

Services Jina AI

Ce script Python interagit avec les services Jina AI en utilisant des clés API et des arguments en ligne de commande. Il prend en charge deux tâches principales : récupérer du contenu à partir d'une URL et effectuer une requête de recherche. Le script récupère la clé API Jina à partir des variables d'environnement, garantissant ainsi un accès sécurisé aux services. Il utilise la bibliothèque `requests` pour effectuer des requêtes HTTP et `base64` pour décoder la requête de recherche. Le script imprime ensuite la réponse du service Jina AI.

```
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("La variable d'environnement JINA_API_KEY n'est pas définie.")

parser = argparse.ArgumentParser()
parser.add_argument("--job", type=str, choices=['url', 'search'], help="Tâche à exécuter (url ou search)", required=True)
parser.add_argument("--input", type=str, help="Entrée pour la tâche", 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("Veuillez spécifier --job url ou --job search")
```