Web Science cs532-s16

Assignment 1 Report

By: Huan Huang

01/28/2016

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.

Answer

For this problem, I first created a form by using Google Form. Here is a picture of the form.

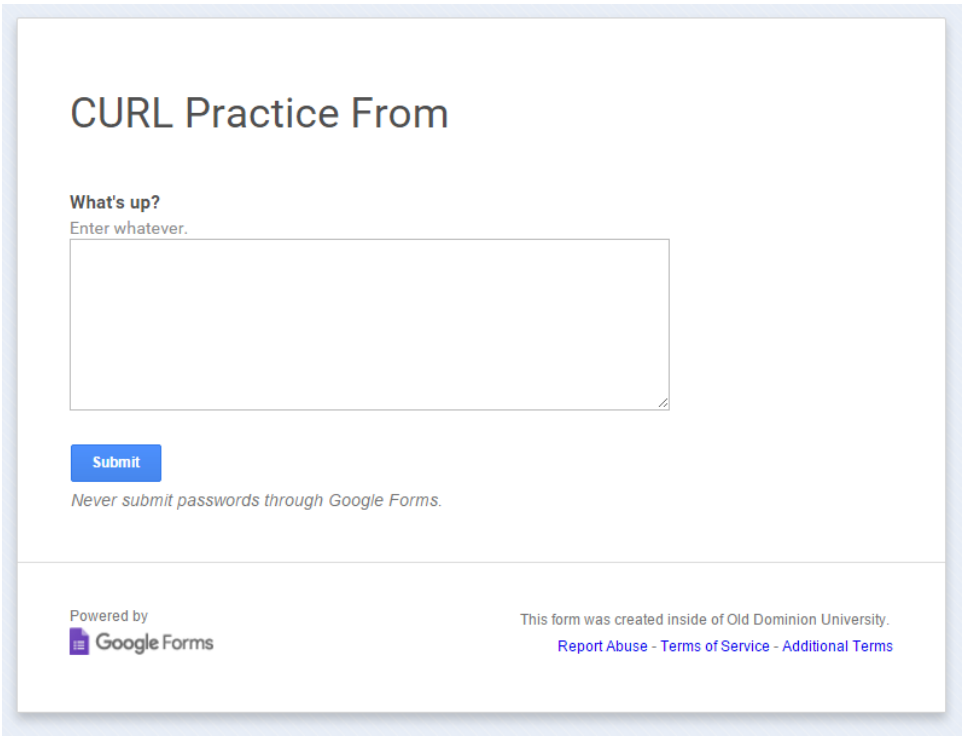


Figure 1: Sample of my form

From there I opened its source page, where I found the text box link by searching for the form and action tags. For this type of form, I also searched for the entry link.

```
</div></div>
<div class="ss-form"><form action="https://docs.google.com/a/odu.edu/forms/d/1IzL-E_COkFP3j6mz93GQmXv4SFr0Wzs4Ad9p3emWe28/formResponse" method="POST" id="ss-form"
style="padding-left: 0">
<div class="ss-form-question errorbox-good" role="listitem">
<div dir="auto" class="ss-item  ss-paragraph-text"><div class="ss-form-entry">
<label class="ss-q-item-label" for="entry_319615868"><div class="ss-q-title">What&#39;s up?
</div>
<div class="ss-q-help ss-secondary-text" dir="auto">Enter whatever.</div></label>
<textarea name="entry.319615868" rows="8" cols="0" class="ss-q-long" id="entry_319615868" dir="auto" aria-label="What&#39;s up? Enter whatever. "></textarea>
<div class="error-message" id="1992178123_errorMessage"></div>
<div class="required-message">This is a required question</div>
</div></div></div>
<input type="hidden" name="draftResponse" value="[,,&quot;-3465971618998774968&quot;]
">
<input type="hidden" name="pageHistory" value="0">

<input type="hidden" name="fvv" value="0">


<input type="hidden" name="fbzx" value="-3465971618998774968">
```

