|  |
| --- |
|  |

|  |
| --- |
| **Dancy** |
| **포팅 메뉴얼** |
|  |
|  |
|  |
|  |
|  |



목차

[I. 개요 2](#_Toc96072459)

[1. 프로젝트 개요 2](#_Toc96072460)

[2. 프로젝트 사용 도구 2](#_Toc96072461)

[3. 개발환경 2](#_Toc96072462)

[4. 외부 서비스 2](#_Toc96072463)

[II. 빌드 및 실행 3](#_Toc96072465)

[1. 환경변수 형태 3](#_Toc96072466)

[2. 빌드 및 실행하기 5](#_Toc96072467)

[3. 서비스 이용 방법 6](#_Toc96072468)

# 개요

## 프로젝트 개요

춤이 각광받는 요즘 시대에 춤 연습을 어떻게 하면 보다 편하게 할 수 있을까?

춤을 좋아하는 사람과의 소통은 어떻게 할 수 있을까?

Dancy는 그러한 니즈를 충족시켜주기 위해 탄생한 댄스 연습 보조 SNS입니다. Dancy와 함께라면, 춤도 쉽게 연습할 수 있습니다!

## 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost

디자인 : Figma

UCC : 프리미어 프로

CI/CD : Jenkins

## 개발환경

- IntelliJ

- JVM : 17

- Springboot : 3.2.2

- Gradle : 8.5

- DB : MariaDB 10.11 LTS

- Server : AWS EC2

- Redis

## 외부 서비스

Google Mediapipe Pose:

AWS S3 : python/.s3connect.py 해당내용 있음,

Opencv :

(과금이 발생할 수 있는 키입니다. 취급 주의)

# 빌드 및 실행

## 환경변수 형태

1) application.properties

spring.datasource.url=jdbc:mariadb://{db\_url}/dancy

spring.datasource.username={db\_username}

spring.datasource.password={db\_password}

spring.jpa.hibernate.ddl-auto=update

spring.data.redis.host=localhost

spring.data.redis.port=6379

spring.data.redis.password={redis\_password}

#spring.data.redis.host=i10d210.p.ssafy.io

#spring.data.redis.port=6379

#spring.data.redis.password=ENC(dPdLBkuK9Py5f9J7gBWJpNCZLCjqBmgK)

jwt.secret.key={secret\_key}

springdoc.default-consumes-media-type=application/json;charset=UTF-8

springdoc.default-produces-media-type=application/json;charset=UTF-8

springdoc.swagger-ui.url=/docs/open-api-3.0.1.json

springdoc.swagger-ui.path=/docs/swagger

spring.servlet.multipart.max-file-size=100MB

spring.servlet.multipart.max-request-size=100MB

python.server=http://{python\_base\_url}/uploadVideo

server.servlet.context-path=/api

#logging.level.org.springframework=debug

#logging.level.org.springframework.web=debug

2) application-prod.properties

spring.datasource.url=jdbc:mariadb://{db\_url}/dancy

spring.datasource.username={db\_username}

spring.datasource.password={db\_password}

spring.jpa.hibernate.ddl-auto=update

spring.data.redis.host={redis\_host\_url}

spring.data.redis.port=6379

spring.data.redis.password={redis\_password}

jwt.secret.key={secret\_key}

springdoc.default-consumes-media-type=application/json;charset=UTF-8

springdoc.default-produces-media-type=application/json;charset=UTF-8

springdoc.swagger-ui.url=/docs/open-api-3.0.1.json

springdoc.swagger-ui.path=/docs/swagger

python.server=http://{python\_base\_url}/uploadVideo

server.servlet.context-path=/api

#logging.level.org.springframework=debug

#logging.level.org.springframework.web=debug

3) email.properties

spring.mail.host=smtp.gmail.com

spring.mail.port=587

spring.mail.username={real\_email\_in\_gmail}

spring.mail.password={real\_password\_registered\_by\_gmail\_smtp}

spring.mail.properties.mail.smtp.auth=true

spring.mail.properties.mail.smtp.starttls.enable=true

spring.mail.properties.mail.smtp.starttls.required=true

spring.mail.properties.mail.smtp.connectiontimeout=5000

spring.mail.properties.mail.smtp.timeout=5000

spring.mail.properties.mail.smtp.writetimeout=5000

spring.auth-code-expiration-millis=1800000 #30?

4) s3.properties

aws.s3.accessKey={s3\_access\_key}

aws.s3.secretKey={s3\_secret\_key}

aws.s3.region=ap-northeast-2

aws.s3.bucket = {s3\_your\_bucket}

aws.s3.commonUrl = {s3\_your\_commonUrl}

## 빌드 및 실행하기

1) 기본 환경

ec2에 docker, nginx, mariadb, redis 설치

2) FrontEnd - React

- ec2 에 client 폴더 복사

- docker build 후 run

docker build -t react .

docker run -d -p 5173:5173 --name react react

3) BackEnd - Spring

- ec2에 server 폴더 복사

- docker build 후 run

docker build -t spring .

docker run -d -p 8080:8080 --name spring spring

4) BackEnd - Python

- ec2에 python 폴더 복사

- docker build 후 run

docker build -t python .

docker run -d -p 5000:5000 --name python python

5) ssl 설정

- certbot 등을 이용하여 ssl 설정 후 /etc/nginx/nginx.conf 파일 수정

events {

worker\_connections 1024;

}

http {

server {

listen 443 ssl http2;

server\_name i10d210.p.ssafy.io;

ssl\_certificate /etc/letsencrypt/live/i10d210.p.ssafy.io/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/i10d210.p.ssafy.io/privkey.pem;

location / {

proxy\_pass http://localhost:5173;

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;

}

location /api/alarm/subscribe {

proxy\_pass http://localhost:8080;

proxy\_set\_header Connection '';

proxy\_set\_header Cache-Control 'no-cache';

proxy\_set\_header X-Accel-Buffering 'no';

proxy\_set\_header Content-Type 'text/event-stream';

proxy\_buffering off;

chunked\_transfer\_encoding on;

proxy\_read\_timeout 86400s;

}

location /api {

proxy\_pass http://localhost:8080;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection '';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

server {

listen 80;

server\_name i10d210.p.ssafy.io;

return 301 https://i10d210.p.ssafy.io$request\_uri;

}

##

# Basic Settings

##

client\_max\_body\_size 100M;

}

6) nginx 재시작 후 서버주소로 접속

sudo service nginx restart

## 서비스 이용 방법

## 1. 네브바 Practice 버튼 클릭

## 2. (좌측) 원본 안무 영상 선택 혹은 desktop에 저장된 영상 업로딩

## 3. (우측) 나의 안무 연습 영상 녹화본 업로딩

## 4. 좌측, 우측 모두 썸네일로 변경된 후 변환하기 버튼 클릭

## 5. 네브바 검색창 좌측에서 영상 변환 되는 동안 Loading창 확인 가능

## 6. Loading창이 Compelete로 변경되면 해당 버튼 클릭하여 Practice 페이지로 이동 가능

## 7. 합성된 영상의 평균 정확도 및 구간별 정확도 확인 가능

## 8. 우측 오차구간에서 유사도 낮은 구간 확인 가능

## 9. 각각의 구간 클릭 시 해당 구간 반복 재생

## 10. 연속 재생 버튼 클릭 시 전체 영상 연속 재생 가능