

OpenAI chatgpt communication

32181771 모바일시스템공학과 박용관

32182144 모바일시스템공학과 서지현

프로젝트 설명

CLI로 터미널/리눅스 환경에서도 원활하게 chatgpt를 사용할 수 있는 서버 개발

프로젝트 배경 및 목표

- ChatGPT란 OpenAI가 개발한 GPT-3.5 기반의 대형 언어 모델(large language model, LLM) 챗봇을 의미합니다.
- ChatGPT는 대화 형태로 상호작용을 하며 놀라울 정도로 인간과 대화하는 것과 같은 반응을 제공하는 능력을 가지고 있습니다.
- 대화 형 인공지능(AI) 'ChatGPT'는 2022년 11월 30일 발표된 이후 고도의 성능으로 큰 주목을 받고 있습니다.
- ChatGPT의 월간 액티브 유저가 출시 두 달 만에 1억 명에 도달하며 사상 가장 빠른 속도로 성장하는 애플리케이션이 되었습니다.

이렇듯 현재 chatgpt의 사용자는 기하급수적으로 늘어나고 있고 거의 모든 분야에서 chatgpt를 사용하고 있습니다. 가장 보편적으로 사용하는 방법은 'https://chat.openai.com/' 사이트에서 회원 인증 후, 질문을 입력해 답을 받아오는 형식으로 사용합니다. 하지만 이를 cli환경에서 chatgpt를 사용하기가 쉽지 않습니다. curl명령어를 통해 실제 openai 서버로 요청을 보내는 형식으로 사용할 수 있긴 하지만, 후에 예시 자료로 보시면 openai로 직접 요청을 보내는 작업은 단순히 질문 -> 정답이 아닌 부가적인 해당 서버에서 요청하는 형식에 맞게 추가적인 정보를 작성해야하는 번거로움이 있습니다. 또한 정답으로 오는 응답 형식 또한 단순 정답만 원할때는 한눈에 보기 쉽지 않습니다. 따라서 이러한 번거로움을 해결하기 위해 이번 프로젝트를 기획하게 되었습니다. 저희는 alias로 명령어를 설정해 cli 환경에서 저희 서버로 접속할 수 있는 명령어를 만들 것이며 추가적으로, 스프링으로 서버구축을 하여 다채로운 dto설정으로 사용자에게 유익한 데이터 제공할 것 입니다.

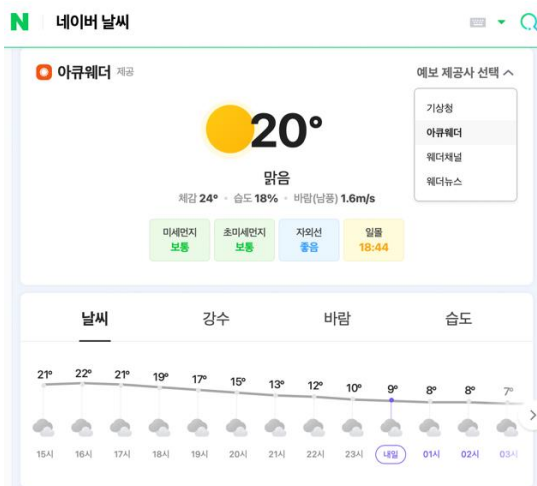
저희 프로젝트의 궁극적인 목표는 “Cli 환경에서 간편하게 chatgpt와 통신할 수 있는 서버를 구축하자” 입니다.

