Porting Manual

1. 사용 도구

- 이슈 관리 : Jira
- 형상 관리 : GitLab
- 커뮤니케이션 : Notion, MatterMost
- 디자인 : Figma
- CI/CD : GitLab CICD(GitLab Runner)

2. 개발 도구

- Visual Studio Code : 1.85.1
- Intellij : 2023.3.2 (Ultimate Edition)

3. 개발 환경

Frontend

Node.js	`20.11.0`
React	`18.2.0`
zustand	`4.5.2`
Axios	`1.6.7`
yarn	`1.22.21`
react-query	`5.28.4`

Backend

Java	`openjdk21.0.1( zulu 21.30.15 )`
Spring Boot	`3.2.1`

Server

AWS S3	Free Tier
AWS EC2	CPU : 4코어 RAM: 16GB 볼륨:320GB(SSD) ,6TB *2 2

Service

Docker	`25.0.1`
Ubuntu	`"20.04.6 LTS (Focal Fossa)”`

4. 환경변수 형태

Frontend

- package.json

```
{
  "name": "front",
  "private": true,
  "version": "0.0.0",
```

```
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "tsc && vite build",
    "lint": "eslint . --ext ts,tsx --report-unused-disable
    "preview": "vite preview"
  },
  "dependencies": {
    "@tanstack/react-query": "^5.28.4",
    "@types/react-query": "^1.2.9",
    "axios": "^1.6.7",
    "moment": "^2.30.1",
    "node-sass": "^9.0.0",
    "react": "^18.2.0",
    "react-calendar": "^4.8.0",
    "react-dom": "^18.2.0",
    "react-router-dom": "^6.22.3",
    "zustand": "^4.5.2"
  },
  "devDependencies": {
    "@types/react": "^18.2.64",
    "@types/react-dom": "^18.2.21",
    "@typescript-eslint/eslint-plugin": "^6.4.0",
    "@typescript-eslint/parser": "^7.1.1",
    "@vitejs/plugin-react": "^4.2.1",
    "eslint": "^8.0.1",
    "eslint-config-prettier": "^9.1.0",
    "eslint-config-standard-with-typescript": "^43.0.1",
    "eslint-plugin-import": "^2.25.2",
    "eslint-plugin-n": "^15.0.0 || ^16.0.0 ",
    "eslint-plugin-prettier": "^5.1.3",
    "eslint-plugin-promise": "^6.0.0",
    "eslint-plugin-react": "^7.34.0",
    "eslint-plugin-react-hooks": "^4.6.0",
    "eslint-plugin-react-refresh": "^0.4.5",
    "prettier": "^3.2.5",
    "sass": "^1.71.1",
    "typescript": "*",
```

```
    "vite": "^5.1.6"
  },
  "main": "index.js",
  "license": "MIT"
}
```

- vite.config.js

```
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';
import path from 'path';

// https://vitejs.dev/config/
export default defineConfig({
  plugins: [react()],
  // 절대 경로 추가
  resolve: {
    alias: [
      { find: '@', replacement: path.resolve(__dirname, 's
      // { find: '@pages', replacement: path.resolve(__dir
    ],
  },
});
```

Backend

- application.yml

```
spring:
  config:
    import:
      - secret.yml
  profiles:
    default: local
```

```
logging:
  level:
    com.spring.mmm: debug
```

- application-local.yml

```
spring:
  jpa:
    hibernate:
      ddl-auto: create-drop
    properties:
      hibernate:
        show_sql: true
        format_sql: true
  h2:
    console:
      enabled: true
  redis:
    port: 6379
```

- application-dev.yml

```
spring:
  jpa:
    hibernate:
      ddl-auto: validate
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MariaDBDialect
  data:
    redis:
      host: j10a110a.p.ssafy.io
      port: 6379
      timeout: 6
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    username: ssafy
    password: 1q2w3e4r!
```

```
    url: jdbc:mariadb://j10a110a.p.ssafy.io:3306/mmm?useSS
 server:
   tomcat:
     mbeanregistry:
       enabled: true
 logging:
   level:
     com.spring.mmm: debug
 management:
   endpoints:
     web:
       exposure:
         include: "*"
```

- application-prod.yml

```
spring:
  jpa:
    hibernate:
      ddl-auto: update
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MariaDBDialect
  data:
    redis:
      host: j10a110.p.ssafy.io
      port: 6379
      timeout: 6
  datasource:
    driver-class-name: org.mariadb.jdbc.Driver
    username: ssafy
    password: 1q2w3e4r!
    url: jdbc:mariadb://j10a110.p.ssafy.io:3306/mmm?useSSL=fa
server:
  tomcat:
    mbeanregistry:
      enabled: true
management:
```

```
  endpoints:
    web:
      exposure:
        include: "*"
```

