Web Science: Assignment #1

Alexander Nwala

Apurva Modi

Sunday, January 28, 2018

Contents

Problem 1

Demonstrate that you know how to use "curl" well enough to correctly POST data to a form.Show that the HTML response that is returned is "correct".That is, the server should take the arguments you POSTed and build a response accordingly. Save the HTML response to a file and then view that file in a browser and take a screen shot.

SOLUTION :

```
curl -i -d "fname=APURVA&lname=MODI" -X POST
http://www.cs.odu.edu/~anwala/files/temp/namesEcho.php
```

```
RocketScientist:~ apurvamodi$ curl -i -d "fname=apurva&lname=modi" -X POST https://www.cs.odu.edu/~anwala/files/temp/namesEcho.php
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 01 Feb 2019 01:25:31 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Front-End-Https: on

<!DOCTYPE html>
<html>
<body>

<br />
<br />
<b>fname Posted: </b>apurva<br />
<b>lname Posted: </b>modi<br />

</body>
</html>RocketScientist:~ apurvamodi$ curl -i -d "fname=APURVA&lname=MODI" -X POST https://www.cs.odu.edu/~anwala/files/temp/namesEcho.php
HTTP/1.1 200 OK
Server: nginx
Date: Fri, 01 Feb 2019 01:26:15 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
Front-End-Https: on

<!DOCTYPE html>
<html>
<body>

<br />
<br />
<b>fname Posted: </b>APURVA<br />
<b>lname Posted: </b>MODI<br />

</body>
</html>RocketScientist:~ apurvamodi$
```

Figure 1: Data posting with curl

Problem 2

Write a Python program that:

1. takes as a command line argument a web page extracts all the links from the page
2. lists all the links that result in PDF files, and prints out the bytes for each of the links. (note: be sure to follow all the redirects until the link terminates with a "200 OK".) show that the program works on 3 different URIs, one of which needs to be: http://www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html

SOLUTION

The solution for this problem is outlined by the following steps:

1. **Extracting the Links from the Web Pages :**

```
RocketScientist:Web-Science-532-s19 apurvamodi$ python a1test.py www.google.com
/anaconda3/lib/python3.6/site-packages/bs4/__init__.py:181: UserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("lxml"). This usually isn't a prob
lem, but if you run this code on another system, or in a different virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 7 of the file a1test.py. To get rid of this warning, change code that looks like this:

 BeautifulSoup(YOUR_MARKUP})

to this:

 BeautifulSoup(YOUR_MARKUP, "lxml")

  markup_type=markup_type))
https://www.google.com/imghp?hl=en&tab=wi
https://maps.google.com/maps?hl=en&tab=wl
https://play.google.com/?hl=en&tab=w8
https://www.youtube.com/?gl=US&tab=w1
https://news.google.com/nwshp?hl=en&tab=wn
https://mail.google.com/mail/?tab=wm
https://drive.google.com/?tab=wo
https://www.google.com/intl/en/about/products?tab=wh
http://www.google.com/history/optout?hl=en
/preferences?hl=en
https://accounts.google.com/ServiceLogin?hl=en&passive=true&continue=https://www.google.com/
/search?site=&ie=UTF-8&q=Sojourner+Truth&oi=ddle&ct=celebrating-sojourner-truth-5641167843622912&hl=en&kgmid=/m/077_8&sa=X&ved=0ahUKEwjenpmf7ZngAhXYIDQIHbY2DYwQPQgD
/advanced_search?hl=en&authuser=0
/language_tools?hl=en&authuser=0
https://www.google.com/url?q=https://artsandculture.google.com/exhibit/vwIi3HBUyJcCIA%3Futm_campaign%3Dsojournertruth19%26utm_source%3Dgoogle%26utm_medium%3Ddoodlehppromo&source=hpp&id=19010564&ct=3&usg=A
FQjCNEiCxghM-zq8lWAhMZgMoJCZrY3TA&sa=X&ved=0ahUKEwjenpmf7ZngAhXYIDQIHbY2DYwQ8IcBCAU
/intl/en/ads/
/services/
https://plus.google.com/116899029375914044550
/intl/en/about.html
/intl/en/policies/privacy/
/intl/en/policies/terms/
RocketScientist:Web-Science-532-s19 apurvamodi$
```

