🔐

환경 변수 정보

프로젝트 구조

```
.
├── ai/
│   └── .env
├── backend/
│   └── src/
│       └── main/
│           └── resources/
│               ├── application-key.yml
│               └── application-db.yml
├── batch/
│   └── src/
│       └── main/
│           └── resources/
│               ├── application-key.yml
│               └── application-db.yml
├── frontend/
│   └── config/
│       └── dev.enc.js
└── env-config/
    └── .env
```

AI

.env

```
#chatgpt API Key
SECRET_KEY = {CHATGPT API KEY}
CLIENT_ID = {CHATGPT CLIENT_ID}
CLIENT_SECRET = {CHATGPT CLIENT_SECRET}
```

Backend

Application-db.yml

```
#개발용 DB profile
spring:
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    url: jdbc:mariadb://{데이터 베이스 ip or 도메인}:{데이터 베이스 port 번}/S08P31C201?
    username: {데이터 베이스 username}
    password: {데이터 베이스 password}
    hikari:
      maximum-pool-size: 2
  config:
    activate:
      on-profile: local

---

#운영 DB profile
spring:
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    url: jdbc:mariadb://ae-book.com:3306/s08p31c201?serverTimezone=UTC&useUnicode=true&characterEncoding=utf8
    username: {데이터 베이스 username}
    password: {데이터 베이스 password}
  config:
    activate:
      on-profile: dev
```

username-password 정보는 docker-compose/.env 의 정보와 일치해야합니다.

Application-key.yml

```yaml
oauth2:
  registration:
    kakao:
      client-id: {카카오 클라이언트 id}
      redirect-uri: http://ae-book.com/user/oauth
      authorization-grant-type: authorization_code
      client-authentication-method: POST
      client-name: Kakao
      scope:
        - profile_nickname
        - profile_image
        - account_email
    provider:
      kakao:
        authorization-uri: https://kauth.kakao.com/oauth/authorize
        token-uri: https://kauth.kakao.com/oauth/token
        user-info-uri: https://kapi.kakao.com/v2/user/me
        user-name-attribute: id

# aws s3
cloud:
  aws:
    s3:
      bucket: aebookbucket
    credentials:
      access-key: {AWS S3 access-key}
      secret-key: {AWS S3 secret-key}
    region:
      static: ap-northeast-2
      auto: false
    stack:
      auto: false

logging:
  level:
    com:
      amazonaws:
        util:
          EC2MetadataUtils: ERROR
#talk
talk:
  api-key: {알리고 api 키}
  user-id: {알리고 user-id}
  sender-key: {sender key}
  lowest-price-tpl-code: TN_0136
  customize-lowest-price-tpl-code: TN_0140
  sender: {알리고 user 번호}
  aebook-url: 'https://ae-book.com'
```

Batch

Application-db.yml

```
#개발용 DB profile
spring:
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    url: jdbc:mariadb://{데이터 베이스 ip명}:3306/S08P31C201?
    username: {데이터 베이스 username}
    password: {데이터 베이스 password}
    hikari:
      maximum-pool-size: 2
  config:
    activate:
      on-profile: local

---

#운영 DB profile
spring:
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    url: jdbc:mariadb://ae-book.com:3306/s08p31c201?serverTimezone=UTC&useUnicode=true
&characterEncoding=utf8
    username: {데이터 베이스 username}
    password: {데이터 베이스 password}
  config:
    activate:
      on-profile: dev
```

Application-key.yml

```
spring:
  config:
    activate:
      on-profile: key

aladin:
  api:
    key: ttbleedy9091441001 # 알라딘 api key
#talk
talk:
  api-key: {알리고 api 키}
  user-id: {알리고 user-id}
  sender-key: {sender key}
  lowest-price-tpl-code: TN_0136
  customize-lowest-price-tpl-code: TN_0140
  sender: {알리고 user 번호}
  aebook-url: 'https://ae-book.com'
```

Frontend

config/dev.env.js

```
'use strict'
const merge = require('webpack-merge')
const prodEnv = require('./prod.env')

module.exports = merge(prodEnv, {
  NODE_ENV: '"development"',
  VUE_APP_REST_API_KEY: '"{카카오 rest api key}"',
  VUE_APP_REDIRECT_URI: '"https://ae-book.com/user/oauth"',
  VUE_APP_LOGOUT_REDIRECT_URI: '"https://ae-book.com"',
  VUE_APP_BASE_URL: '"https://ae-book.com"'
})
```

Env-config

.env

```
MARIADB_USER={mariaDB user}
MARIADB_PASSWORD={mariaDB password}
MARIADB_ROOT_PASSWORD={mariaDB root user password}
```