포팅 메뉴얼

사용 서비스 및 버전

Frontend

React 18
Redux 5.0.1
Next 14.2.16
Tailwind 3.4.1
Shadcn-ui 0.9.2
Three 0.170.0
Stompjs 2.3.3

DB

MySQL 9.1.0
MongoDB 8.0.3
Redis 7.4.1

ETC

RabbitMQ 4.0.3

Backend

Spring boot

jdk17
spring boot 3.3.4
spring data jpa
spring cloud openfeign 4.1.3
spring cloud netflix-eureka-client:4.1.3
querydsl 5.0.0

Fast Api

uvicorn
scikit-learn
numpy
pandas

Infra

AWS EC2
AWS S3
Ubuntu 20.04.6 LTS
Docker 27.3.1
Nginx 1.27.2
Jenkins 2.484

프로젝트 설정

frontend

backend (application.yml)

discovery-service

```
server:
  port: 8761

spring:
  application:
    name: homeggu-discovery

eureka:
  client:
    register-with-eureka: false
    fetch-registry: false
```

gateway-service

```
server:
  port: 8000
  servlet:
    context-path: /api

eureka:
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://discovery-backend:8761/eureka

spring:
  application:
    name: homeggu-gateway

  security:
    oauth2:
      client:
        registration:
          kakao:
            client-name: kakao
            authorization-grant-type: authorization_code
            redirect-uri: https://frontend/kakaoCallback;
            client-id: <발급 필요>
            client-secret: <발급 필요>
            client-authentication-method: <발급 필요>
            scope:
              - profile_nickname
              - account_email

  cloud:
    gateway:
      #      default-filters:
      #        - name: GlobalFilter
      #          args:
      #            baseMessage: Spring Cloud Gateway GlobalFilter
      #            preLogger: true
      #            postLogger: true
      routes:
        - id: homeggu-pay
              uri: lb://HOMEGGU-PAY
          predicates:
            - Path=/pay/**
        - id: homeggu-goods
          uri: lb://HOMEGGU-GOODS
          predicates:
            - Path=/goods/**
        - id: homeggu-chat
          uri: lb://HOMEGGU-CHAT
          predicates:
            - Path=/chat/**
        - id: homeggu-user
          uri: lb://HOMEGGU-USER
          predicates:
            - Path=/user/**

jwt:
  salt: <토큰 salt>
  secret: <jwt 암호 키>
  access-token:
    expiretime: 3600000
  refresh-token:
    expiretime: 2592000000
```

user-service

```
server:
  port: 8084
  servlet:
    context-path: /api

spring:
```

```
  application:
    name: homeggu-user

  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://mysql:3306/homeggu_user
    username: root
    password: homeggu11

  data:
    web:
      pageable:
        max-page-size: 2000
        default-page-size: 10
    redis:
      host: redis
      port: 6379

  servlet:
    multipart:
      max-file-size: 25MB
      max-request-size: 25MB

  jpa:
    open-in-view: false
    properties:
      hibernate:
        default_batch_fetch_size: 50
        show_sql: true
        format_sql: true
        dialect: org.hibernate.dialect.MySQL8Dialect
        jdbc:
          time_zone: Asia/Seoul

  sql:
    init:
      mode: always


eureka:
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://discovery-backend:8761/eureka


random-nickname:
  first-names: "수상한,사랑스러운,비련의,애처로운,멋있는,천진난만한,소심한"
  last-names: "돌멩이,여주인공,길고양이,억만장자,나뭇잎,행인1,주인공,화분,강아지,비둘기,히어로,먼지,버스,마법사,댄서,모범생,알바생,할아버지,할머니,외
```

goods-service

```
server:
  port: 8082
  servlet:
    context-path: /api

spring:
  application:
    name: homeggu-goods

  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://mysql:3306/homeggu_goods
    username: root
    password: homeggu11

  data:
    web:
      pageable:
        max-page-size: 2000
        default-page-size: 10
    redis:
      host: redis
      port: 6379
```

```
  servlet:
    multipart:
      max-file-size: 25MB
      max-request-size: 25MB

  jpa:
    open-in-view: false
    properties:
      hibernate:
        default_batch_fetch_size: 50
        show_sql: true
        format_sql: true
        dialect: org.hibernate.dialect.MySQL8Dialect
        jdbc:
          time_zone: Asia/Seoul

  sql:
    init:
      mode: always


eureka:
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://discovery-backend:8761/eureka


cloud:
  aws:
    s3:
      bucket: homeggu-s3
    stack.auto: false
    region.static: ap-northeast-2
    credentials:
      accessKey: <S3 액세스 키>
      secretKey: <S3 시크릿 키>
```

pay-service

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

spring:
  application:
    name: homeggu-pay

  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://mysql:3306/homeggu_pay
    username: root
    password: homeggu11

  data:
    web:
      pageable:
        max-page-size: 2000
        default-page-size: 10
    redis:
      host: redis
      port: 6379

  servlet:
    multipart:
      max-file-size: 25MB
      max-request-size: 25MB

  jpa:
    open-in-view: false
    properties:
      hibernate:
        default_batch_fetch_size: 50
```