공개 시 기대효과

공개 시 기대효과로는,

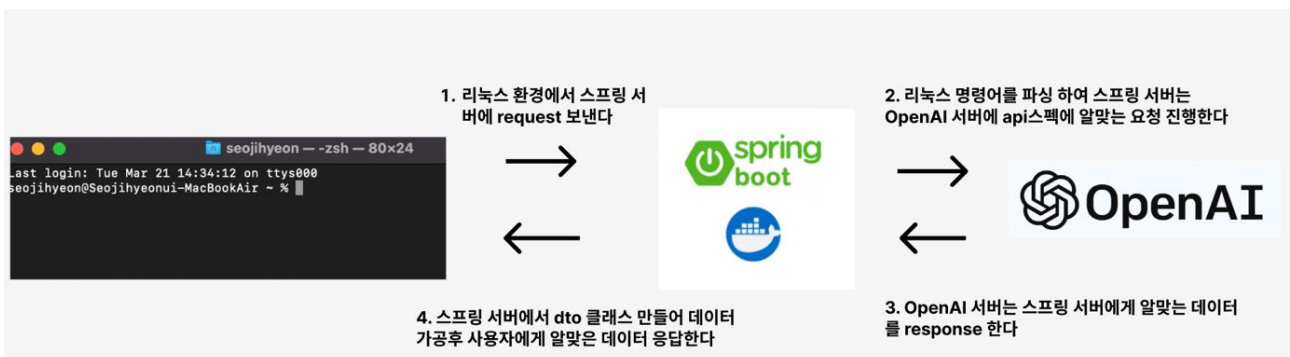
- I. 사용자는 openai에 직접 요청할때 작성해야하는 번거로운 요청 패킷 형식을 알 필요가 없습니다.
- II. openai를 사용하기 위한 api key값을 생성할 필요가 없습니다.
- III. 사용자는 최근 서버 사용자들의 질문들을 볼 수 있습니다.
- IV. cli환경에서 편리하게 chatgpt를 사용할 수 있습니다.

유사 프로젝트 분석



네이버 서버의 경우도 날씨 데이터를 얻기 위해서 기상청 서버, 아쿠웨더 서버, 등 다양한 서버에 요청을 보내고 해당 응답 값을 받습니다. 다양한 서버에 요청을 보낼때 각각의 서버에 요청하는 요청 방식이 다르지만, 해당 서버에 맞는 형식의 요청을 보내고 응답을 받습니다. 응답 또한 응답 받은 형식을 파싱해 원하는 형식으로 응답을 바꿔 출력합니다. 오늘날 이처럼 서버끼리의 통신은 사용자에게 더 좋은 정보, 더 풍부한 검색을 제공해 줄 수 있습니다.

핵심 기술



서버 끼리의 데이터 요청과 응답을 잘 다루는 것을 목표로 하고 있습니다. CLI 환경에서 명령어를 통해 request를 전송하면 스프링 서버는 이를 잘 파싱 하여 OpenAI 서버에게 api팩에 알맞는 요청을 보냅니다. 그 뒤 OpenAI 서버는 스프링 서버에게 알맞는 데이터를 response 해줍니다. 이 때, 스프링 서버는 해당 데이터를 잘 가공하여 사용자에게 알맞는 데이터를 응답해 줍니다.

