



# 포팅 메뉴얼

## 목차

1. 사용 도구
2. 개발 환경
3. 외부 서비스
4. 환경변수 형태
5. 배포하기
6. 빌드 및 실행

## 1. 사용 도구

---

- 이슈 관리 : JIRA
- 형상 관리 : GitLab
- 커뮤니케이션 : Notion, Mattermost
- 디자인 : Figma
- CI/CD : Jenkins
- 웹 브라우저 : Chrome

## 2. 개발 환경

---

### Frontend

name	version
VSCode	1.87.2
React	18.3.1

name	version
npm	10.8.2
Tailwind CSS	3.4.6
Vite	5.3.4
Chrome	127.0.6533.100

## Backend

name	version
IntelliJ	2024.1.1
Java	17.0.11 2024-04-16 LTS
Spring Boot	3.3.2

## Server

name	version
AWS EC2	Ubuntu 20.04.6 LTS

## Service

name	version
Docker	27.1.1
NginX	1.27.0-alpine
MySQL	9.0.1
openVidu	3.0.0-beta2
redis	7.2.5-alpine
rabbitmq	3.13.6
elasticSearch	7.17.1
komoran	3.3.9

## Port

name	port
서비스 접속	8443
jenkins	8081

name	port
mysql	3306
redis	6379
rabbitmq	5672
rabbitmq dashboard	15672
elasticsearch	9200
Api Rest docs	9999 (http)

### 3. 외부 서비스

---

- Kakao OAuth
- Google OAuth
- Chat GPT : gpt-4o-mini
- ETRI AI API/DATA : 발음 평가

### 4. 환경 변수 형태

---

#### 백엔드 환경 변수 목록

- DB 정보
- oauth2 - kakao, google
- openai → llama 로 변경 필요
- livekit
- server url

#### application.yml

```
spring:
  application:
    name: ssafyro

  profiles:
```

```
default: local

datasource:
  url: jdbc:h2:mem:~/ssafyroApplication
  username: sa
  password:

jpa:
  hibernate:
    ddl-auto: none
  properties:
    hibernate:
      default_batch_fetch_size: 100

data:
  redis:
    host: localhost
    port: 6379

rabbitmq:
  host: localhost
  port: 5672
  username: guest
  password: guest

security:
  oauth2:
    client:
      registration:
        google:
          client-id: {client-id}
          client-secret: {client-secret}
          scope:
            - email
            - profile
        kakao:
          client-id: {client-id}
          client-secret: {client-secret}
```

```
client-authentication-method: client_secret_post
redirect-uri: http://localhost:8080/login/oauth2/
authorization-grant-type: authorization_code
client-name: kakao
scope:
  - profile_nickname
  - profile_image
  - account_email
provider:
  kakao:
    authorization-uri: https://kauth.kakao.com/oauth/
    token-uri: https://kauth.kakao.com/oauth/token
    user-info-uri: https://kapi.kakao.com/v2/user/me
    user-name-attribute: id

ai:
  openai:
    api-key: {api-key}
  chat:
    options:
      model: gpt-4o-mini
      temperature: 0.8

livekit:
  api:
    key: {key}
    secret: {secret}

jwt:
  header: Authorization
  issuer: test
  client-secret: test
  expiry-seconds: 180000

front:
  domain: http://localhost:3000
```

```

---
spring:
  config:
    activate:
      on-profile: local

  jpa:
    hibernate:
      ddl-auto: create
    show-sql: true
    properties:
      hibernate:
        format_sql: true
    defer-datasource-initialization: true # (2.5~) Hibernate

  h2:
    console:
      enabled: true

  data:
    redis:
      host: localhost
      port: 6379

  elasticsearch:
    uris: http://localhost:9200

---
spring:
  config:
    activate:
      on-profile: test

  jpa:
    hibernate:
      ddl-auto: create
    show-sql: true