```
        show_sql: true
        format_sql: true
        dialect: org.hibernate.dialect.MySQL8Dialect
        jdbc:
          time_zone: Asia/Seoul

  sql:
    init:
      mode: always


eureka:
  client:
    register-with-eureka: true
    fetch-registry: true
    service-url:
      defaultZone: http://discovery-backend:8761/eureka
```

chat-service

```
server:
  port: 8083

spring:
  application:
    name: homeggu-chat

  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://mysql:3306/homeggu_chat
    username: root
    password: homeggu11

  data:
    web:
      pageable:
        max-page-size: 2000
        default-page-size: 10
    mongodb:
      uri: mongodb://root:homeggu11@mongodb:27017/chat_message?authSource=admin&authMechanism=SCRAM-SHA-1

  stomp:
    port: 61613

  rabbitmq:
    host: rabbitmq
    port: 5672
    username: homeggu
    password: homeggu11

  servlet:
    multipart:
      max-file-size: 25MB
      max-request-size: 25MB

  jpa:
    open-in-view: false
    properties:
      hibernate:
        default_batch_fetch_size: 50
        show_sql: true
        format_sql: true
        dialect: org.hibernate.dialect.MySQL8Dialect
        jdbc:
          time_zone: Asia/Seoul

  sql:
    init:
      mode: always

 eureka:
   client:
     register-with-eureka: true
     fetch-registry: true
```

```
    service-url:
      defaultZone: http://discovery-backend:8761/eureka
```

배포 설정 (docker-compose.yml)

웹 서버 관련

```
services:

  nginx:
    image: nginx
    container_name: nginx
    restart: always
    environment:
      TZ: Asia/Seoul
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/nginx.conf
      - ./nginx/conf.d/:/etc/nginx/conf.d/
      - ./nginx/ssl:/etc/letsencrypt
      - ./nginx/html:/usr/share/nginx/html
    command: "/bin/sh -c 'while :; do sleep 6h & wait $${!}; nginx -s reload; done & nginx -g \"daemon off;\"'"
    networks:
      - homeggu

  certbot:
    image: certbot/certbot
    container_name: certbot
    # restart: unless-stopped
    environment:
      TZ: Asia/Seoul
    volumes:
      - ./nginx/ssl:/etc/letsencrypt
      - ./nginx/html:/usr/share/nginx/html
    entrypoint: "/bin/sh -c 'trap exit TERM; while :; do certbot renew; sleep 12h & wait $${!}; done;'"
    depends_on:
      - nginx
    networks:
      - homeggu

  frontend:
    image: frontend
    container_name: frontend
    environment:
      TZ: Asia/Seoul
    expose:
      - "3000"
    #volumes:
      #- ./frontend/nginx.conf:/etc/nginx/nginx.conf
      #- ./frontend/logs:/var/log/nginx
    networks:
      - homeggu

  jenkins:
    image: jenkins/jenkins
    container_name: jenkins
    restart: unless-stopped
    volumes:
      - jenkins_home:/var/jenkins_home
      - ./jenkins:/var/jenkins_shared
      - /var/run/docker.sock:/var/run/docker.sock
      - /usr/bin/docker:/usr/bin/docker
      - /home/ubuntu:/home/ubuntu
    environment:
      JENKINS_OPTS: --prefix=/jenkins
      TZ: Asia/Seoul
    networks:
      - homeggu

volumes:
  jenkins_home:
```

```
networks:
  homeggu:
    external: true
```

백엔드 관련

```
services:

  pay-backend:
    image: pay-backend
    container_name: pay-backend
    environment:
      - TZ=Asia/Seoul
      - EUREKA_CLIENT_SERVICE_URL=http://discovery-backend:8761/eureka/
        #expose:
    ports:
      - "8081:8081"
    depends_on:
      - gateway-backend
      - discovery-backend
    networks:
      - homeggu

  goods-backend:
    image: goods-backend
    container_name: goods-backend
    environment:
      - TZ=Asia/Seoul
      - EUREKA_CLIENT_SERVICE_URL=http://discovery-backend:8761/eureka/
        #expose:
    ports:
      - "8082:8082"
    depends_on:
      - gateway-backend
      - discovery-backend
    networks:
      - homeggu

  chat-backend:
    image: chat-backend
    container_name: chat-backend
    environment:
      - TZ=Asia/Seoul
      - EUREKA_CLIENT_SERVICE_URL=http://discovery-backend:8761/eureka/
    ports:
      - "8083:8083"
      - "61613:61613"
    depends_on:
      - gateway-backend
      - discovery-backend
    networks:
      - homeggu

  user-backend:
    image: user-backend
    container_name: user-backend
    environment:
      - TZ=Asia/Seoul
      - EUREKA_CLIENT_SERVICE_URL=http://discovery-backend:8761/eureka/
        #expose:
    ports:
      - "8084:8084"
    depends_on:
      - gateway-backend
      - discovery-backend
    networks:
      - homeggu

  recommendation-backend:
    image: recommendation-backend
    container_name: recommendation-backend
    environment:
      TZ: Asia/Seoul
    ports:
      - "8001:8001"
```

