

Chatgpt 리눅스 명령어

사용자 가이드

목차

| | |
|-----------------------------------|---|
| > 프로젝트 개요 | 3 |
| > 프로젝트 특징 | 3 |
| > 제공하는 서비스 | 3 |
| > 서비스 대상 | 3 |
| > 서비스 구성 | 4 |
| > 파일 구조 | 4 |
| > 서비스 사용법 | 5 |
| I. Spring 서버를 로컬에서 띄워서 사용하고 싶은 경우 | 5 |
| II. 배포된 서버를 사용하는 방법 | 6 |
| > 추가적인 기여 방법 | 7 |
| I. 소셜 이메일 사용 | 7 |
| II. github 사용 | 7 |
| > 참고 자료 | 7 |

> 프로젝트 개요

리눅스 환경에서 Open AI 서버와 통신할 수 있는 리눅스 환경 CLI 명령어를 제공합니다.

> 프로젝트 특징

- 누구나 쉽게 사용할 수 있는 리눅스 기반 CLI 명령어 기능을 제공합니다.
- 편리하게 Open AI 서버와 통신할 수 있는 서버와 리눅스 전용 CLI 명령어를 제공합니다.
- CLI 환경에서 리눅스 명령어만으로 Open AI에서 제공하는 서비스를 사용할 수 있습니다.

> 제공하는 서비스

- 질문하기, 문법확인하기, 음성파일 대본 만들기, 감정 분석, 문장 번역, 문장 요약, 상위 키워드 추출, 등 다양한 서비스를 제공합니다.
- Amazon Linux, Mac OS, Ubuntu 등 리눅스 기반 많은 OS에서 서비스를 이용할 수 있습니다.

> 서비스 대상

- 리눅스 명령어를 이용하여 개발하는 하드웨어 업체
- Open AI 서버 활용 도입을 목표로 하고 있는 업체
- Open AI 서버로 부터 깔끔한 응답을 확인하고 싶은 분
- 리눅스 명령어에 관심이 있으신 분
- 리눅스 환경에서 서버를 개발할 때 chatgpt 서비스를 이용하고 싶은 분

> 서비스 구성

| 기능 | 명령어 형식 | 역할 |
|-------------|--|------------------------------|
| 도움말 | <code>chatAsk -help</code> | 사용자 메뉴얼을 확인한다. |
| 질문하기 | <code>chatAsk -m "Content of questions"</code> | 질문에 대한 응답을 확인한다. |
| 세션종료 | <code>chatAsk -clear</code> | 질문 세션 기능을 종료한다. |
| 문법 확인하기 | <code>chatAsk -gc "Question Content"</code> | 영어문장을 문법을 확인 후 올바른 문장을 제공한다. |
| 음성파일 대본 만들기 | <code>chatAsk -at voice file.m4a</code> | 음성파일을 전송하면 이에 맞는 대본을 만들어 준다. |
| 감정 분석 | <code>chatAsk -md "Question Content"</code> | 문장에 해당하는 감정 분석 결과값을 확인한다. |
| 문장 번역 | <code>chatAsk -t "Question Contents"</code> | 영어문장을 한국어로 번역한다. |
| 문장 요약 | <code>chatAsk -md "Question Content"</code> | 긴 문장을 짧은 문장으로 요약한다. |
| 상위 키워드 추출 | <code>chatAsk -rank</code> | 사용자가 질문한 상위 5개의 키워드를 확인한다. |

※ 해당 명령어는 이 후 언제든지 추가될 수 있습니다.

> 파일 구조

1. shell 폴더

리눅스 명령어(chatAsk)를 alias로 등록해 리눅스기반 CLI 환경 안에서 명령어를 실행할 수 있는 환경 설정 shell script(ChatgptCommand.sh)가 존재합니다.

2. src 폴더

Springboot 서버 관련 프로젝트 파일이 존재합니다.

3. 그 외 파일

Springboot 전용 파일 & 폴더가 존재합니다.

> 서비스 사용법

I. Spring 서버를 로컬에서 띄워서 사용하고 싶은 경우

```
git clone https://github.com/YGwan/spring-chatgpt-communication.git
cd spring-chatgpt-communication/shell
source chatgptCommand.sh
```

1. **git**과 **java** 버전 **11**을 다운로드 (이미 다운이 되어 있는 상태라면 생략해도 된다.)

2. 리눅스 서버에서 위의 명령어를 실행하여 **chatgptCommand.sh** 파일을 실행

```
[ec2-user@ip-172-31-40-7 demo]$ sudo chmod 777 ./gradlew
[ec2-user@ip-172-31-40-7 demo]$ ll ./gradlew
-rwxrwxrwx 1 ec2-user ec2-user 5296 Feb 21 05:04 ./gradlew
[ec2-user@ip-172-31-40-7 demo]$ ./gradlew build
Downloading https://services.gradle.org/distributions/gradle-6.0.1-bin.zip
.....

Welcome to Gradle 6.0.1!
...
Starting a Gradle Daemon (subsequent builds will be faster)
...

BUILD SUCCESSFUL in 2m 17s
5 actionable tasks: 5 executed

[ec2-user@ip-172-31-40-7 demo]$ cd build/libs/
[ec2-user@ip-172-31-40-7 libs]$ java -jar spring-chatgpt-communication-0.0.1-SNAPSHOT.jar
```

3. 리눅스 기반 서버에서 위의 명령어를 실행하여 실행 권한을 얻고 프로젝트 빌드 및 서버배포를 진행

(이때 쉘 스크립트의 **ip**가 **localhost**인지 확인해야 한다. 아니면 수정)

4. 터미널에 접속하여 **chatAsk -help**를 통해 제공되는 명령어 형식을 확인

5. 제공되는 명령어를 사용하며 해당 결과를 확인

II. 배포된 서버를 사용하는 방법

```
git clone https://github.com/YGwan/spring-chatgpt-communication.git
cd spring-chatgpt-communication/shell
source chatgptCommand.sh
```

1. 배포된 서버 확인 (현재 비용 문제로 배포된 서버를 중지한 상태입니다. 필요하다면 문의 바랍니다.)
2. **git**과 **java** 버전 **11**을 다운로드 (이미 다운이 되어 있는 상태라면 생략해도 된다.)
3. 리눅스 서버에서 위의 명령어를 실행하여 **chatgptCommand.sh** 파일을 실행 (이때 쉘 스크립트의 **ip** 주소가 배포된 서버 주소와 맞는지 확인)
4. 터미널에 접속하여 **chatAsk -help**를 통해 제공되는 명령어 형식을 확인
5. 제공되는 명령어를 사용하며 해당 결과를 확인

현재 **shell script**의 서버 **ip** 주소는 **localhost**, 포트는 **8081**로 설정되어있다. 배포된 서버를 사용하려면 문의 후 얻은 서버 주소와 포트로 해당 파일 부분을 수정하면 된다.

> 추가적인 기여 방법

I. 소셜 이메일 사용

devygwan@gmail.com에 메일 보내기

II. github 사용

기여하고 싶은 방향에 대한 내용과 함께 PR

※ 답변은 최대 1주일까지 소요될 수 있습니다.

> 참고 자료

<https://platform.openai.com/docs/>

감사합니다.