요구사항 명세서

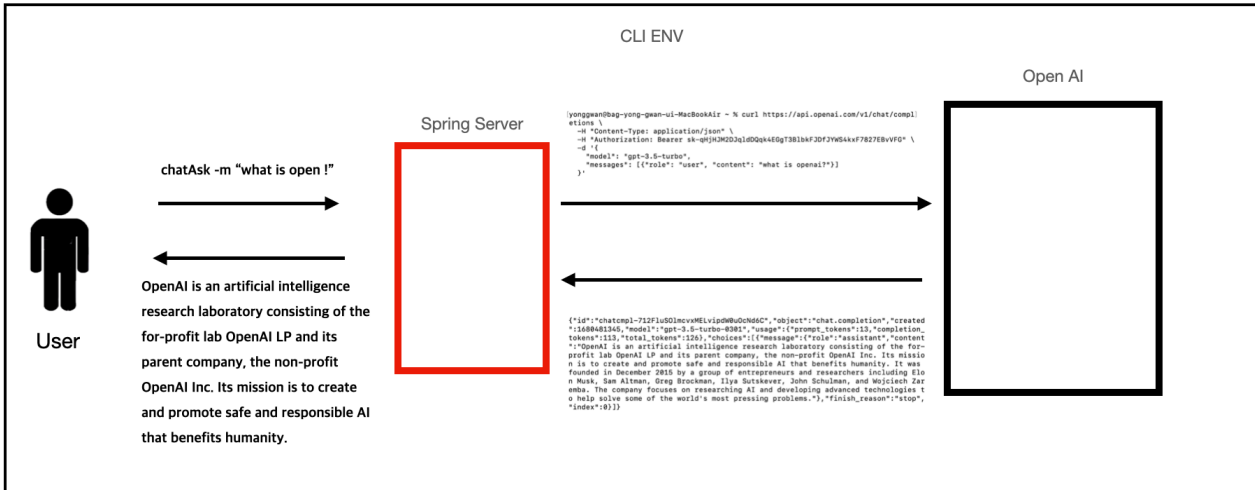
기본적인 chatgpt통신에서 필요한 기능

- 요청 / 응답 패킷 파싱하기
- Chatgpt 통신을 위한 api key를 header에 추가 후(spring 서버에서 작업) open ai 서버에 요청 보내기
- 질문 정보 DB에 저장하기

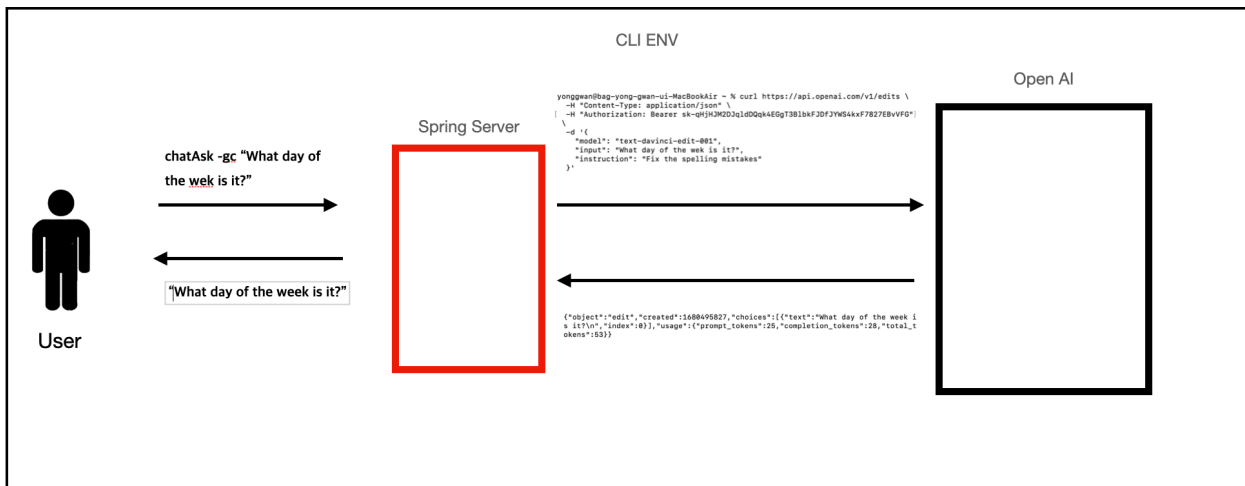
구분	서비스	필요 기능	기능 설명	우선 순위
사용자	chatgpt 통신	질문하기	<ul style="list-style-type: none"> 서버에게 요청 보내기 <ul style="list-style-type: none"> 질문 내용(text 형식) 응답 받은 것을 콘솔 창에 출력 	1
		문장 맞춤법 수정	<ul style="list-style-type: none"> 서버에게 요청 보내기 <ul style="list-style-type: none"> 맞춤법 검사를 위한 문장(text 형식) 응답 받은 것을 콘솔 창에 출력 <ul style="list-style-type: none"> 틀린 부분이 있다면 틀린 부분 알려주기 틀린 부분이 없다면 "this sentence is perfect" 출력하기 	1
		이미지 생성	<ul style="list-style-type: none"> 서버에게 요청 보내기 <ul style="list-style-type: none"> 이미지 설명 이미지 생성 갯수 응답 받은 것을 콘솔 창에 출력 <ul style="list-style-type: none"> 이미지 url 출력 	1
		문장 감성 분석	<ul style="list-style-type: none"> 서버에게 요청 보내기 <ul style="list-style-type: none"> 질문 내용(text 형식) 응답 받은 것을 콘솔 창에 출력 <ul style="list-style-type: none"> 가장 높은 성향을 추출해 콘솔창에 출력 	1
		최근 사용자들의 질문 불러오기	<ul style="list-style-type: none"> 서버에게 요청 보내기 응답 받은 것을 콘솔 창에 출력 <ul style="list-style-type: none"> text형식으로 최근 질문을 보여줌 	1
		유사 이미지 제공	<ul style="list-style-type: none"> 서버에게 요청 보내기 <ul style="list-style-type: none"> 이미지 이미지 생성 갯수 응답 받은 것을 콘솔 창에 출력 <ul style="list-style-type: none"> 이미지 url 출력 	2
		음성 인식	<ul style="list-style-type: none"> 서버에게 요청 보내기 <ul style="list-style-type: none"> 음성 파일 보내기 응답 받은 것을 콘솔 창에 출력 <ul style="list-style-type: none"> text형식의 음성값을 문자화여 응답으로 받음 	2
		음성 번역	<ul style="list-style-type: none"> 서버에게 요청 보내기 <ul style="list-style-type: none"> 음성 파일 보내기 응답 받은 것을 콘솔 창에 출력 <ul style="list-style-type: none"> text형식의 번역된 값을 응답으로 받음 	2

