

이음페이

B103 포팅 매뉴얼 1

빌드 및 배포 정리

버전 정리

Back-end

Java - 17
SpringBoot - 2.7.0
SpringCloud - 2021.0.3
Swagger - 1.6.6
JavaMail - 1.6.2
QueryDSL - 5.0.0
MariaDB - 11.3.2
Redis - 7.2.4
Kafka - 2.8.1
zookeeper - 3.4.13
Jenkins - 2.440.1
Docker - 2.25

Front-end

Node - 20.11.1
Next.js - 14.1.3
MSW - 2.0
Zustand - 4.0
TanStack Query - 5.0
Storybook - 7.0
SCSS - 1.72
TypeScript - 5

환경 변수 - **application.yml** - 각 서버마다 다른 파일 사용

```
// Common Server

server:
  port: 8081

spring:
  mail:
    host: smtp.gmail.com
    port: 587
    username: # 여기에 이메일 주소 사용
    password: # 여기에 앱 비밀번호 사용
    properties:
      mail:
        smtp:
          auth: true
          starttls:
            enable: true

  datasource:
    url: # mariaDB 주소 사용
    username: # DB 사용자 이름 사용
    password: # DB password 사용
    driver-class-name: org.mariadb.jdbc.Driver

  jpa:
    open-in-view: false
    hibernate:
      ddl-auto: validate # or create, validate, none
    show-sql: true
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MariaDBDialect

  redis:
    host: # 여기에 redis host 주소 사용
    port: 6379
    password: # 여기에 redis password 사용
    lettuce:
      pool:
        max-active: 10
        max-idle: 5
        min-idle: 1

  servlet:
```

multipart:
maxFileSize: 10MB

data:
redis:
repositories:
enabled: false

kafka:
producer:
bootstrap-servers: # 여기에 kafka 주소 사용

logging:
level:
org.springframework.security: INFO
org.springframework.boot.autoconfigure.security: OFF

gateway:
donation: http://localhost:8082/api
pay: http://localhost:8083/api
alert: http://localhost:8084/api

jwt:
secret-key:
44LRhp5SLaazMmuastyRW6hR323scRALI9Qspzz2tl0gkeC8H9l25jIPjBSikMJymrnzxTw6l7yF
hekBdMUPrETSMnA9Ema4uDOYYPG2veMGuhd4owN4riT6XjWndSQPdsav
access-time: 18000
refresh-time: 604800

filter-path:
paths:
member: "/api/member"
member-login: "/api/member/login"
check-member: "/api/member/exist"
get-authentication-code: "/api/auth"
apiDocs: "/api-docs/**"
v2ApiDocs: "/v2/api-docs/**"
v3ApiDocs: "/v3/api-docs/**"
webjars: "/webjars/**"
swagger: "/swagger/**"
swaggerUi: "/swagger-ui/**"
swaggerConfig: "/swagger-config/**"
swaggerResources: "/swagger-resources/**"

```
// Pay Server

server:
  port: 8083
naver:
  cloud:
    ocr:
      url: # 여기에 naver-cloud OCR 주소 사용
    client:
      secret: # 여기에 naver-cloud secret key 사용

spring:
  servlet:
    multipart:
      maxFileSize: 10MB

datasource:
  url: # 여기에 mariaDB 주소 사용
  username: # 여기에 mariaDB - username 사용
  password: # 여기에 mariaDB - password 사용
  driver-class-name: org.mariadb.jdbc.Driver
jpa:
  open-in-view: false
  hibernate:
    ddl-auto: update # or create, validate, none
  show-sql: true
  properties:
    hibernate:
      dialect: org.hibernate.dialect.MariaDBDialect
```

// Funding Server

server:

port: 8082

spring:

datasource:

url: # 여기에 mariaDB 주소 사용

username: # 여기에 mariaDB-username 사용

password: # 여기에 mariaDB-password 사용

driver-class-name: org.mariadb.jdbc.Driver

jpa:

open-in-view: false

hibernate:

ddl-auto: update # or create, validate, none

show-sql: true

properties:

hibernate:

dialect: org.hibernate.dialect.MariaDBDialect

// Alert Server

server:

port: 8084

spring:

kafka:

consumer:

bootstrap-servers: # 여기에 kafka 주소 사용

group-id: ieum-consumer

topic:

transferReceived: transfer-received

sseConnectionRequest: sse-connection-request

특이사항

MSA-BFF 사용

본 프로젝트는 은행 API의 사용에 있어서의 어려움을 극복하기 위해 MSA-BFF 구조를 도입했습니다. 각 기능을 독립된 서버로 구성함으로써, 기능의 세분화 및 독립성을 높였습니다. 실제 금융권에서도 MSA 구조를 활용하고 있음을 현직자 컨설팅을 통해 확인받았습니다. 일반적인 MSA 구조는 각 기능별로 독립된 인스턴스와 데이터베이스를 보유하지만, 리소스의 제한으로 인해 모든 서버를 하나의 인스턴스에서 운영할 필요가 있었습니다. 이러한 문제를 해결하기 위해, 각 기능이 하나의 공통 데이터베이스를 참조하며 생성된 정보를 Common 모듈에서 조합하여 프론트엔드로 전달하는 방식을 채택했습니다.

PWA 사용

간편결제 시스템의 주 사용처가 모바일 환경임을 고려하여, PWA(Progressive Web App) 방식으로 애플리케이션을 개발했습니다. 이를 통해 사용자는 앱 설치 없이도 웹을 통해 앱과 유사한 경험을 할 수 있게 되었습니다. 하지만, 안드로이드와 iOS 간 PWA 지원에 차이가 있음을 발견했으며, 이에 따라 안드로이드 환경에 보다 최적화하여 개발을 진행했습니다. 특히 iOS에서의 제한적인 PWA 지원 범위를 고려하여, 사용자 경험을 극대화하기 위한 추가적인 조치를 취했습니다.

DB 접속 정보

```
// DB 관련 YML 설정

spring:
  datasource:
    url: # mariaDB 주소 사용
    username: # DB 사용자 이름 사용
    password: # DB password 사용
    driver-class-name: org.mariadb.jdbc.Driver

  jpa:
    open-in-view: false
    hibernate:
      ddl-auto: validate # or create, validate, none
    show-sql: true
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MariaDBDialect

  redis:
    host: # 여기에 redis host 주소 사용
    port: 6379
    password: # 여기에 redis password 사용
    lettuce:
      pool:
        max-active: 10
        max-idle: 5
        min-idle: 1

  servlet:
    multipart:
      maxFileSize: 10MB

  data:
    redis:
      repositories:
        enabled: false
```

8

