👨‍🔧

포팅매뉴얼 1번 항목

목차

1. 개발 환경

Backend

- **Server** : Ubuntu 20.04 LTS
- **JDK** : openjdk:17-alpine
- **Spring Boot** 3.2.1
  - **Spring Security**
  - **Spring Data JPA**
- **WAS** : Spring Boot 3.2.1 내장 톰켓 서버
- **Gradle** 8.5

- **DB**
    - **MariaDB** : 8.0.36-0ubuntu0.20.04.1
    - **Redis** : redis-cli 5.0.7

Frontend

- **React** : 18.2.0
- **Node.js** : 20.11.0
- **npm** : 10.2.4
- **vite** : 5.0.8
- **typescript** : 5.2.2
- **상태관리**
    - **tanstack/react-query** : 5.17.15
    - **zustand** : 4.5.0
- **router - react-router-dom** : 6.21.3
- **api 통신 - axios** : 1.6.7
- **css - tailwindcss** : 3.4.1

CI/CD

- **AWS EC2 Ubuntu** 20.11.0
- **Docker** 25.0.0
- **nginx** 1.18.0
- **Jenkins** 2.426.2

협업 툴

- **GitLab**
- **Jira**
- **Gerrit** 3.8.1

- **Mattermost**

2. 환경 변수

Backend(application.yml) - 로컬

```
server:
  port: 8081
  servlet:
    context-path: /api

spring:
  # DB Connection Info
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/ninetofive?useSSL=false&
    username: ninetofive
    password: 98765

  # Jpa Setting
  jpa:
    open-in-view: false
    hibernate:
      ddl-auto:
    properties:
      hibernate:
        default_batch_fetch_size: 1000
        format_sql: true
#         show_sql: true  # DB로 날리는 쿼리를 출력합니다. - inhwa

  # jwt setting
  jwt:
    secret:
      access: R0dVTVNnZ3VtdGxlQWNjZXNzVG9rZW5Gb3JTZWNyZXRLZXk
      refresh: R0dVTVNnZ3VtdGxlUmVmcmVzaFRva2VuRm9yU2VjcmV0S2
      develop_dummy_1: bmluZXRvZml2ZeyEnOykgO2YuA
      develop_dummy_2: bmluZXRvZml2ZeyLoOywveyXvQ
      develop_dummy_3: bmluZXRvZml2ZeydtOyasOyEsQ
      develop_dummy_4: bmluZXRvZml2ZeydtOybkOyjvA
      develop_dummy_5: bmluZXRvZml2ZeyghOyngO2ZmA
      develop_dummy_6: bmluZXRvZml2ZeyhsOyduO2ZlA

  # Redis setting
  data:
    redis:
      port: 6379
      host: localhost
      address: redis://127.0.0.1:6379
      password: 1234

  # Java Mail Sender Setting
  mail:
    host: smtp.gmail.com
    port: 587
    username: 9to5ggumms@gmail.com
    password: vcla ybsf wsvn ivgf
    properties:
      mail:
        smtp:
          auth: true
          starttls:
            enable: true

  # Web Address
  web:
    baseUrl: http://localhost:8081

  # Kakao, google oauth
  security:
    oauth2:
      client:
        registration:
          kakao:
            client-id: 29eeda8b3276b5868e20a11f3b77b183
            client-secret: oPY938rjWAqVCJvsWhKo6Uvr7kQziQpC
```

```
            authorization-grant-type: authorization_code
            client-authentication-method: client_secret_post
            redirect-uri: http://localhost:5173
            scope:
              - profile_nickname
              - profile_image
              - account_email
          google:
            client-id: 437109759271-sfaobp93ep5da5dc27srfpq66
            client-secret: GOCSPX-FPTW0qAOJd1iWCXGouJFq-XZ172
            scope:
              - openid
              - profile
              - email
            redirect-uri: http://localhost:5173
            client-authentication-method: basic
            authorization-grant-type: authorization_code
        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
          google:
            authorization-uri: https://accounts.google.com/o/
            token-uri: https://oauth2.googleapis.com/token
            user-info-uri: https://www.googleapis.com/oauth2/
            user-name-attribute: sub

# Swagger
springdoc:
  packages-to-scan: com.ggums.ggumtle.controller
  default-consumes-media-type: application/json;charset=UTF-8
  default-produces-media-type: application/json;charset=UTF-8
  swagger-ui:
    path: /swagger-ui
    disable-swagger-default-url: true
    display-request-duration: true
    tags-sorter: alpha
    operations-sorter: alpha
    info:
      title: GGUMTLE API Server
      description: GGUMTLE GGUMTLE
      version: 0.0.1

# logging
logging:
  level:
    root: INFO
    web: DEBUG
    com.ggums.ggumtle: DEBUG
    org.springframework.security: DEBUG
    org.springframework.web.cors: DEBUG
    org.hibernate.sql: DEBUG
    org.hibernate.type.descriptor.sql: trace
```