사용자 시나리오

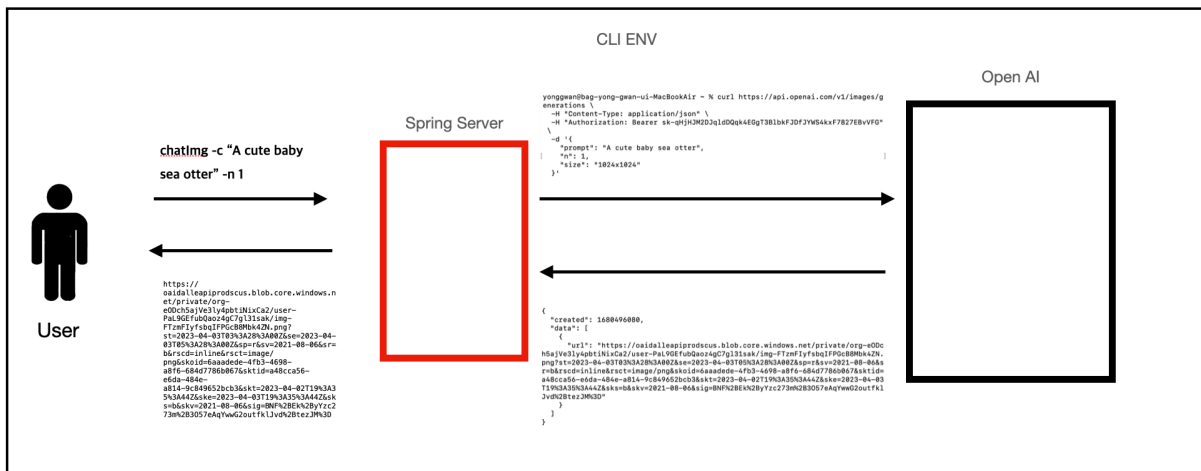
1. 질문 하기



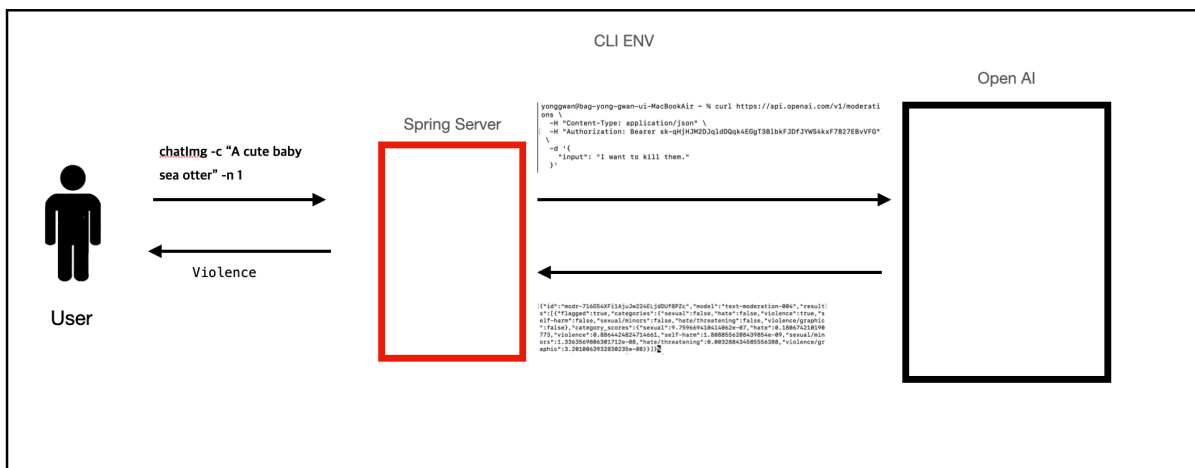
2. 문장 맞춤법 수정



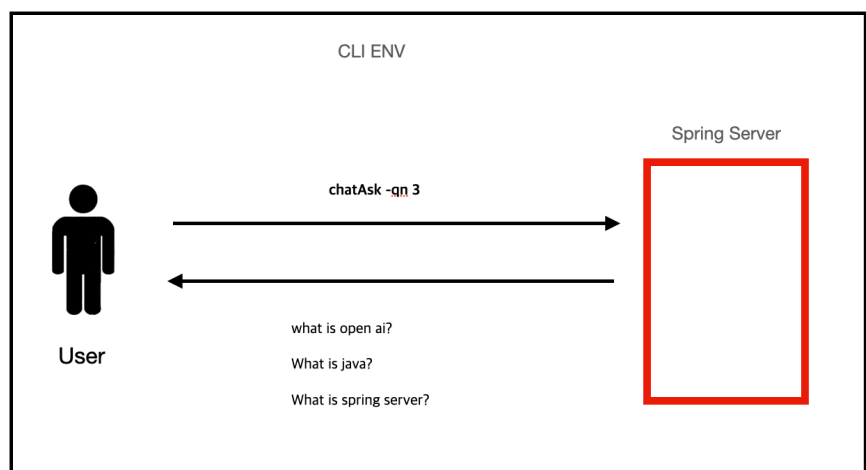