Problem 2 continued on next page. . .

```
RocketScientist:Web-Science-532-s19 apurvamodi$ python a1test.py www.cs.odu.edu/~amod
/anaconda3/lib/python3.6/site-packages/bs4/__init__.py:181: UserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("lxml"). This usually isn't a prob
lem, but if you run this code on another system, or in a different virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 7 of the file a1test.py. To get rid of this warning, change code that looks like this:

 BeautifulSoup(YOUR_MARKUP})

to this:

 BeautifulSoup(YOUR_MARKUP, "lxml")

  markup_type=markup_type))
index.html
about.html
work.html
contact.html
https://www.odu.edu/
https://www.instagram.com/code_er_404/
https://www.facebook.com/apurva.modi.7
https://www.linkedin.com/in/apurva-modi-53a51751/
https://github.com/apurva-modi
https://medium.com/@modiapurva03
apurvResume.pdf
RocketScientist:Web-Science-532-s19 apurvamodi$ python a1test.py www.cs.odu.edu/~mln/teaching/cs532-s17/test/pdfs.html
/anaconda3/lib/python3.6/site-packages/bs4/__init__.py:181: UserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("lxml"). This usually isn't a prob
lem, but if you run this code on another system, or in a different virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 7 of the file a1test.py. To get rid of this warning, change code that looks like this:

 BeautifulSoup(YOUR_MARKUP})

to this:

 BeautifulSoup(YOUR_MARKUP, "lxml")

  markup_type=markup_type))
http://twitter.com/webscidl
http://www.dlib.org/dlib/november15/vandesompel/11vandesompel.html
http://arxiv.org/abs/1508.02315
http://arxiv.org/abs/1508.02315
http://www.cs.odu.edu/~mln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
http://arxiv.org/pdf/1512.06195
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-stories.pdf
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
http://dx.doi.org/10.1007/s00799-015-0150-6
http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-damage.pdf
http://arxiv.org/abs/1506.06279
http://dx.doi.org/10.1007/s00799-015-0155-1
http://bit.ly/1ZDatNK
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf
http://bit.ly/jcdl-pdf
http://dx.doi.org/10.1007/s00799-015-0140-8
RocketScientist:Web-Science-532-s19 apurvamodi$
```

Figure 2: Command Line Argument

Listing 1: A1.py

```
from bs4 import BeautifulSoup
import requests
url = input("Enter the search query :");

re=requests.get("http://" + url);
data =re.text
soup =BeautifulSoup(data)
file = open("pdfurl.txt","w")
for link in soup.find_all('a'):
    line = link.get('href')
    if "http" not in line :
        continue
    else:
        response = requests.get(line)
        if (response.headers['content-type'] == 'application/pdf'):
            hrefLinks=link.get('href')
```

```
            contentLength =response.headers['content-length']
            file.write(hrefLinks+"\n"+contentLength + " bytes\n")

file.close()
```

2. **The above code when runs with test URI and Adds PDF links to pdfurl.txt file with size in bytes.**

Listing 2: PDF files with their size

```
http://www.cs.odu.edu/~mln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
2184076 bytes
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
622981 bytes
http://arxiv.org/pdf/1512.06195
1748959 bytes
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
4308768 bytes
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-stories.pdf
1274604 bytes
http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
639001 bytes
http://www.cs.odu.edu/~mln/pubs/jcdl-2014/jcdl-2014-brunelle-damage.pdf
2205546 bytes
http://bit.ly/1ZDatNK
720476 bytes
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
1254605 bytes
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
709420 bytes
http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf
2350603 bytes
```

Problem 3

Consider the "bow-tie" graph in the Broder et al. paper (fig 9): "**http://www9.org/w9cdrom/160/160.html**
Now consider the following graph:

```
A --> B
B --> C
C --> D
C --> A
C --> G
E --> F
G --> C
G --> H
I --> H
I --> K
L --> D
M --> A
M --> N
N --> D
O --> A
P --> G
```

