

E203 timing포팅 메뉴얼

프로젝트 기간 : 2023.10.10 ~ 2023.11.17(6주)

팀장 : 김정희

팀원 : 김규리, 신현탁, 이동규, 정영빈, 하성호

목차

[1. 사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정 값, 버전 기재 2](#_Toc147455660)

[2. 빌드 시 사용되는 환경 변수 등의 내용 3](#_Toc147455661)

[3. 배포 시 특이사항 9](#_Toc147455662)

[4. DB 접속 정보 12](#_Toc147455663)

# 사용한 JVM, 웹서버, WAS 제품 등의 종류와 설정 값, 버전 기재

1. Java

21

1. 웹서버

NginX

1. WAS

Tomcat

1. Gradle

8.3.x

1. Spring Boot

3.2.x

1. Node

18.17.0

1. React

18.2.0

1. Intellij

2019.3.5, 2023.1.5

1. Visual Studio Code

1.81.1

# 2. 빌드 시 사용되는 환경 변수 등의 내용

### Backend

- application.yml

|  |
| --- |
| spring:  profiles:  active: dev  group:  "local": "local"  "dev": "dev"  security:  oauth2:  client:  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  registration:  kakao:  client-id: ${KAKAO\_CLIENT\_ID}  client-secret: ${KAKAO\_CLIENT\_SECRET}  client-authentication-method: client\_secret\_post  authorization-grant-type: authorization\_code  client-name: kakao  scope:  - account\_email  - profile\_image  - profile\_nickname  servlet:  multipart:  max-file-size: 100MB  max-request-size: 100MB *# threads: # virtual: # enabled: true* springdoc:  packages-to-scan: com.kkukku.timing.apis  default-consumes-media-type: application/json;charset=UTF-8  default-produces-media-type: application/json;charset=UTF-8  swagger-ui:  disable-swagger-default-url: true  display-request-duration: true  operations-sorter: alpha  tags-sorter: alpha cloud:  aws:  credentials:  access-key: ${AWS\_S3\_ACCESS\_KEY}  secret-key: ${AWS\_S3\_SECRET\_KEY}  region:  static: ${AWS\_S3\_REGION}  s3:  bucket: ${AWS\_S3\_BUCKET}  url: ${AWS\_S3\_URL} application:  security:  jwt:  secret-key: ${JWT\_SECRET\_KEY}  access-token:  expiration: 900  refresh-token:  expiration: 604800  cors:  front-domain: ${FRONT\_DOMAIN\_URI}  back-domain: ${BACK\_DOMAIN\_URI}  management:  endpoint:  health:  show-details: *always* elasticsearch:  host: ${ELASTICSEARCH\_HOST}  external:  vision-ai:  url: ${VISION\_AI\_URL}  server:  tomcat:  max-swallow-size: 100MB  max-http-form-post-size: 100MB  max-http-response-header-size: 50KB  connection-timeout: 1800000  --- spring:  config:  activate:  on-profile: "local"  datasource:  driver-class-name: org.h2.Driver  url: 'jdbc:h2:mem:test'  h2:  console:  enabled: true  path: /h2-console  jpa:  database-platform: org.hibernate.dialect.H2Dialect  hibernate:  ddl-auto: update *# data.sql내용 유지, JPA Entity와 일치하도록 만듦* properties:  hibernate:  dialect: org.hibernate.dialect.H2Dialect  format\_sql: true  show\_sql: true  security:  oauth2:  client:  registration:  kakao:  redirect-uri: ${LOCAL\_KAKAO\_REDIRECT\_URI}  redis:  host: ${LOCAL\_REDIS\_HOST}  port: ${LOCAL\_REDIS\_PORT}  password: ${LOCAL\_REDIS\_PASSWORD}    ---  spring:  config:  activate:  on-profile: "dev"  datasource:  url: ${MARIADB\_URL}  username: ${MARIADB\_USERNAME}  password: ${MARIADB\_PASSWORD}  driver-class-name: org.mariadb.jdbc.Driver  jpa:  generate-ddl: false  show-sql: true  security:  oauth2:  client:  registration:  kakao:  redirect-uri: ${KAKAO\_REDIRECT\_URI}  redis:  host: ${REDIS\_HOST}  port: ${REDIS\_PORT}  password: ${REDIS\_PASSWORD}  *# 무중단 배포를 위한 profile 설정* --- spring:  config:  activate:  on-profile: set1  datasource:  url: ${MARIADB\_URL}  username: ${MARIADB\_USERNAME}  password: ${MARIADB\_PASSWORD}  driver-class-name: org.mariadb.jdbc.Driver  jpa:  generate-ddl: false  show-sql: false  security:  oauth2:  client:  registration:  kakao:  redirect-uri: ${KAKAO\_REDIRECT\_URI}  redis:  host: ${REDIS\_HOST}  port: ${REDIS\_PORT}  password: ${REDIS\_PASSWORD} server:  port: 8081  --- spring:  config:  activate:  on-profile: set2  datasource:  url: ${MARIADB\_URL}  username: ${MARIADB\_USERNAME}  password: ${MARIADB\_PASSWORD}  driver-class-name: org.mariadb.jdbc.Driver  jpa:  generate-ddl: false  show-sql: false  security:  oauth2:  client:  registration:  kakao:  redirect-uri: ${KAKAO\_REDIRECT\_URI}  redis:  host: ${REDIS\_HOST}  port: ${REDIS\_PORT}  password: ${REDIS\_PASSWORD} server:  port: 8082 |

