



신문물

삼성 SW 청년 아카데미 서울캠퍼스 6기
외부 서비스 정리

A406팀 뉴스빅

임재현, 김하영, 배용한, 손수연, 유민상, 허은아

목차

1. 카카오 로그인 API

가. 사전 설정

나. 카카오 로그인 프로세스

2. AWS EC2

가. 배포 서버

나. 클러스터 서버


1. 카카오 로그인 API

(가) 사전 설정

1) 카카오 Developers(<https://developers.kakao.com/>)에 로그인 해 애플리케이션을 추가합니다.

애플리케이션 추가하기

앱 아이콘



파일 선택

JPG, GIF, PNG
권장 사이즈 128px, 최대 250KB

앱 이름

신문물

사업자명

뉴스빅

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

취소

저장

2) [내 애플리케이션 - 앱 설정 - 플랫폼] 에서 사이트 도메인을 등록합니다.

Web

삭제수정

사이트 도메인

<https://j6a406.p.ssafy.io>

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

3) [내 애플리케이션 - 제품 설정 - 카카오로그인] 에서 카카오 로그인을 활성화 합니다.

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.

4) [내 애플리케이션 - 제품설정 - 카카오로그인] 에서 Redirect URI를 설정합니다.

Redirect URI

삭제

수정

Redirect URI	https://j6a406.p.ssafy.io/oauth
--------------	---------------------------------

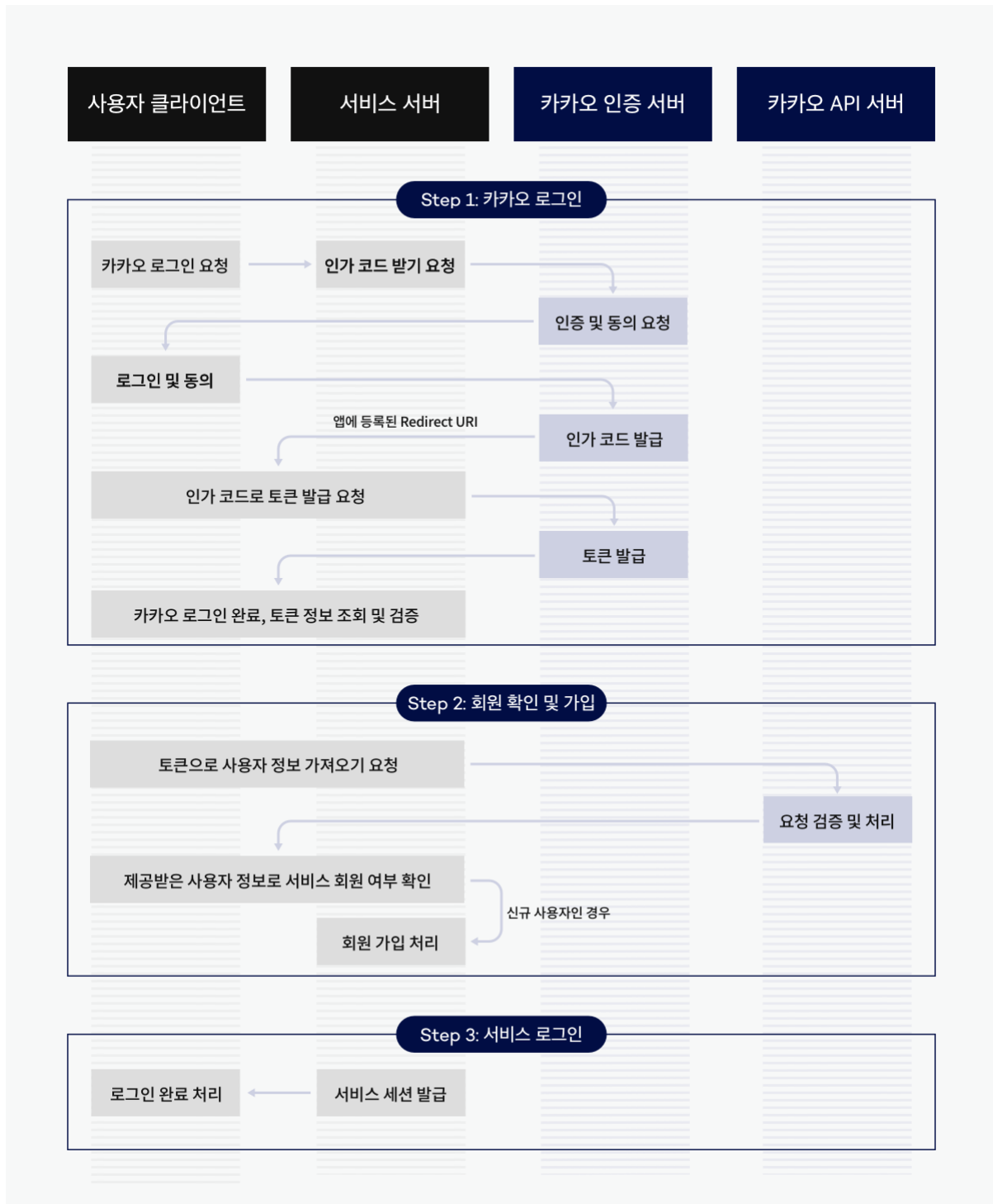
- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)
- REST API로 개발하는 경우 필수로 설정해야 합니다.

여기서 Redirect URI는 카카오 회원정보를 조회할 수 있는 인가 코드가 Redirect되는 URI를 의미합니다.

