ℹ️

1. 빌드 및 배포 정보

1-1. 개발 환경

OS

- Windows10
- Ubuntu 20.0.4

Frontend

- 개발툴
    - VisualStudio Code
- VS Code Extensions
    - eslint 2.4.2
    - prettier 10.1.0
- 프레임워크 / 라이브러리
    - React 5.0.1
    - node.js 18.17.1
    - npm 9.6.7
- 상태관리
    - recoil 0.7.7

▼ package.json

```
{
    "name": "minination",
    "version": "0.1.0",
    "private": true,
    "dependencies": {
        "@emotion/react": "^11.11.1",
        "@emotion/styled": "^11.11.0",
        "@mui/icons-material": "^5.14.8",
        "@mui/material": "^5.14.8",
        "@testing-library/jest-dom": "^5.17.0",
        "@testing-library/react": "^13.4.0",
        "@testing-library/user-event": "^13.5.0",
        "axios": "^1.5.0",
        "chart.js": "^4.4.0",
        "eslint-config-airbnb": "^19.0.4",
        "eslint-plugin-react-hooks": "^4.6.0",
        "lightweight-charts": "^4.0.1",
        "react": "^18.2.0",
        "react-chartjs-2": "^5.2.0",
        "react-dom": "^18.2.0",
        "react-router-dom": "^6.15.0",
        "react-scripts": "5.0.1",
        "react-transition-group": "^4.4.5",
        "react-uuid": "^2.0.0",
        "recoil": "^0.7.7",
        "recoil-persist": "^5.1.0",
        "sass": "^1.66.1",
        "sweetalert2": "^11.7.28",
        "web-vitals": "^2.1.4"
    },
    "scripts": {
        "start": "react-scripts start",
        "build": "react-scripts build",
        "test": "react-scripts test",
        "eject": "react-scripts eject"
    },
    "eslintConfig": {
        "extends": [
            "react-app",
            "react-app/jest"
        ]
    },
    "browserslist": {
        "production": [
            ">0.2%",
            "not dead",
            "not op_mini all"
        ],
        "development": [
            "last 1 chrome version",
            "last 1 firefox version",
            "last 1 safari version"
        ]
    },
    "devDependencies": {
        "eslint": "^8.48.0",
        "eslint-config-airbnb-base": "^15.0.0",
        "eslint-config-prettier": "^9.0.0",
        "eslint-plugin-import": "^2.28.1",
        "eslint-plugin-jsx-a11y": "^6.7.1",
        "eslint-plugin-prettier": "^5.0.0",
        "eslint-plugin-react": "^7.33.2",
        "prettier": "^3.0.3"
    }
}
```

Backend

- 개발 툴
  - Intellij 2023.1.3
  - Apache Tomcat
- 언어 / 프레임워크
  - Java 11
  - Spring boot 2.7.15
- 빌드 툴
  - gradle 8.2.1
- Database
  - mariaDB 11.1.2

▼ dependency

  - aop
  - jwt 0.11.1
  - redis
  - lombok
  - mapper 1.4.2
  - swagger2 2.9.2
  - QueryDsl 5.0.0
  - OpenFeign 3.1.5

Infra

- AWS EC2
  - Ubuntu 20.0.4
- Docker 24.0.6
- Jenkins

기타

- 이슈 관리
  - Jira

- 형상 관리
    - GitLab
- UI/UX툴
    - Figma
- 협업 툴
    - Mattermost
    - Notion
- API 테스트
    - Postman
    - Swagger2

1-2. 빌드 시 사용되는 환경 변수

Frontend 소스코드 실행 방법

1. Node.js 사이트 접속해서 LTS 설치
2. 레포지토리에서 소스코드 다운로드

    ```
    >> git clone https://lab.ssafy.com/s09-fintech-finance-sub2/S09P22A508.git
    ```

3. 프론트엔드 폴더로 이동 및 Node.js와 npm 버전 확인

    ```
    >> cd S09P22A508/Frontend
    >> node -v
    v18.17.1
    >> npm -v
    9.6.7
    ```

4. 필요한 라이브러리 설치

    ```
    >> npm i
    ```

5. 실행

    ```
    >> npm start
    ```

Backend

1. 버전에 맞는 자바(JDK 11) 다운로드
2. Backend/src/main/resources 경로에 아래 파일 붙여 넣기

▼ application.yml

```
spring:
  profiles:
    active: dev # local, dev 중 선택

  mvc: # Swagger 설정
    pathmatch:
      matching-strategy: ant_path_matcher

server:
  port: 8080 # 구동 포트
  error: # error code: Exception handler 처리
    include-exception: false
    include-message: always
    include-stacktrace: on_param
    whitelabel:
      enabled: true

springfox: # Swagger: Restful Api 문서 자동화 도구
documentation:
  swagger:
    use-model-v3: false
```

