W3 Discussion

1. CURL provides a simple interface to extract information from a webpage. It can extract resources from a web-page, ready to be parsed by another program. It can get header information, submit post requests, and even authenticate through numerous protocols. CURL can be used via the command line and there are also libraries available for many different languages such as Python. For web bots, CURL can quickly gather the information specifies (i.e. HTTP header, or content) and store it quickly, bypassing mechanisms like authentication that would otherwise stop the download of the information.

2. Web bots or web spiders can run into several challenges when crawling the web. Server-side scripts can generate links to the same content in several different ways. Parameters such as results-per-page, grid or list, filters, etc. all create different unique URL’s that impede the crawler’s ability to uniquely identify content. Another challenge is the resource load on the webpages themselves. The web bot can make dozens of servers calls in a second which will put considerable stress on the server. Web pages might attempt to block web bot requests or request that the bot only index part of the page in a “robots.txt” file.

3. The CURL command line tools supports HTTP, HTTPS, FTP, FTPS, SCP, SFTP, TFTP, LDAP, DAP, DICT, TELNET, FILE, IMAP, POP3, SMTP and RTSP.

4. Pycurl is Python’s interface to the libcurl library for CURL. It provides similar functionality to the urllib built-in module for Python. At a basic level, it will retrieve web page information and store it into an object. You specify a variety of parameters such as invoking callback function after the headers or body is ready.

5.

import pycurl

from io import BytesIO

buffer = BytesIO()

url = "http://pycurl.io/"

curlHandle = pycurl.Curl()

curlHandle.setopt(curlHandle.URL, url)

curlHandle.setopt(curlHandle.WRITEDATA, buffer)

curlHandle.perform()

curlHandle.close()

body = buffer.getvalue().decode("UTF-8")

print(body)