- secret.yml

```
cloud:
  aws:
    s3:
      bucket: <bukect 이름>
    stack:
      auto: false
    region:
      static: < 리전 이름 >
    credentials:
      access-key: < 엑세스 토큰 값 >
      secret-key: < 엑세스 토큰 값 >

jwt:
  secret: qlalfqjsghfmfanjfhgodigkfwlahfmrptsmsrjfgkgkgkgkgkg

spring:
  mail:
    host: smtp.gmail.com
    port: 587
    username: < 계정 이름 >
    password: < 비밀번호 >
api:
  service-key: < 기상청 API 서비스 키>
```

---

5. 배포 관련

Gitlab CI/CD

- 깃랩 CICD 동작 파일

```
stages:          # List of stages for jobs, and their order o
  - test
  - build
  - deploy
image: gradle:alpine

test-backend:       # This job runs in the build stage, which
  image: gradle:jdk21-alpine
  stage: test
  script:
    - cd backend
    - chmod +x gradlew
    - ./gradlew test
  rules:
    - if: $CI_PIPELINE_SOURCE == 'merge_request_event' && $CI

test-frontend:       # This job runs in the build stage, whic
  image: node:18-alpine
  stage: test
  script:
    - cd frontend
    - rm -rf node_modules/
    - npm install --legacy-peer-deps
    - yarn build
  rules:
    - if: $CI_PIPELINE_SOURCE == 'merge_request_event' && $CI

build-backend:       # This job runs in the build stage, whic
  image: docker:stable
  stage: build
```

```
  services:
    - name: docker:20-dind
      alias: localhost
      command: ['--tls=false']
  variables:    # activate container-to-container networking
    CPU: 1024
    MEMORY: 2048
    DOCKER_HOST: tcp://localhost:2375
    DOCKER_DRIVER: overlay2
    DOCKER_TLS_CERTDIR: ''
  script:
    - cd backend
    - docker build -t $DOCKER_HUB_ID/mmm-backend-dev .
    - docker login -u $DOCKER_HUB_ID -p $DOCKER_HUB_PW
    - docker push $DOCKER_HUB_ID/mmm-backend-dev
  after_script:
    - docker logout
  rules:
    - if: $CI_PIPELINE_SOURCE == 'push' && $CI_COMMIT_BRANCH =

build-frontend:       # This job runs in the build stage, whi
  image: docker:stable
  stage: build
  services:
    - name: docker:20-dind
      alias: localhost
      command: ['--tls=false']
  variables:    # activate container-to-container networking
    CPU: 1024
    MEMORY: 2048
    DOCKER_HOST: tcp://localhost:2375
    DOCKER_DRIVER: overlay2
    DOCKER_TLS_CERTDIR: ''
  script:
    - cd frontend
    - rm -rf node_modules/
    - docker build -t $DOCKER_HUB_ID/mmm-frontend-dev .
    - docker login -u $DOCKER_HUB_ID -p $DOCKER_HUB_PW
```

```
    - docker push $DOCKER_HUB_ID/mmm-frontend-dev
  after_script:
    - docker logout
  rules:
    - if: $CI_PIPELINE_SOURCE == 'push' && $CI_COMMIT_BRANCH =

deploy-backend:       # This job runs in the build stage, whi
  image: alpine:latest
  stage: deploy
  script:
    - 'command -v ssh-agent >/dev/null || ( apk update  && ap
    - eval $(ssh-agent -s)
    - chmod 400 "$SSH_PRIVATE_KEY"
    - ssh-add "$SSH_PRIVATE_KEY"
    - mkdir -p ~/.ssh
    - chmod 700 ~/.ssh
    - ssh-keyscan -H $DEVELOP_SERVER_ADDRESS >> ~/.ssh/known_
    - ssh $REMOTE_USERNAME@$DEVELOP_SERVER_ADDRESS 'sh be.sh'
  needs: [build-backend]
  only:
    - be



deploy-frontend:       # This job runs in the build stage, wh
  image: alpine:latest
  stage: deploy
  script:
    - 'command -v ssh-agent >/dev/null || ( apk update  && ap
    - eval $(ssh-agent -s)
    - chmod 400 "$SSH_PRIVATE_KEY"
    - ssh-add "$SSH_PRIVATE_KEY"
    - mkdir -p ~/.ssh
    - chmod 700 ~/.ssh
    - ssh-keyscan -H $DEVELOP_SERVER_ADDRESS >> ~/.ssh/known_
    - ssh $REMOTE_USERNAME@$DEVELOP_SERVER_ADDRESS 'sh fe.sh'
  needs: [build-frontend]
  only:
```

```
    - fe
```

- 깃랩 러너 도커-컴포즈 파일

```
version: '3.9'
services:
  gitlab-runner:
    container_name: gitlab-runner
    image: gitlab/gitlab-runner:latest
    restart: always
    volumes:
      - ./config:/etc/gitlab-runner
      - /var/run/docker.sock:/var/run/docker.sock
```

원격 서버에서 이 도커 컴포즈 파일을 실행시켜야 합니다.

