

CustomGPT 포팅 메뉴얼

목차

- 개요
 - 프로젝트 개요
 - 프로젝트 사용 도구
 - 개발 환경
 - 외부 서비스
- 빌드와 배포
 - 환경 변수
 - 빌드
 - 배포

개요

프로젝트 개요

chatGPT가 유행하며 수많은 기업에서 chatGPT를 이용하고 있습니다. 그러나 chatGPT를 이용하면서 기업의 기밀이 유출되는 사고가 계속해서 발생하고 있어 대기업들은 각자의 대형 언어 모델을 구축하여 이용하고 있는 상황입니다.

하지만 기존의 대형 언어 모델을 학습하기 위해서는 대량의 데이터와 컴퓨팅 자원이 필요합니다. 이러한 비용을 감당하지 못하는 중,소규모 기업들을 위하여 CustomGPT는 소형 언어 모델을 이용한 맞춤형 chatGPT서비스를 제공합니다.

프로젝트 사용 도구

- 이슈 관리: Jira
- 형상 관리: Gitlab
- 커뮤니케이션: Discord, Notion, MatterMost
- UCC: Movavi
- CI/CD: Jenkins

개발 환경

- IntelliJ: 2022.03 Ultimate Edition
- JVM: azul-11
- SpringBoot: 2.7.11
- Server: AWS EC2 t2.xlarge Ubuntu 20.04 5 LTS
- GPU Server: JupyterHub
- sLLM: ko_vicuna_7b

- DB:MySQL Ver 8.0.33-0ubuntu0.20.04.2 for Linux on x86_64 ((Ubuntu))

외부 서비스

- 도메인 구매: Gabia
- 호스팅: CloudFlare

빌드와 배포

환경변수

Port

- Spring: 8080
- React: 80/443

복호화 키

- JAVA_ENC=-Djasypt.encryptor.password=gjslqhtakwg!

Papago API 키

- Client_ID=_y0ncsEw7dDCxkmbNxY4
- PAPAGO_API_URL=<https://openapi.naver.com/v1/papago/n2mt>
- CLIENT_SECRET=pMyhoGUybp
- GPU_API_URL=<http://175.209.203.185:8000/chat>

빌드

DockerFile

```
FROM node:16.15.0 as build-stage
WORKDIR /var/jenkins_home/workspace/react
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
FROM nginx:stable-alpine as production-stage

COPY --from=build-stage /var/jenkins_home/workspace/react/build /usr/share/nginx/html
COPY --from=build-stage /var/jenkins_home/workspace/react/nginx_conf/nginx.conf /etc/nginx/conf.d/default.conf

EXPOSE 80
EXPOSE 443
CMD ["nginx", "-g","daemon off;"]
```

```
FROM adoptopenjdk/openjdk11 AS builder
COPY gradlew .
COPY gradle gradle
COPY build.gradle .
COPY settings.gradle .
COPY src src
RUN chmod +x ./gradlew
RUN ./gradlew bootJAR
```

```
FROM adoptopenjdk/openjdk11
COPY --from=builder build/libs/*.jar app.jar
EXPOSE 8080
ENTRYPOINT ["sh", "-c", "java ${JAVA_ENC} ${PROFILE} -jar /app.jar"]
```

배포

Nginx

```
upstream backend{
    ip_hash;
    server 172.26.11.110:8080;
}

server {

    listen      80;
    listen     [::]:80;
    server_name customgptchat.com;

    location / {
        root    /usr/share/nginx/html;
        index   index.html index.htm;
        try_files $uri /index.html;
    }

    location /api {
        proxy_pass http://backend;
        proxy_redirect    off;
        proxy_set_header   Host $host;
        proxy_set_header   X-Real-IP $remote_addr;
        proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    #error_page 404                /404.html;

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

Jenkins

```
docker build -t backimg ./Backend
if (docker ps | grep "backimg"); then docker stop backimg; fi
docker run -it -d --rm -p 8080:8080 --env-file /env_file -e JAVA_ENC=-Djasypt.encryptor.password=gjslqhtakwg! -e PROFILE=-Dspring.profiles..
echo "Run backend"

docker build -t frontimg ./Frontend
if (docker ps | grep "frontimg"); then docker stop frontimg; fi
docker run -it -d --rm -p 80:80 -p 443:443 --name frontimg frontimg
echo "Run frontend"
```