For the above graph, give the values for:

```
IN:
SCC:
OUT:
Tendrils:
Tubes:
Disconnected:
```

SOLUTION

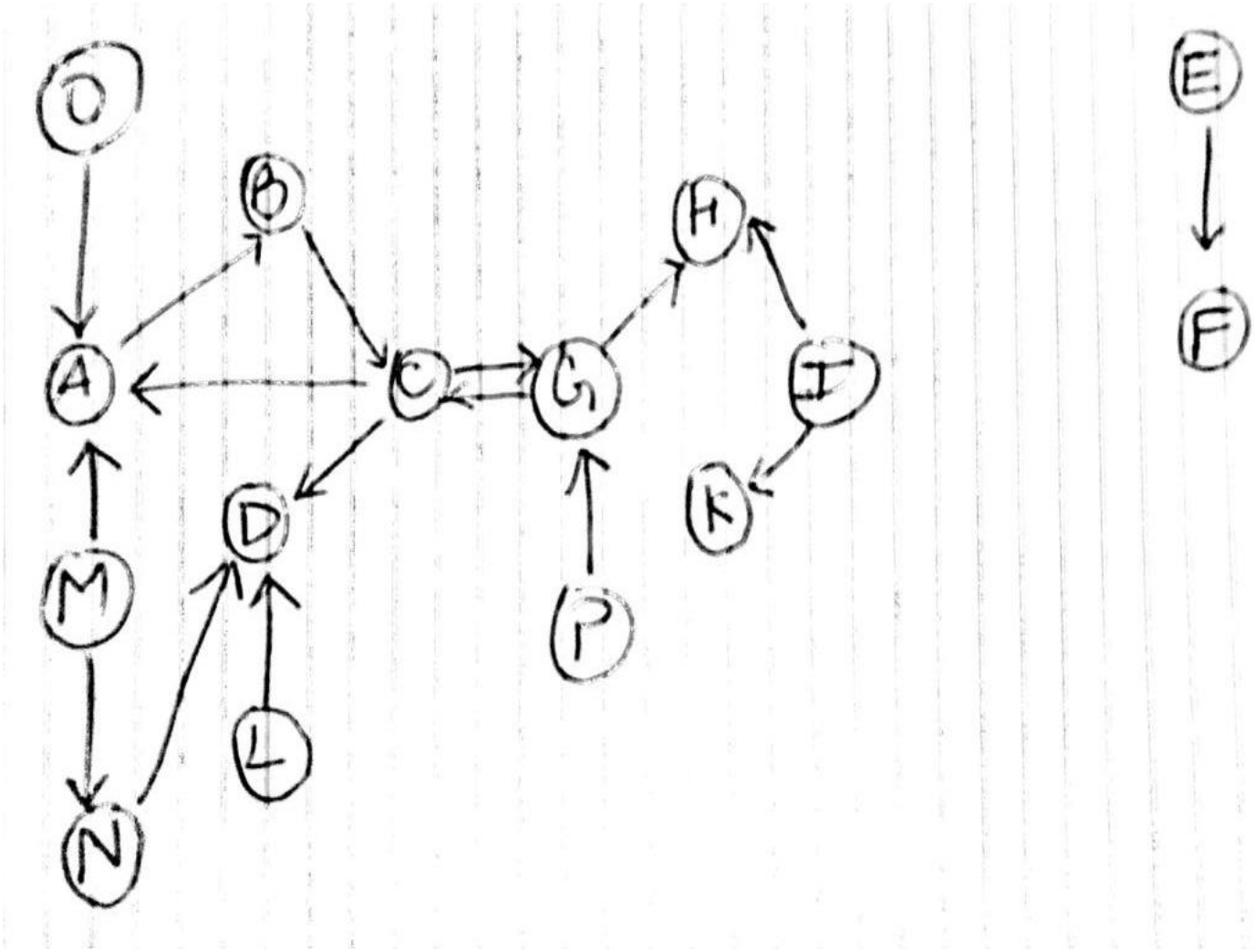

Figure 3: Demonstration of the graph

```
IN: M , O , P
SCC: C , B , G , A
OUT:  D , H
Tendrils: K , I , L
Tubes: N
Disconnected: E , F
```