Backend Dockerfile

이 파일은 백엔드의 도커 파일입니다.

프로필에 따라서 시작 명령어가 달라집니다.

dev 프로필의 경우 `docker run -d -p 8080:8080 --name backend kgh2120/mmm-backend-dev` 이렇게 실행시키고

prod 프로필의 경우

`docker run -d -p 8080:8080 --name backend -e ENV_PROFILE=prod  kgh2120/mmm-backend-dev`

- Dockerfile

```
FROM gradle:jdk21-alpine as builder
WORKDIR /build

# 그래들 파일이 변경되었을 때만 새롭게 의존패키지 다운로드 받게함.
COPY build.gradle settings.gradle /build/
RUN gradle build -x test --parallel --continue > /dev/null

# 빌더 이미지에서 애플리케이션 빌드
COPY . /build
RUN gradle build -x test --parallel

# APP
FROM openjdk:21-slim
WORKDIR /app

# 빌더 이미지에서 jar 파일만 복사
COPY --from=builder /build/build/libs/*-SNAPSHOT.jar ./app

ENV ENV_PROFILE=dev

EXPOSE 8080

ENTRYPOINT [
    "java",
    "-jar",
    "-Dspring.profiles.active=${ENV_PROFILE:-dev}",
    "app.jar"
]
```

Nginx Conf

- DEV Server Nginx

```
#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_loc
                      '$status $body_bytes_sent "$http_refere
                      '"$http_user_agent" "$http_x_forwarded_

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

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    #gzip  on;

    server {
        listen       80;
        listen  [::]:80;
        server_name  j10a110a.p.ssafy.io;
        client_max_body_size 100M;
        location / {
```

```
            return 301 https://$host$request_uri;
        }

        location /.well-known/pki-validation/ {
            alias /usr/share/nginx/html/.well-known/pki-valid
        }
    }

    server {

        listen  443 ssl;
        server_name j10a110a.p.ssafy.io;
        client_max_body_size 100M;

        ssl_certificate /etc/nginx/ssl/certificate.crt;
        ssl_certificate_key /etc/nginx/ssl/private.key;

         location / {
                proxy_pass http://localhost:8090/;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_http_version 1.1;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x
                proxy_set_header X-Forwarded-Proto $scheme;
        }
        location /api/ {
                proxy_pass http://localhost:8080/;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_http_version 1.1;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x
                proxy_set_header X-Forwarded-Proto $scheme;
        }
```

```
        location /pro/ {
                proxy_pass http://localhost:8081/;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_http_version 1.1;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x
                proxy_set_header X-Forwarded-Proto $scheme;
        }
        location /dash/ {
            proxy_pass http://localhost:8082/;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection "upgrade";
            proxy_http_version 1.1;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_for
            proxy_set_header X-Forwarded-Proto $scheme;
        }

        #error_page  404              /404.html;

        # redirect server error pages to the static page /50x
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   /usr/share/nginx/html;
        }
    }



}
```

- PROD Server Nginx

```
#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_loc
                      '$status $body_bytes_sent "$http_refere
                      '"$http_user_agent" "$http_x_forwarded_

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

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    #gzip  on;

    upstream backend {
        server localhost:8080;
        server localhost:8081;
    }
    server {
        listen       80;
        listen  [::]:80;
        server_name  j10a110.p.ssafy.io;
        client_max_body_size 100M;
```

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

        location /.well-known/pki-validation/ {
            alias /usr/share/nginx/html/.well-known/pki-valid
        }
    }

    server {
        listen  443 ssl;
        server_name j10a110.p.ssafy.io;

        ssl_certificate /etc/nginx/ssl/certificate.crt;
        ssl_certificate_key /etc/nginx/ssl/private.key;
        client_max_body_size 100M;

         location / {
                proxy_pass http://localhost:8090/;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_http_version 1.1;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x
                proxy_set_header X-Forwarded-Proto $scheme;
        }

        location /api/ {
                proxy_pass http://backend;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                proxy_http_version 1.1;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x
                proxy_set_header X-Forwarded-Proto $scheme;
        }
```

```
        #error_page  404              /404.html;

        # redirect server error pages to the static page /50x
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   /usr/share/nginx/html;
        }
    }
}
```