5) [내 애플리케이션 – 제품설정 – 카카오로그인 – 동의항목] 에서 회원 가입 시 필요한 이메일과 성별 정보를 제공받도록 동의항목으로 설정합니다.

개인정보		
항목 이름	ID	상태
닉네임	profile_nickname	● 사용 안함 설정
프로필 사진	profile_image	● 사용 안함 설정
카카오계정(이메일)	account_email	● 선택 동의 설정
성별	gender	● 선택 동의 설정
연령대	age_range	● 사용 안함 설정
생일	birthday	● 사용 안함 설정
출생 연도	birthyear	○ 권한 없음
카카오계정(전화번호)	phone_number	○ 권한 없음
CI(연계정보)	account_ci	○ 권한 없음
카카오 서비스 내 친구목록(프로필사진, 닉네임, 즐겨찾기 포함)	friends	● 사용 안함 설정
카카오톡 채널 추가 상태 및 내역	plusfriends	○ 권한 없음
배송지정보(수령인명, 배송지 주소, 전화번호)	shipping_address	○ 권한 없음
카카오토리 프로필 URL	story_permalink	● 사용 안함 설정

(나) 카카오 로그인 프로세스



1. [내 애플리케이션 - 앱 설정 - 요약 정보 - 앱 키] 의 REST API 키를 사용해 ([https://kauth.kakao.com/oauth/authorize?response_type=code&client_id=\\${REST_API_KEY}&redirect_uri=\\${REDIRECT_URI}](https://kauth.kakao.com/oauth/authorize?response_type=code&client_id=${REST_API_KEY}&redirect_uri=${REDIRECT_URI}))로 요청을 보냅니다.

2. 1번 요청이 정상적으로 수행되면 사용자가 로그인 및 동의를 수행합니다.

3. 로그인과 동의가 완료되면 인가 코드가 발급되고, 사전에 설정한 Redirect URI를 통해 발급받은 인가 코드로 사용자의 정보를 조회할 수 있는 토큰 발급을 요청합니다.

4. 발급받은 토큰을 사용해 카카오 사용자 정보를 조회할 수 있습니다.

2. AWS EC2

(가) 배포 서버

1) mysql 설치

sudo apt-get install mysql-server 명령어를 사용해 mysql DB 생성

2). Jenkins 설치

sudo apt install Jenkins 명령어를 사용해 Jenkins 설치하고, /etc/default/Jenkins 파일의 HTTP_PORT를 9090으로 변경해 jenkins의 포트 번호를 9090으로 설정합니다.

```
# port for HTTP connector (default 8080; disable with -1)
HTTP_PORT=9090
```

이후 Gitlab의 Repository의 Webhook을 사용해 master 브랜치에 특정 이벤트가 발생할 때 마다 자동으로 서버에서 빌드되도록 설정했습니다.

s06-bigdata-dist-sub2 > S06P22A406 > Webhook Settings

Webhooks

[Webhooks](#) enable you to send notifications to web applications in response to events in a group or project. We recommend using an [integration](#) in preference to a webhook.

URL

URL must be percent-encoded if neccessary.

Secret token

Use this token to validate received payloads. It is sent with the request in the X-Gitlab-Token HTTP header.

Trigger

☒ **Push events**

URL is triggered by a push to the repository

3) Docker 설치

sudo apt install Docker 명령어를 사용해 도커를 설치합니다.

```
ubuntu@ip-172-26-11-242:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
82859403a825	sinmunmul/fastapi	"uvicorn main:app --..."	4 hours ago	Up 4 hours	0.0.0.0:3031->3031/tcp, :::3031->3031/tcp
e2c4f423cca1	nginx	"/docker-entrypoint..."	4 hours ago	Up 4 hours	0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp
d08d0dd0e745	sinmunmul/backend	"java -jar /app.jar"	4 hours ago	Up 4 hours	0.0.0.0:3030->3030/tcp, :::3030->3030/tcp

도커 컨테이너는 FastAPI 서버를 구동하는 fastapi 컨테이너, Spring Boot 서버를 구동하는 backend 컨테이너, Nginx를 구동하는 nginx 컨테이너를 사용해 배포 환경을 구축했습니다.

4) crontab 설정

```
SHELL=/bin/sh
#PATH=/home/ubuntu/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/jvm/zulu-8-amd64/bin
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr,...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )

# * * * * root /home/ubuntu/test.sh >> /var/log/cron.log
0 * * * * root /home/ubuntu/every_hour.sh >> /var/log/cron.log

*/12 * * * * root /home/ubuntu/crawling.sh > /home/ubuntu/cra.log 2>&1 &

15 * * * * root /home/ubuntu/tfidf.sh >> /var/log/cron_tfidf.log
```

주기적으로 뉴스 데이터를 저장하기 위해 크롤러를 등록하고, 클러스터 서버와 연결해 워드카운트, TF-IDF 유사도 계산을 위한 쉘 스크립트를 등록했습니다.

(나) 클러스터 서버

1. 배포 서버에서 Jenkins를 통해 빌드가 완료 되면, Repository의 Hadoop 디렉토리를 JAR 파일로 빌드하고, JAR 파일을 서버의 hadoop 디렉토리로 전송합니다.
2. 이 후 배포서버에서 등록한 crontab에 의해 쉘 스크립트가 실행되면, 클러스터 서버에서 맵리듀스 함수를 실행해 hdfs에 실행 결과를 저장합니다.