

FrontEnd, Backend 프레임워크, 언어 버전

FrontEnd

Android Studio , v 2024.01

Android API 35

Flutter 3.22.3

Dart 3.4.4

Flutter SDK : 35

Backend

Intelij v 2024.01

Java 17

Spring Boot 3.3.2

MySQL 8.0,

Redis 7.4.0,

Elastic Search 8.14.3

Firebase Cloud Messaging

FrontEnd 실행방법 & 환경변수

실행방법

1. flutter clean
2. dart run build_runner watch
3. flutter run

순입니다.

환경변수

a : kakaoNativeAppKey

b : kakaoJavaScriptAppKey

c : naverClientId

BackEnd 실행방법 & 환경변수

실행방법

Eureka -> GateWay -> MNServer -> UTRServer 로 서버를 업합니다.

- 모든 서버를 킨 후, 1분 후부터 서비스가 동작됩니다.

환경변수

Eureka Server

Application.properties

Server.port = 8761

spring.application.name=eureka

eureka.client.register-with-eureka=false : 유레카 서버 등록 환경변수입니다.

eureka.client.fetch-registry= false : 유레카 서버 정보 가져오는 환경변수입니다.

Gateway Server

Application.yml

server:

port: 80

eureka:

client:

register-with-eureka: true :유레카 서버에 서버이름을 등록하는 환경변수입니다.

fetch-registry: true : 유레카 서버에 서버 메타데이터를 등록하는 환경변수 입니다.

serviceUrl:

defaultZone: <http://localhost:8761/eureka/> : 로컬에 올라온 유레카서버 uri를 저장합니다.

JWT secret key : JWT 키입니다.

MN-Server

Application.properties

server.port = 8082

spring.application.name=MNSERVER

spring.profiles.active=local

spring.data.redis.port=6379 : 레디스 서버포트 환경변수 입니다.

spring.data.redis.host=i11a802.p.ssafy.io : 레디스 서버 uri 환경변수 입니다.

spring.data.redis.password=mdnd : 레디스 서버 password 환경변수입니다.

MySQL

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver : mysql 의존성 환경변수 입니다.

spring.datasource.url=jdbc:mysql://i11a802.p.ssafy.io:3306/myground?serverTimezone=UTC&characterEncoding=UTF-8 : mysql 주소 환경변수입니다.

spring.datasource.username=mtnt : mysql 유저네임 환경변수 입니다.

spring.datasource.password=mdnd : mysql password 환경변수 입니다.

spring.jpa.properties.hibernate.dialect=org.hibernate.spatial.dialect.mysql.MySQLSpatialDialect

Eureka

유레카 서버 세팅 환경변수 입니다.

eureka.client.register-with-eureka=true

eureka.client.fetch-registry=true

eureka.client.service-url.defaultZone= http://localhost:8761/eureka

UTR-SERVER

Application.properties

서버 포트를 설정합니다.

#server.address=localhost

서버 주소를 설정합니다.

#server.servlet.contextPath=/

웹 애플리케이션의 컨텍스트 경로를 설정합니다.

server.port=8081

spring.profiles.active=local

애플리케이션 이름을 설정합니다.

```
spring.application.name=UTRSERVER

#server.address=localhost

## 서버 주소를 설정합니다.

#server.servlet.contextPath=/

## 웹 애플리케이션의 컨텍스트 경로를 설정합니다.

# 서버 포트를 설정합니다.

#server.address=localhost

# 서버 주소를 설정합니다.

#server.servlet.contextPath=/

# 웹 애플리케이션의 컨텍스트 경로를 설정합니다.

server.servlet.encoding.charset=UTF-8

# HTTP 요청과 응답의 문자 인코딩을 UTF-8로 설정합니다.

server.servlet.encoding.enabled=true

# HTTP 요청과 응답의 문자 인코딩을 활성화합니다.

server.servlet.encoding.force=true

# HTTP 요청과 응답의 문자 인코딩을 강제로 적용합니다.

# Swagger 설정

springdoc.packages-to-scan=com.mongddangnaeddang.UTRServer

springdoc.default-consumes-media-type=application/json;charset=UTF-8

springdoc.default-produces-media-type=application/json;charset=UTF-8

springdoc.swagger-ui.path=/

springdoc.swagger-ui.disable-swagger-default-url=true

springdoc.swagger-ui.display-request-duration=true

springdoc.swagger-ui.operations-sorter=alpha

#spring.jpa.hibernate.naming.implicit-
strategy=org.springframework.boot.orm.jpa.hibernate.SpringImplicitNamingStrategy
```

Hibernate의 암시적 네이밍 전략을 SpringImplicitNamingStrategy로 설정합니다.

```
#spring.jpa.hibernate.naming.physical-  
strategy=org.springframework.boot.orm.jpa.hibernate.SpringPhysicalNamingStrategy
```

Hibernate의 물리적 네이밍 전략을 SpringPhysicalNamingStrategy로 설정합니다.

```
#spring.jpa.hibernate.ddl-auto=create
```

Hibernate의 DDL 생성 전략을 create로 설정합니다. 테이블이 생성됩니다.

```
#spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect
```

Hibernate의 SQL 방언을 MySQL 5.7로 설정합니다.

```
spring.datasource.url=jdbc:mysql://i11a802.p.ssafy.io:3306/myground?useUnicode=true&characterE  
ncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTimeBehavior=convertToNull&rewriteBatched  
Statements=true
```

```
#spring.datasource.url=jdbc:mysql://52.79.172.95:3306/myround?useUnicode=true&characterEncod  
ing=utf8&serverTimezone=Asia/Seoul&zeroDateTimeBehavior=convertToNull&rewriteBatchedState  
ments=true
```

데이터베이스 연결 URL을 설정합니다.

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

데이터베이스 드라이버 클래스 이름을 설정합니다.

```
spring.datasource.hikari.username=mtnt
```

HikariCP 데이터베이스 접속 사용자 이름을 설정합니다.

```
spring.datasource.hikari.password=mdnd
```

HikariCP 데이터베이스 접속 비밀번호를 설정합니다.

elasticsearch 설정

```
#spring.elasticsearch.username=elastic
```

```
#spring.elasticsearch.password=qkrtpjrs
```

```
#spring.elasticsearch.uris=localhost:9200
```

```
spring.elasticsearch.uris=i11a802.p.ssafy.io:9200
```

```
spring.data.elasticsearch.repositories.enabled=true
```

elasticsearch 설정 끝

spring.mvc.hiddenmethod.filter.enabled=true

Eureka 서버 설정입니다.

eureka.client.register-with-eureka=true

eureka.client.fetch-registry=true

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect