빌드, 배포 및 외부 서비스 문서

1. 프로젝트 정보

1-1. 개발 환경

Backend

- **Java** : 17
- **Spring Boot** : 3.2.3
- **JPA** : hibernate-core:6.4.4
- **DB** : MySQL 8.0
- **Intellij** : 2023.3.2

Frontend

- **Node.js** : 21.7.1
- **TypeScript**: 4.9.5
- **React**: 18.2.0
- **Recoil**: 0.7.7
- **Axios**: 1.6.8
- **Vscode**: 1.85

Server

- AWS EC2
  - **OS** : Ubuntu
- AWS S3
  - 2.2.6.RELEASE

Service

- **Nginx** : 1.25.4
- **Jenkins** : 2.449
- **Docker** : 25.0.4

1-2. 사용 도구

- **이슈 관리** : JIRA
- **형상 관리** : Gitlab
- **커뮤니케이션** : Notion, Mattermost
- **디자인** : Figma
- **UCC** : VLLO
- **CI/CD** : Jenkins

1-3. 외부 서비스

- Amazon S3
- 환율 API

1-4. gitignore 정보

- Spring
  - application.yml, application-secret.properties, .env
  - 2-1 중 Spring에 파일 첨부해둠
- React
  - .env
  - 2-1 중 React에 파일 첨부해둠

---

2. 빌드

2-1. 빌드 시 필요한 파일

Spring

1. application.yml

- 구분 : 설정파일
- 위치 : S10P22B205\back\src\main\resources\application.yml

application.yml

```
#default
server:
```

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

spring:
  profiles:
    active: prod
    include: secret
    group:
      "local" : "local"
      "prod" : "prod"

cloud:
  aws:
    s3:
      bucket: hico-books
    credentials:
      access-key: AKIA2UC3ACYKD26NTOAY
      secret-key: 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
    region:
      static: ap-northeast-2
      auto: false
    stack:
      auto: false

logging:
  level:
    org.hibernate.SQL: debug
    org.hibernate.type: trace

---
spring:
  config:
    activate:
      on-profile: local
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://127.0.0.1:3306/hico
```

```
    username: hico
    password: hicohico52
  jpa:
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        format_sql: true
        default_batch_fetch_size: 10
  sql:
    init:
      data-locations: classpath*:db/data.sql
      mode: always
      platform: mysql
---
spring:
  config:
    activate:
      on-profile: prod

  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://mysql-hico:3306/hico?useSSL=false&all
owPublicKeyRetrieval=true&serverTimezone=Asia/Seoul&charact
erEncoding=UTF-8
    username: hico
    password: hicohico52

  jpa:
    database: mysql
    hibernate:
      ddl-auto: none
    properties:
      hibernate:
        format_sql: true
        default_batch_fetch_size: 10
    defer-datasource-initialization: true
```

2. application-secret.properties

- 구분 : 환경변수
- 위치 : S10P22B205\back\src\main\resources\application-secret.properties

application-secret.properties

```
ssafy.bank.manager-id=cucumber@ssafy.co.kr
ssafy.bank.api-key=4b8281517aae4fdb80a4e58fc0d8c90a
ssafy.bank.account-type-unique-no=001-1-81fe2deafd1943
ssafy.bank.institution-code=00100
ssafy.bank.fintech-app-no=001
ssafy.bank.bank-name=\uD55C\uAD6D\uC740\uD589

#jwt
jwt.secret-key=Y3VjdW1iZXIgaXMgdmVyeSBjdXRlIGN1Y3VtYmVyIGZv
cmV2ZXIgY3VjdW1iZXIgaXMgdmVyeSBjdXRlIGN1Y3VtYmVy

##currency
currency.secret-key=0j4LxTQLjeIZ56itctsOy118aX5MM80g
```

3. .env

- 구분 : 환경변수
- 위치 : S10P22B205\.env (docker-compose.yml와 같은 위치)

.env

```
MYSQL_ROOT_PASSWORD: hicohico52
MYSQL_DATABASE: hico
```

```
MYSQL_USER : hico
MYSQL_PASSWORD : hicohico52
```

React

- .env.development
- 위치 : S10P22B205\front\.env.development

