

# Servicios de IA Jina

Este script de Python interactúa con los servicios de Jina AI utilizando claves API y argumentos de la línea de comandos. Soporta dos tareas principales: obtener contenido de una URL y realizar una consulta de búsqueda. El script recupera la clave API de Jina desde las variables de entorno, asegurando un acceso seguro a los servicios. Utiliza la biblioteca `requests` para realizar solicitudes HTTP y `base64` para decodificar la consulta de búsqueda. Luego, el script imprime la respuesta del servicio de 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 de entorno JINA_API_KEY no está configurada.")

parser = argparse.ArgumentParser()
parser.add_argument("--job", type=str, choices=['url', 'search'], help="Tarea a ejecutar (url o search)", required=True)
parser.add_argument("--input", type=str, help="Entrada para la tarea", 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("Por favor, especifique --job url o --job search")
```