Backend(application-prod.yml) - 배포

```
#auth 설정
auth:
  whiteList: ENC(SgJQQGAOGGVgqocjx9c5r9TtTmOFICQ862r2TfpuxB3o
jwt:
  header: Authorization
  secret: ENC(fv3uJfbrB/x6anJJ9WlWZcBGj/af5ty4Zl2aZOmNCE1zqLe
  token-access-time: ENC(XsKWqhfuxpvk2x0PWKAn+w==)
  token-refresh-time: ENC(XsKWqhfuxpvk2x0PWKAn+w==)
spring:
  security:
    oauth2:
      client:
        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
          naver:
            authorization_uri: https://nid.naver.com/oauth2.0
            token_uri: https://nid.naver.com/oauth2.0/token
            user-info-uri: https://openapi.naver.com/v1/nid/m
            user_name_attribute: response

--- # local
spring:
  config:
    activate:
      on-profile: "auth-local"
  security:
    oauth2:
      client:
        registration:
          kakao:
            client-name: kakao
            client-id: ENC(C1BmOi7lmh67oOzRMKHI3pnNIHfXljSE4H
            redirect-uri: ENC(hHI41AUmy4Nss1sLWV5an1iCW7kzLc2
            authorization-grant-type: authorization_code
            client-authentication-method: POST
            scope:
              - account_email
          naver:
            client-name: naver
            client-id: ENC(PuOzCnHPlzXLtF2QNi8/df0LUXyrBx4hi8
            client-secret: ENC(o16SuX8pfIhOy3RIEfkCcoaUGGVhYm
            redirect-uri: ENC(SwOi0TZSqX4+pw1xt5W6PuNEVzzZ/Bc
            authorization-grant-type: authorization_code
            scope:
              - email

--- # server
spring:
  config:
    activate:
      on-profile: "auth-server"
  security:
    oauth2:
      client:
        registration:
          kakao:
            client-name: kakao
            client-id: ENC(C1BmOi7lmh67oOzRMKHI3pnNIHfXljSE4H
            redirect-uri: ENC(DXQPeCU88Dui1O67yubOE3ymYp/uMpN
            authorization-grant-type: authorization_code
            client-authentication-method: POST
            scope:
              - account_email
          naver:
            client-name: naver
            client-id: ENC(PuOzCnHPlzXLtF2QNi8/df0LUXyrBx4hi8
            client-secret: ENC(o16SuX8pfIhOy3RIEfkCcoaUGGVhYm
            redirect-uri: ENC(8EBYCxxHdMUUGkp+1++eJZCMfRMWQ77
            authorization-grant-type: authorization_code
            scope:
              - email
```

Frontend(.env)

```
VITE_KAKAO_JAVASCRIPT_KEY=072dba1ae2db13e72fb3cd62d68d2375
VITE_BASE_URL=https://i10d111.p.ssafy.io/api
VITE_USER1_TOKEN=bmluZXRvZml2ZeyhsOyduO2ZlA
```

3. 배포 시 특이사항

인증서 발급

- CertBot 사용하여 인증서 발급

나머지 배포 관련 스크립트는 deploy 폴더 안에 모두 있음

4. DB접속 정보

MySQL

- Hostname : i10d111.p.ssafy.io
- port: 3306
- Username: ggumms
- Password: 98765

Redis

- Hostname : i10d111.p.ssafy.io
- port: 6379
- Password: 98765