▼ application-dev.yml

Database와 Redis는 같은 Docker 네트워크에 있어서 컨테이너 이름으로 호출

```
spring:
  config:
    activate:
      on-profile: dev

  datasource:
    hikari:
      driver-class-name: com.mysql.cj.jdbc.Driver
      username: ssafy
      password: ssafya508
    url: jdbc:mysql://miniNation-db:3306/mini_nation?serverTimezone=Asia/Seoul&useUniCode=yes&characterEncoding=UTF-8

  jpa: # jpa: Spring Data JPA 설정
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        format_sql: true
    show_sql: true
  redis:
    host: redis
    port: 6379
    password: 'minia5085612'

# 로깅 레벨 설정
logging:
  level:
    com.ssafy.mini: info

jwt:
  secret: eyJzdWIiOiJtaW5pbmF0aW9uIiwibmFtZSI6Im1pbmluYXRpb24iLCJpYXQiOjE1MTYyMzkwMjJ9
  accessTokenValidity: 1.08e+7 # 3시간
  refreshTokenValidity: 8.64e+7 # 1일

# aws 설정
cloud:
  aws:
    credentials:
      accessKey:  AKIAVUHDOQKKTTCBGCVP
      secretKey:  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
    s3:
      bucket: mination
    region:
      static: ap-southeast-2
    stack:
      auto: false

# openFeign 설정
feign:
    client:
      config:
        default:
            connectTimeout: 5000
            readTimeout: 5000
            loggerLevel: full
    info:
      url: https://apis.data.go.kr/1160100/service/GetStockSecuritiesInfoService/
      key: fGQVEj1xcBzD14BH6qowYpu86OG3fgTswem6xQd%2B9xbkEN%2FoCYbF%2FUhp20zZK7yNRghHmq%2B4YxG%2BJASZlO65%2BA%3D%3D
```

3. BackendApplication 실행

1-3. 배포 시 특이사항

- letsencrypt를 통한 ssl 설정
- nginx를 통한 리버스 프록시 설정 (ec2 /etc/nginx/conf.d/default.conf)

```
upstream client {
    server 0.0.0.0:3000;
}
```

```
upstream server {
    server 0.0.0.0:8080;
}

server {

    listen 80;
    server_name 3.34.53.16 j9a508.p.ssafy.io minination.site www.minination.site;

    location / {
        return 301 https://minination.site$request_uri;
    }
}

server {
        listen 443 ssl; # managed by Certbot
        server_name minination.site www.minination.site;

        ssl_certificate /etc/letsencrypt/live/minination.site/fullchain.pem; # managed by Certbot
        ssl_certificate_key /etc/letsencrypt/live/minination.site/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

        location /api{
                rewrite ^/api(/.*)$ $1 break;
                proxy_pass http://server;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                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;
        }

        location / {
                proxy_pass http://client;
                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;
        }

}
```

- docker-compose를 통한 컨테이너 다중 실행
- Jenkins pipeline을 통한 CI/CD
- ufw-docker를 통한 방화벽 설정

1-4. 계정 및 프로퍼티 목록

Frontend

- .env 없음

Backend

▼ application.yml

```
spring:
  profiles:
    active: dev # local, dev 중 선택

  mvc: # Swagger 설정
    pathmatch:
      matching-strategy: ant_path_matcher

server:
  port: 8080 # 구동 포트
  error: # error code: Exception handler 처리
    include-exception: false
    include-message: always
    include-stacktrace: on_param
    whitelabel:
      enabled: true

springfox: # Swagger: Restful Api 문서 자동화 도구
documentation:
  swagger:
    use-model-v3: false
```

▼ application-dev.yml

```
spring:
  config:
    activate:
      on-profile: dev

  datasource:
    hikari:
      driver-class-name: com.mysql.cj.jdbc.Driver
      username: ssafy
      password: ssafya508
    url: jdbc:mysql://miniNation-db:3306/mini_nation?serverTimezone=Asia/Seoul&useUniCode=yes&characterEncoding=UTF-8

  jpa: # jpa: Spring Data JPA 설정
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        format_sql: true
    show_sql: true

  redis:
    host: redis
    port: 6379
    password: 'minia5085612'

# 로깅 레벨 설정
logging:
  level:
    com.ssafy.mini: info

jwt:
  secret: eyJzdWIiOiJtaW5pbmF0aW9uIiwibmFtZSI6Im1pbmluYXRpb24iLCJpYXQiOjE1MTYyMzkwMjJ9
  accessTokenValidity: 1.08e+7 # 3시간
  refreshTokenValidity: 8.64e+7 # 1일

# aws 설정
cloud:
  aws:
    credentials:
      accessKey:  AKIAVUHDOQKKTTCBGCVP
      secretKey:  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
    s3:
      bucket: minination
    region:
      static: ap-southeast-2
    stack:
      auto: false

# openFeign 설정
feign:
    client:
      config:
        default:
            connectTimeout: 5000
            readTimeout: 5000
            loggerLevel: full
    info:
      url: https://apis.data.go.kr/1160100/service/GetStockSecuritiesInfoService/
      key: fGQVEj1xcBzD14BH6qowYpu86OG3fgTswem6xQd%2B9xbkEN%2FoCYbF%2FUhp20zZK7yNRghHmq%2B4YxG%2
BJASZlO65%2BA%3D%3D
```

- DB
    - public IP: 3.34.53.16
    - id: ssafy
    - password: ssafya508
- application.yml
    - AWS EC2
    - AWS S3
    - mariaDB
    - redis

    - openFeign