포팅 메뉴얼

환경

1. BackEnd
   - openjdk 17.0.7
   - spring boot 3.1.2
   - mysql 8.0.34
   - redis 5.0.7
   - openvidu 2.28.0
   - intellij 2023.2
2. FrontEnd
   - node 18.16.1
   - vue 3.3.4
   - openai 3.3.0
   - vscode 1.79.2

빌드

BackEnd

1. 아래 application.yml 파일을 .gitignore 해두었기에 `backend/src/main/resources` 경로에 추가
2. Gradle 실행
3. Bootjar 실행

```
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://54.180.9.43:3306/ssap?serverTimezone=Asia/Seoul&characterEncoding=UTF-8
    username: ssap
    password: ssap

  jpa:
    hibernate:
      ddl-auto: create
    properties:
      hibernate:
        format_sql: true
        hbm2ddl:
          import_files: 'classpath:static-data.sql'

  mail:
    host: smtp.gmail.com
    port: 587
    username: {your_mail_id}
    password: {your_mail_password}
```

```
    properties:
      mail:
        smtp:
          auth: true
          timeout: 5000
          starttls:
            enable: true

  data:
    redis:
      host: 54.180.9.43
      port: 6379


  #s3관련 파일 크기 제한
  servlet:
    multipart:
      max-file-size: 2MB
      max-request-size: 2MB

jwt:
  header: Authorization
  secret: {your_secret}
  token-validity-in-seconds: 86400

# s3관련.
cloud:
  aws:
    s3:
      bucket: ssapbucket #버킷이름
    stack.auto: false #AWS의 자동 CloudFormation구성
    region.static: ap-northeast-2 #한국
    credentials:
      accessKey: {your_accessKey}
      secretKey: {your_accessKey}

# openvidu
openvidu:
  server:
    url: https://i9d205.p.ssafy.io:8442/
    secret: ssapssap
  local:
    url: http://localhost:4443/
    secret: MY_SCRET

logging:
  level:
    org.hibernate.sql: debug
    org.hibernate.orm.jdbc.bind: trace
    com.ssafy.ssap.controller.RoomController: trace
    com.ssafy.ssap.service.RoomService: trace
```

주의 사항

- aws s3 관련 속성은 과금이 발생할 수 있으므로 키 관리 주의
- 이메일 인증을 위한 smtp 사용을 위해서는 이메일 앱 비밀번호를 발급 받아야 한다

FrontEnd

1. npm run build

주의사항

- chatGPT 기능을 이용하기 위해서는 openAI API에서 개인 API key를 발급받아야함

배포 특이사항

✔️ Jenkins 내에 jdk 버전 17을 직접 설치해주고 설정해야 함

1. Jenkins 컨테이너 안에 접속
   `docker exec -it jenkins-docker_jenkins_1 bash`
2. `apt-get update`
3. `apt-get install openjdk-17-jdk -y`
4. `export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64`
5. `echo $JAVA_HOME`
6. Jenkins는 아직 JAVA_HOME이 11을 가리키고 있으므로 설정 변경
   a. Jenkins 관리
   b. Global Tool Configuration
   c. Install automatically 체크 해제
   d. Name : openjdk-17-jdk
   e. JAVA_HOME : /usr/lib/jvm/java-17-openjdk-amd64

✔️ Jenkins 내에 node 설치

1. plugin에서 node 설치
2. global tools 설정
3. 빌드 정보 수정에서 Provide Node & npm bin/ folder to PATH 클릭

nginx 설정

/etc/nginx/sites-available/default

```
server {
  root /var/www/html;

  location / {
          try_files $uri $uri/ /index.html;
          add_header 'Access-Control-Allow-Origin' '*';
          add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT, DELETE';
          add_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Rang
          add_header 'Access-Control-Expose-Headers' 'Content-Length,Content-Range';
  }

  location /api {
          rewrite ^/api(/.*)$ $1 break;
          proxy_pass http://localhost:8080;
          proxy_http_version 1.1;
          proxy_set_header Upgrade $http_upgrade;
          proxy_set_header Connection 'upgrade';
          proxy_set_header Host $host;
          proxy_cache_bypass $http_upgrade;
  }

  listen [::]:443 ssl ipv6only=on; # managed by Certbot
  listen 443 ssl; # managed by Certbot
  ssl_certificate /etc/letsencrypt/live/i9d205.p.ssafy.io/fullchain.pem; # managed by Certbot
  ssl_certificate_key /etc/letsencrypt/live/i9d205.p.ssafy.io/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
}
```

Mysql

```
sudo apt-get install mysql-server
sudo systemctl start mysql
sudo systemctl enable mysql
sudo /usr/bin/mysql -u root -p
CREATE DATABASE ssap;
CREATE USER 'ssap'@'%' IDENTIFIED BY 'ssap';
```

```
GRANT ALL PRIVILEGES ON ssap.* FOR 'ssap'@'*';
FLUSH PRIVILEGES;
```

Redis

```
sudo apt-get install redis-server
sudo systemctl start redis-server start
sudo systemctl enable redis-server.service
redis-cli
keys *
exit
```