```
REACT_APP_SERVER_URI=http://localhost:8081/api
DANGEROUSLY_DISABLE_HOST_CHECK=true
```

2-2. 빌드하기

Backend

- build.gradle 실행

Frontend

- npm i
- npm start / npm run build

---

3. 배포

Linux 기준

3.1 배포 순서

0. 사전 준비

- Git 설치
- Docker 설치

1. Git clone

- url: 'https://lab.ssafy.com/s10-fintech-finance-sub2/S10P22B205.git'

2. 빌드 시 필요한 파일 추가

- “2-1. 빌드 시 필요한 파일” 에 있는 파일들 다운로드 후 해당 위치에 기재

3. 빌드

- Backend 빌드
- Frontend 빌드

4. docker-compose

- 프로젝트 최상단에 위치한 docker-compose.yml 로 `docker compose up -d --build` 진행

5. nginx

- docker-compose.yml 파일을 통해 nginx 컨테이너 생성

```
version: '3.8'
services:
  nginx:
    image: nginx:latest
    container_name: nginx
    restart: always
    volumes:
      - /home/nginx/conf.d:/etc/nginx/conf.d
      - ./data/certbot/conf:/etc/letsencrypt
      - ./data/certbot/www:/var/www/certbot
    ports:
      - 80:80
      - 443:443
  certbot:
    image: certbot/certbot:latest
    container_name: certbot
```

```
    #restart: unless-stopped
    command: certonly --webroot --webroot-path=/var/www/cer
tbot --email tndms0308@gmail.com --agree-tos --no-eff-email
-d j10b205.p.ssafy.io
    volumes:
      - ./data/certbot/conf:/etc/letsencrypt
      - ./data/certbot/www:/var/www/certbot
      - ./data/certbot/logs:/var/log/letsencrypt

networks:
    default:
            name: hico_net
            external: true
```

- nginx 설정 (conf)

```
server {
    listen       80;
    listen  [::]:80;
    server_name  j10b205.p.ssafy.com www.j10b205.p.ssafy.co
m;

    location / {
        return 301 https://$host$request_uri;
    }
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}

server {
    listen 443 ssl;
    server_name j10b205.p.ssafy.io;
    server_tokens off;

    ssl_certificate /etc/letsencrypt/live/j10b205.p.ssafy.i
```

```
o/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/j10b205.p.ssa
fy.io/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;

    location / {
        proxy_pass http://frontend:3000;
        proxy_set_header    Host                $http_host;
        proxy_set_header    X-Real-IP           $remote_add
r;
        proxy_set_header    X-Forwarded-For     $proxy_add_
x_forwarded_for;
    }

    location /api {
        proxy_pass http://backend:8081;
        proxy_set_header    Host                $http_host;
        proxy_set_header    X-Real-IP           $remote_add
r;
        proxy_set_header    X-Forwarded-For     $proxy_add_
x_forwarded_for;
    }
}
```

---

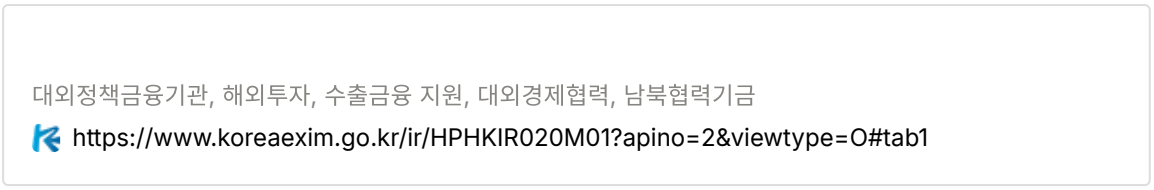
4. 외부 서비스 이용 방법

4.1 AWS S3

1. ‘hico-books’로 S3 버킷을 생성한다.
2. ‘hico-books’ 버킷 안에 books 폴더를 생성한다.
3. application.yml에 본인의 access-key와 secret-key를 입력한다.
4. 만화 이미지를 S3 버킷에 업로드하기 위해 포스트맨에서 http://localhost:8081/api/book/start 를 호출한다.

4.2 환율 API

1. 한국수출입은행에서 인증키 발급을 신청한다.



2. application-secret.properties의 currency.secret-key에 발급받은 인증키를 입력한다.
3. 최근 한 달 환율을 가져오기 위해 포스트맨에서 http://localhost:8081/api/rate/start를 호출한다.