A screenshot of a computer

AI-generated content may be incorrect.

**User-agent: \* (Applies to all bots)**

* Disallow: /api  
  Crawlers **cannot access anything under /api** path.
* Disallow: /asset-manifest.json  
  The specific file asset-manifest.json is disallowed.
* Allow: /search/$  
  The exact URL /search/ (end of URL) **is allowed**.
* Disallow: /search/  
  But any other URLs starting with /search/ (like /search/something) **are disallowed**.
* Disallow: /home/search?q=  
  Disallows URLs starting with /home/search?q= query.

**Specific User-agents Disallowed Completely**

* These user-agents are explicitly **disallowed from crawling any part of the site** (Disallow: / means full block):
  + anthropic-ai
  + ChatGPT-User
  + ClaudeBot
  + Claude-Web
  + cohere-ai
  + GPTBot
  + PerplexityBot
  + Bytespider

**Sitemap URLs**

* The file lists multiple sitemap URLs which help crawlers find all public pages:
  + https://www.aljazeera.com/sitemap.xml
  + https://www.aljazeera.com/news-sitemap.xml
  + https://www.aljazeera.com/sitemaps/article-archive.xml
  + https://www.aljazeera.com/sitemaps/article-new.xml
  + https://www.aljazeera.com/sitemaps/video-archive.xml
  + https://www.aljazeera.com/sitemaps/video-new.xml

Code to get any site robots.txt :

import requests

def fetch\_robots\_txt(url):

if not url.endswith('/'):

url += '/'

robots\_url = url + 'robots.txt'

response = requests.get(robots\_url)

if response.status\_code == 200:

return response.text

else:

print(f"Failed to fetch robots.txt: HTTP {response.status\_code}")

return None

def parse\_robots\_txt(content):

rules = {}

sitemaps = []

current\_user\_agent = None

for line in content.splitlines():

line = line.strip()

if not line or line.startswith('#'):

continue

if line.lower().startswith('user-agent:'):

current\_user\_agent = line.split(':', 1)[1].strip()

if current\_user\_agent not in rules:

rules[current\_user\_agent] = {'allow': [], 'disallow': []}

elif line.lower().startswith('allow:') and current\_user\_agent:

path = line.split(':', 1)[1].strip()

rules[current\_user\_agent]['allow'].append(path)

elif line.lower().startswith('disallow:') and current\_user\_agent:

path = line.split(':', 1)[1].strip()

rules[current\_user\_agent]['disallow'].append(path)

elif line.lower().startswith('sitemap:'):

sitemap\_url = line.split(':', 1)[1].strip()

sitemaps.append(sitemap\_url)

return rules, sitemaps

if \_\_name\_\_ == "\_\_main\_\_":

url = "https://www.aljazeera.com"

content = fetch\_robots\_txt(url)

if content:

rules, sitemaps = parse\_robots\_txt(content)

for agent, paths in rules.items():

print(f"User-agent: {agent}")

print(" Allow:")

for p in paths['allow']:

print(f" {p}")

print(" Disallow:")

for p in paths['disallow']:

print(f" {p}")

print()

print("Sitemaps found:")

for sitemap in sitemaps:

print(f" {sitemap}")

Crawl-delay :

import requests

from urllib.robotparser import RobotFileParser

robots\_url = "https://www.aljazeera.com/robots.txt"

rp = RobotFileParser()

rp.set\_url(robots\_url)

rp.read()

# crawl\_delay

delay = rp.crawl\_delay("\*")

print(f"Crawl-delay is: {delay} seconds")