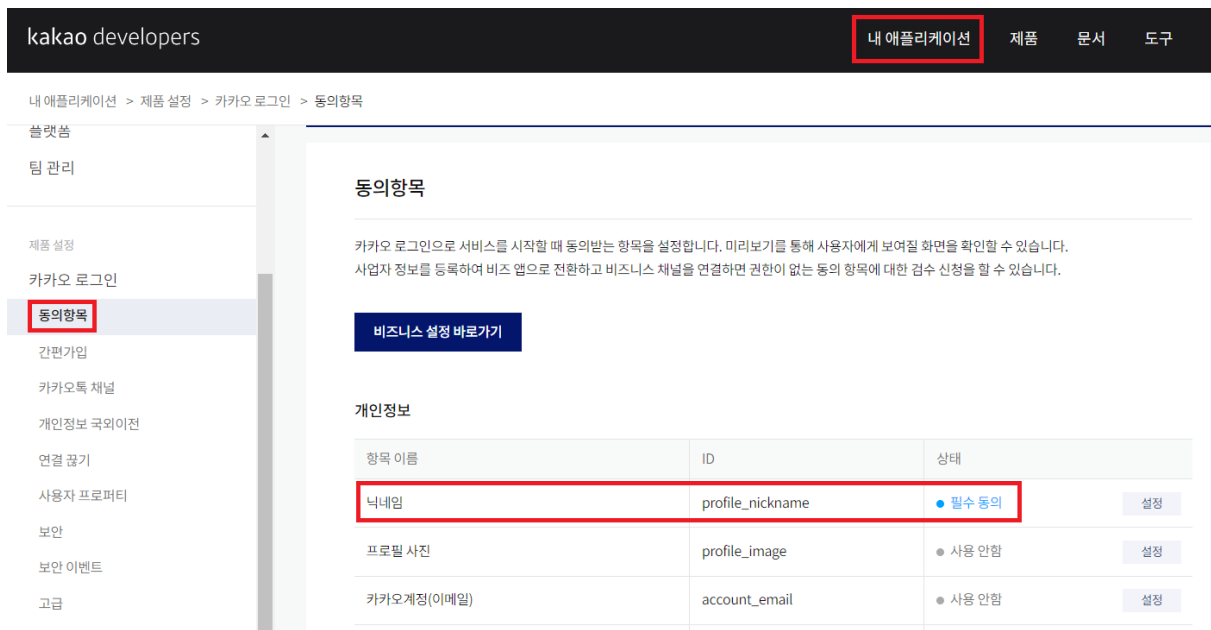
- 사이트 도메인 등록 : 서비스 도메인, Local 도메인 등 Kakao OAuth를 사용할 도메인 등록

4) Redirect URI 설정



- [서비스 도메인]/[Front-end 개발자와 Back-end 개발자가 협의한 Redirect URI]

5) 내 애플리케이션 > 제품 설정 > 카카오톡 로그인 > 동의항목



- 필요한 유저 정보에 대해 동의 항목 설정 추가 : 서비스 목적에 따라 상이

6) 프로젝트 환경 변수 추가

- REST API 키, Redirect URI 환경 변수 추가
 - Front-end : [1. 개발 환경 - 2. 환경 변수 설정 - Front-end 참고](#)
 - Back-end : [1. 개발 환경 - 2. 환경 변수 설정 - Back-end 참고](#)

2. 전자공시 OPENDART API

1) 전자공시 OPENDART API 홈페이지 접속

2) 인증키 신청/관리 > 인증키 신청

- 기본 사항 입력 후 인증키 신청

3) 프로젝트 환경 변수 추가

- API KEY, 파일 경로 등 환경 변수 추가
 - [1. 개발 환경 - 2. 환경 변수 설정 - Back-end 참고](#)