```
    networks:
      - homeggu

  discovery-backend:
    image: discovery-backend
    container_name: discovery-backend
    hostname: discovery-backend
    environment:
      TZ: Asia/Seoul
    ports:
      - "8761:8761"
    networks:
      - homeggu

  gateway-backend:
    image: gateway-backend
    container_name: gateway-backend
    environment:
      - TZ=Asia/Seoul
      - EUREKA_CLIENT_SERVICE_URL=http://discovery-backend:8761/eureka/
      #expose:
    ports:
      - "8000:8000"
    depends_on:
      - discovery-backend
    networks:
      - homeggu

networks:
  homeggu:
    external: true
```

DB, ETC

```
services:

  redis:
    image: redis
    container_name: redis
    expose:
      - "6379"
    volumes:
      - ./redis/redis.conf:/usr/local/etc/redis/redis.conf
    command: redis-server /usr/local/etc/redis/redis.conf
    networks:
      - homeggu

  rabbitmq:
    image: rabbitmq:4.0-management
    container_name: rabbitmq
    ports:
      - "5672:5672"
      - "61613:61613"
      - "15672:15672"
    environment:
      RABBITMQ_DEFAULT_USER: homeggu
      RABBITMQ_DEFAULT_PASS: homeggu11
      TZ: Asia/Seoul
    command: >
      bash -c "
      rabbitmq-plugins enable rabbitmq_management &&
      rabbitmq-plugins enable rabbitmq_stomp &&
      rabbitmqctl add_user homeggu homeggu11 &&
      rabbitmqctl set_permissions -p / homeggu '.*' '.*' '.*' &&
      rabbitmqctl set_user_tags homeggu administrator &&
      rabbitmq-server"
    networks:
      - homeggu

  mongodb:
    image: mongodb/mongodb-community-server
    container_name: mongodb
    ports:
      - "27017:27017"
    environment:
```

```
      MONGODB_INITDB_ROOT_USERNAME: root
      MONGODB_INITDB_ROOT_PASSWORD: homeggu11
      TZ: Asia/Seoul
    volumes:
      - mongodb-data:/data/db
    networks:
      - homeggu

  mysql:
    image: mysql
    container_name: mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: homeggu11
      TZ: Asia/Seoul
    ports:
      - "3306:3306"
    volumes:
      - ./mysql/data:/var/lib
      - ./mysql/dump:/var/lib/mysql
    networks:
      - homeggu

volumes:
  mongodb-data:

networks:
  homeggu:
    external: true
```

nginx 설정

```
user  nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log notice;
pid        /var/run/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"'
                      '"$host" "$server_name" "$request_uri" "$uri" '
                      '"$request_body" "$args" "$upstream_addr" "$upstream_status"';

    access_log  /var/log/nginx/access.log  main;

    chunked_transfer_encoding on;

    sendfile        on;

    client_max_body_size 50M;

    keepalive_timeout  200;
    proxy_read_timeout 200s;
    proxy_send_timeout 200s;
    send_timeout 200s;

    limit_req_zone $binary_remote_addr zone=mylimit:1000m rate=1000r/s;

    include /etc/nginx/conf.d/*.conf;

    server {

        listen 80;
```

```
        server_name k11b206.p.ssafy.io;

        location /.well-known/acme-challenge/ {
            root /usr/share/nginx/html;
        }

        location / {
            return 301 https://$host$request_uri;
        }

    }

    server {
        listen 443 ssl;
        server_name k11b206.p.ssafy.io;

        ssl_certificate /etc/letsencrypt/live/k11b206.p.ssafy.io/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/k11b206.p.ssafy.io/privkey.pem;

        client_max_body_size 10G;

        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_set_header X-Forwarded-Port 443;

        #proxy_redirect off;

        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";

        proxy_set_header Connection keep-alive;

        location / {
            limit_req zone=mylimit burst=10000 nodelay;

            limit_req_status 429;

            proxy_pass http://frontend:3000;
            root /usr/share/nginx/html;
            index index.html;
        }

        location /jenkins {
            proxy_pass http://jenkins:8080/jenkins;
        }

        location /api/user {
            proxy_pass http://user-backend:8084;
            proxy_read_timeout 120s; # Upstream 서버에서 응답을 기다리는 시간
            proxy_connect_timeout 30s; # Upstream 서버와 연결하는 시간
            proxy_send_timeout 120s; # Upstream 서버로 데이터를 보내는 시간
        }

        location /api/goods {
            proxy_pass http://goods-backend:8082;
        }

        location /api/pay {
            proxy_pass http://pay-backend:8081;
        }

        location /api/chat {
            proxy_pass http://chat-backend:8083;
        }

        location ~ ^/(pub|exchange|chat|ws) {
            proxy_pass http://chat-backend:8083;
        }

        location /rabbitmq/ {
            proxy_pass http://rabbitmq:15672/;
            limit_req zone=mylimit burst=5000 nodelay;
        }
```

```
    }

}
```