- build.gralde

|  |
| --- |
| plugins **{** id 'java'  id 'org.springframework.boot' version '3.2.0-RC1'  id 'io.spring.dependency-management' version '1.1.3' **}** group = 'com.kkukku' version = '0.0.1-SNAPSHOT'  java **{** sourceCompatibility = '21' **}** configurations **{** compileOnly **{** extendsFrom annotationProcessor  **} }** repositories **{** mavenCentral()  maven **{** url 'https://repo.spring.io/milestone' **} }** dependencies **{** implementation 'org.springframework.boot:spring-boot-starter-actuator'  implementation 'org.springframework.boot:spring-boot-starter-data-jpa'   implementation 'org.springframework.boot:spring-boot-starter-data-redis'  implementation 'org.springframework.boot:spring-boot-starter-oauth2-client'  implementation 'org.springframework.boot:spring-boot-starter-security'  implementation 'org.springframework.boot:spring-boot-starter-web'  implementation 'io.jsonwebtoken:jjwt-api:0.12.3'  compileOnly 'org.projectlombok:lombok'  runtimeOnly 'io.jsonwebtoken:jjwt-impl:0.12.3'  runtimeOnly 'io.jsonwebtoken:jjwt-jackson:0.12.3'  annotationProcessor 'org.projectlombok:lombok'  testImplementation 'org.springframework.boot:spring-boot-starter-test'  testImplementation 'org.springframework.security:spring-security-test'   // Swagger  implementation 'org.springdoc:springdoc-openapi-starter-webmvc-ui:2.2.0'   // h2 for test  runtimeOnly 'com.h2database:h2'  implementation 'org.springframework.boot:spring-boot-starter-jdbc'   // database  implementation 'org.mariadb.jdbc:mariadb-java-client:3.2.0'   // aws  implementation 'com.amazonaws:aws-java-sdk-s3:1.12.573'   // junit  testImplementation group: 'org.junit.jupiter', name: 'junit-jupiter-api', version: '5.10.0'  testImplementation group: 'org.junit.jupiter', name: 'junit-jupiter-engine', version: '5.10.0'   // actuator  implementation 'org.springframework.boot:spring-boot-starter-actuator'   // ElasticSearch  implementation 'org.springframework.boot:spring-boot-starter-data-elasticsearch'  **}** tasks.named('test') **{** useJUnitPlatform() **}** |

# 3. 배포 시 특이사항

### Backend NginX설정

|  |
| --- |
| server {  listen 80 default\_server;  listen [::]:80 default\_server;  root /var/www/html;  index index.html index.htm index.nginx-debian.html;  server\_name timingkuku.shop www.timingkuku.shop;  include /etc/nginx/conf.d/service-url.inc;  location / {  proxy\_set\_header HOST $host;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  proxy\_set\_header X-Forwarded-Proto $scheme;  proxy\_set\_header Upgrade $http\_upgrade;  proxy\_set\_header Connection "upgrade";  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_buffer\_size 128k;  proxy\_buffers 4 256k;  proxy\_busy\_buffers\_size 256k;  proxy\_pass $service\_url;  proxy\_connect\_timeout 6000s;  proxy\_send\_timeout 6000s;  proxy\_read\_timeout 6000s;  }  listen [::]:443 ssl ipv6only=on; # managed by Certbot  listen 443 ssl; # managed by Certbot  ssl\_certificate /etc/letsencrypt/live/timingkuku.shop/fullchain.pem; # managed by Certbot  ssl\_certificate\_key /etc/letsencrypt/live/timingkuku.shop/privkey.pem; # managed by Certbot  include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot  ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot  } |

sudo service nginx start

### Frontend NginX설정

|  |
| --- |
| server {  listen 443 ssl default\_server;  listen [::]:443 ssl default\_server;  root /var/www/html;  index index.html index.htm index.nginx-debian.html;  server\_name timingkuku.store www.timingkuku.store; # managed by Certbot  location /objectDetection {  proxy\_buffer\_size 128k;  proxy\_buffers 4 256k;  proxy\_busy\_buffers\_size 256k;  proxy\_pass http://127.0.0.1:8000;  proxy\_connect\_timeout 6000s;  proxy\_send\_timeout 6000s;  proxy\_read\_timeout 6000s;  }  location / {  proxy\_pass http://127.0.0.1:3000;  proxy\_connect\_timeout 6000s;  proxy\_send\_timeout 6000s;  proxy\_read\_timeout 6000s;  }  ssl\_certificate /etc/letsencrypt/live/timingkuku.store/fullchain.pem; # managed by Certbot  ssl\_certificate\_key /etc/letsencrypt/live/timingkuku.store/privkey.pem; # managed by Certbot  include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot  ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot  } |

sudo service nginx start

# 4. DB 접속 정보

----MariaDB---

Host : ${MARIADB\_URL}

Port : 3306

User : ${MARIADB\_USERNAME}

Password : ${MARIADB\_PASSWORD}

Database : timing