```

```
    properties:
      hibernate:
        format_sql: true

sql:
  init:
    mode: never

data:
  redis:
    host: localhost
    port: 6379

rabbitmq:
  host: localhost
  port: 5672
  username: guest
  password: guest

elasticsearch:
  uris: http://localhost:9200

---
spring:
  config:
    activate:
      on-profile: jenkins

jpa:
  hibernate:
    ddl-auto: create
    show-sql: true
  properties:
    hibernate:
      format_sql: true

sql:
```

```
init:
  mode: never

data:
  redis:
    host: {server-url}
    port: 6379

  rabbitmq:
    host: {server-url}
    port: 5672
    username: guest
    password: guest

  elasticsearch:
    uris: http://{server-url}:9200

---
spring:
  config:
    activate:
      on-profile: develop

  datasource:
    url: jdbc:mysql://{server-url}:3306/{database}?serverTime
    username: {username}
    password: {password}

  jpa:
    hibernate:
      ddl-auto: none
      show-sql: true
    properties:
      hibernate:
        format_sql: true

data:
```



```
redis:
  host: {server-url}
  port: 6379

rabbitmq:
  host: {server-url}
  port: 5672
  username: {username}
  password: {password}

elasticsearch:
  uris: http://{server-url}:9200

security:
  oauth2:
    client:
      registration:
        google:
          client-id: {client-id}
          client-secret: {client-secret}
          scope:
            - email
            - profile
        kakao:
          client-id: {client-id}
          client-secret: {client-secret}
          client-authentication-method: client_secret_post
          redirect-uri: http://{server-url}:8443/login/oauth/
          authorization-grant-type: authorization_code
          client-name: kakao
          scope:
            - profile_nickname
            - profile_image
            - account_email
      provider:
        kakao:
          authorization-uri: https://kauth.kakao.com/oauth/
          token-uri: https://kauth.kakao.com/oauth/token
```

```
user-info-uri: https://kapi.kakao.com/v2/user/me
user-name-attribute: id
```

front:

```
domain: https://{server-url}:8443
```

## 5. 배포하기

### Nginx 설정

```
server {
    listen 8443 ssl;
    server_name i11c201.p.ssafy.io;

    ssl_certificate /etc/letsencrypt/live/i11c201.p.ssafy.io/
    ssl_certificate_key /etc/letsencrypt/live/i11c201.p.ssafy

    location / {
        proxy_pass http://localhost:6666;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location /api {
        proxy_pass http://localhost:9999;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_read_timeout 20m;
    }
}
```

```

location /oauth2 {
    proxy_pass http://localhost:9999;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_read_timeout 20m;
}

location /login/oauth2 {
    proxy_pass http://localhost:9999;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_read_timeout 20m;
}

location /ssafyro-chat {
    proxy_pass http://localhost:9999;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_read_timeout 20m;
}

location /WiseASR {
    proxy_pass http://aiopen.etri.re.kr:8000;

```

```

        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_read_timeout 20m;
        client_max_body_size 10M;
    }

}

```

## Docker 실행

```
docker run -d --name mysql -e MYSQL_ROOT_PASSWORD={root_password}
```

```
docker run -d --name rabbitmq -p 5672:5672 -p 15672:15672 --rm
```

```
docker run --name redis -d -p 6379:6379 redis
```

```
docker run -d --name elasticsearch -p 9200:9200 -p 9300:9300
```

## 6. 빌드 및 실행

### | 초기 필수 데이터 삽입 필요 (더미 데이터)

2. article\_insert.sql
3. coding\_test\_problem.sql
4. essay\_question.sql

## 서버 오류 발생 시 도커 실행 확인

## 서비스 사용 시 필수 설정!

- **[성능 향상]** chrome 설정 → 시스템 → '가능한 경우 그래픽 가속 사용' 활성화
- **[권한 허용]** 면접 연습 시, '마이크 및 녹화 권한' 허용

## Frontend

- (root)/frontend 디렉토리로 이동

```
cd ./frontend
```

- npm 의존성 패키지 설치

```
npm install
```

- Frontend 개발 서버로 구동

```
npm run dev
```

## Backend

- (root)/backend 디렉토리로 이동

```
cd ./backend
```

- jar 파일 생성

```
./gradlew bootJar
```

- jar 파일 실행

```
java -jar ssafyro-0.0.1-SNAPSHOT.jar
```