3. 이미지 생성



4. 문장 감성 분석

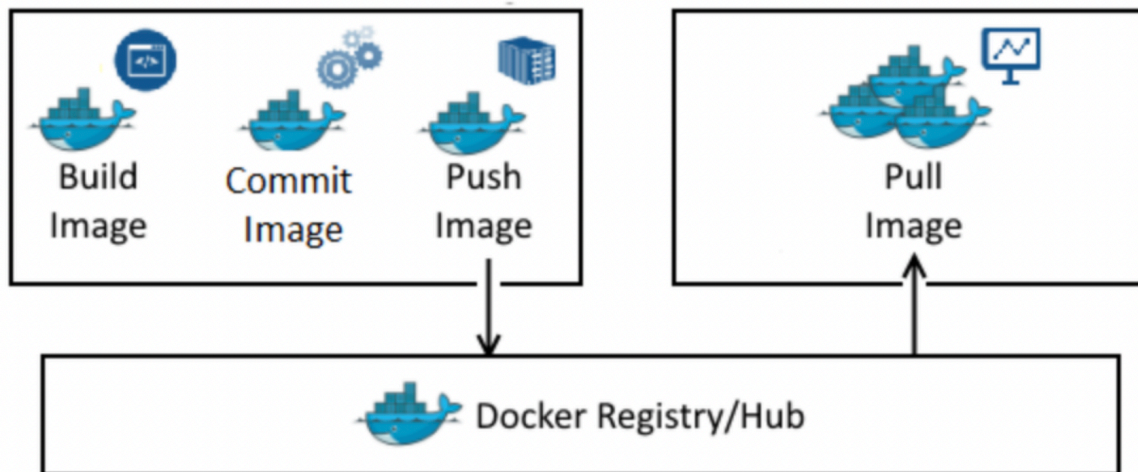


5. 최근 사용자들의 질문 불러오기



추가적인 명령어는 후에 개발 시간이 남는다면 추가적으로 구현하도록 하겠습니다. (요구사항 명세서 기준 우선순위 2 명령어들)

