Jina AI Services

This Python script interacts with Jina AI services using API keys and command-line arguments. It supports two main jobs: fetching content from a URL and performing a search query. The script retrieves the Jina API key from the environment variables, ensuring secure access to the services. It uses the `requests` library to make HTTP requests and `base64` to decode the search query. The script then prints the response from the Jina AI service.

```
import os
import requests
from dotenv import load_dotenv
import argparse
import base64

load_dotenv()

api_key = os.environ.get("JINA_API_KEY")
if not api_key:
    raise ValueError("JINA_API_KEY environment variable not set.")

parser = argparse.ArgumentParser()
parser.add_argument("--job", type=str, choices=['url', 'search'], help="Job to execute (url or search)", requi
parser.add_argument("--input", type=str, help="Input for the job", required=True)
args = parser.parse_args()

if args.job == 'url':
    url = f'https://r.jina.ai/{args.input}'
    headers = {'Authorization': f'Bearer {api_key}'}
    print(f"URL: {url}")
    print(f"Headers: {headers}")
    response = requests.get(url, headers=headers)
    print(response.text)

elif args.job == 'search':
    question = base64.b64decode(args.input).decode('utf-8', errors='ignore')
    url = f'https://s.jina.ai/{question}'
    headers = {
        'Authorization': f'Bearer {api_key}',
        'X-Engine': 'direct',
        'X-Retain-Images': 'none'
```

```
    }
    print(f"URL: {url}")
    print(f"Headers: {headers}")
    response = requests.get(url, headers=headers)
    print(response.text)

else:
    print("Please specify --job url or --job search")
```