스프링 서버 배포



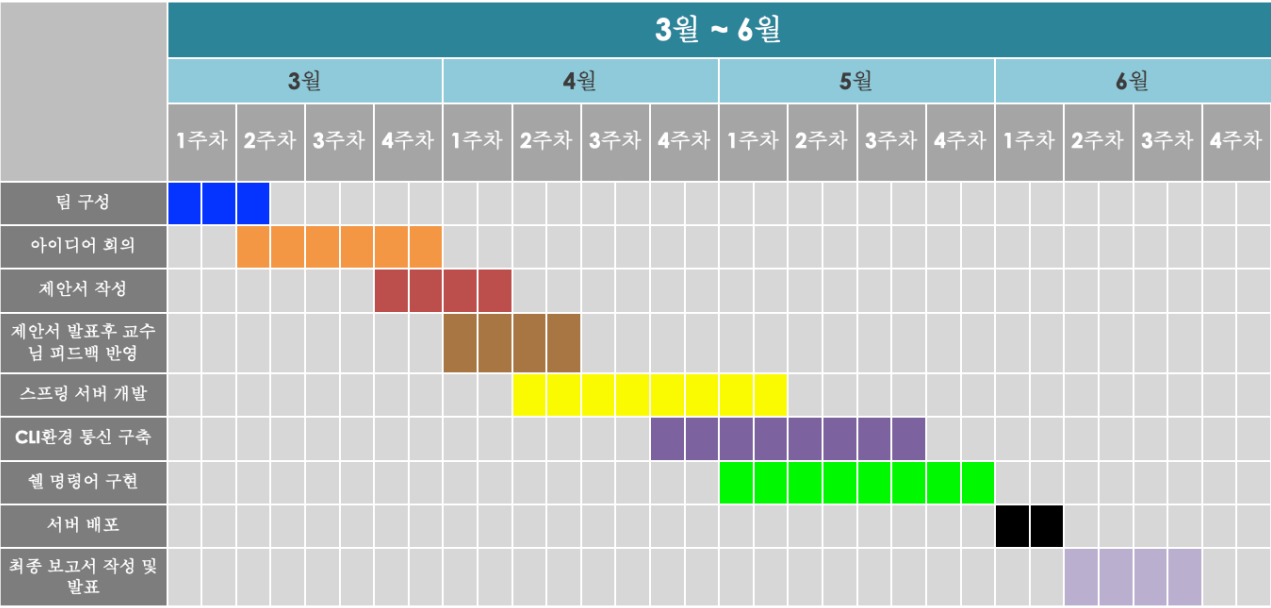
스프링 서버에서 Dockerfile을 통해 docker 이미지를 생성합니다. 해당 이미지를 docker hub에 등록합니다. 클라우드 기반 서버를 하나 생성한 뒤 해당 서버에 docker를 설치합니다. 그 후 docker 이미지를 해당 서버에 다운로드 받은 뒤 컨테이너로 무중단 배포를 진행합니다.

목표 달성 기준 여부

1. 위에서 보여준 User diagram에 맞게 명령어가 제대로 실행이 되는가?
2. 서버 배포가 제대로 완료가 됐는가?
3. 사용자들의 질문이 제대로 서버 DB에 저장이 되는가?

추진 계획

> GANTT CHART



역할 분담

이름	역할	세부 역할
박용관	백엔드 개발	1. Spring 서버 개발 2. 쉘 프로그래밍을 통한 리눅스 명령어 개발
서지현	백엔드 개발	1. Spring 서버 개발 2. 도커를 통한 서버 배포

참고 문헌

- <https://platform.openai.com/docs/api-reference/>
- <https://platform.openai.com/account/api-keys>
- <https://subicura.com/2017/02/10/docker-guide-for-beginners-create-image-and-deploy.html>
- <https://velog.io/@chlwogur2/Spring-%EC%99%B8%EB%B6%80-API-%ED%98%B8%EC%B6%9C-%EB%A1%9C%EC%A7%81%EC%97%90-%EA%B4%80%ED%95%B4>