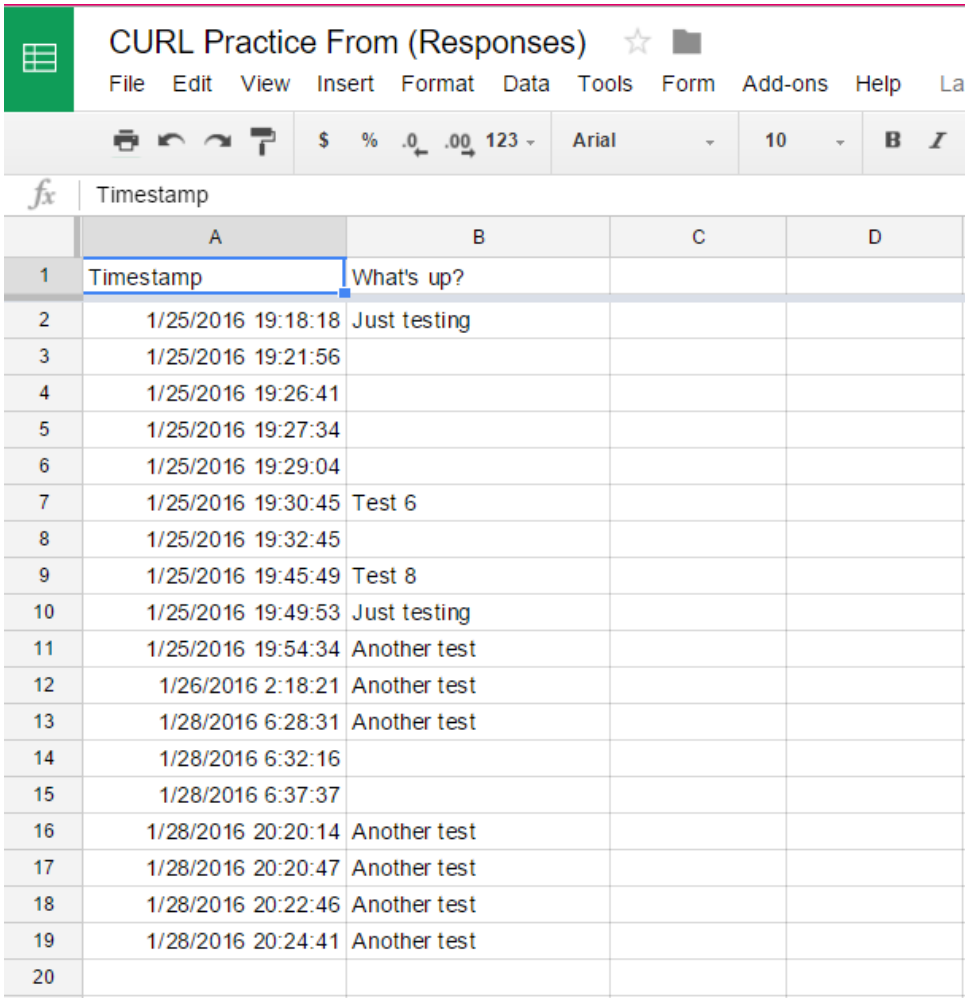
Figure 2: Part of the source code of my form page

Now I just use curl command with -i and -d options. The -i option requests for response from the page server and -d option posts my message on the form. Following the options are the entry link and my actions; after that is the text box link. The image below is just a screen capture of the response header, the full response is saved in a html file call post.html which I will provide in my github repository.

```
hhuang@ubuntu:~/Documents/CS532$ curl -i -d "entry.319615868=Another test & submit=Submit" "h
ttps://docs.google.com/a/odu.edu/forms/d/1IzL-E_COkFP3j6mz93GQmXv4SFr0Wzs4Ad9p3emWe28/formRes
ponse"
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
X-Robots-Tag: noindex, nofollow, nosnippet
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: Fri, 01 Jan 1990 00:00:00 GMT
Date: Fri, 29 Jan 2016 01:22:46 GMT
P3P: CP="This is not a P3P policy! See https://support.google.com/accounts/answer/151657?hl=e
n for more info."
P3P: CP="This is not a P3P policy! See https://support.google.com/accounts/answer/151657?hl=e
n for more info."
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Server: GSE
Set-Cookie: NID=76=r4MuzmGCi0eKA85b1Tjq-8ZPXH7jhWR1bTPh7NrS0VpV9YegXulLKTLsUDPXDhEur-fy4b4wJn
mr5AabXKYChwV6_8gXISngVZ4b7Di8D_U6nMne5izl_V6VO60qpP05;Domain=.google.com;Path=/;Expires=Sat,
 30-Jul-2016 01:22:46 GMT;HttpOnly
Set-Cookie: NID=76=tr38h9Xg0VNqDIuPV8Nq1qPX-zO1LL9_qZOW1sPHXhs82qmrozVY8h9OyiAFlZ6DMVS4FLnNrN
0bbQmwBsMElYFK0F9oCLv5laHstROeTeU8A6WkI3swrj-0oLoZ23Sd;Domain=.google.com;Path=/;Expires=Sat,
 30-Jul-2016 01:22:46 GMT;HttpOnly
Set-Cookie: S=spreadsheet_forms=XKoJwlEZmJ5wYNtDgvIWiw; Domain=.docs.google.com; Expires=Fri,
 29-Jan-2016 02:22:46 GMT; Path=/a/odu.edu/forms/d/1IzL-E_COkFP3j6mz93GQmXv4SFr0Wzs4Ad9p3emWe
28; Secure; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding
Transfer-Encoding: chunked
```

Figure 3: Response of my post

Whatever message that was posted to the form is automatically saved to a spread sheet by Google. Here are the posts I made to the form.



CURL Practice From (Responses)

File Edit View Insert Format Data Tools Form Add-ons Help La

Timestamp

	A	B	C	D
1	Timestamp	What's up?		
2	1/25/2016 19:18:18	Just testing		
3	1/25/2016 19:21:56			
4	1/25/2016 19:26:41			
5	1/25/2016 19:27:34			
6	1/25/2016 19:29:04			
7	1/25/2016 19:30:45	Test 6		
8	1/25/2016 19:32:45			
9	1/25/2016 19:45:49	Test 8		
10	1/25/2016 19:49:53	Just testing		
11	1/25/2016 19:54:34	Another test		
12	1/26/2016 2:18:21	Another test		
13	1/28/2016 6:28:31	Another test		
14	1/28/2016 6:32:16			
15	1/28/2016 6:37:37			
16	1/28/2016 20:20:14	Another test		
17	1/28/2016 20:20:47	Another test		
18	1/28/2016 20:22:46	Another test		
19	1/28/2016 20:24:41	Another test		
20				

Figure 4: My posts

Problem 2

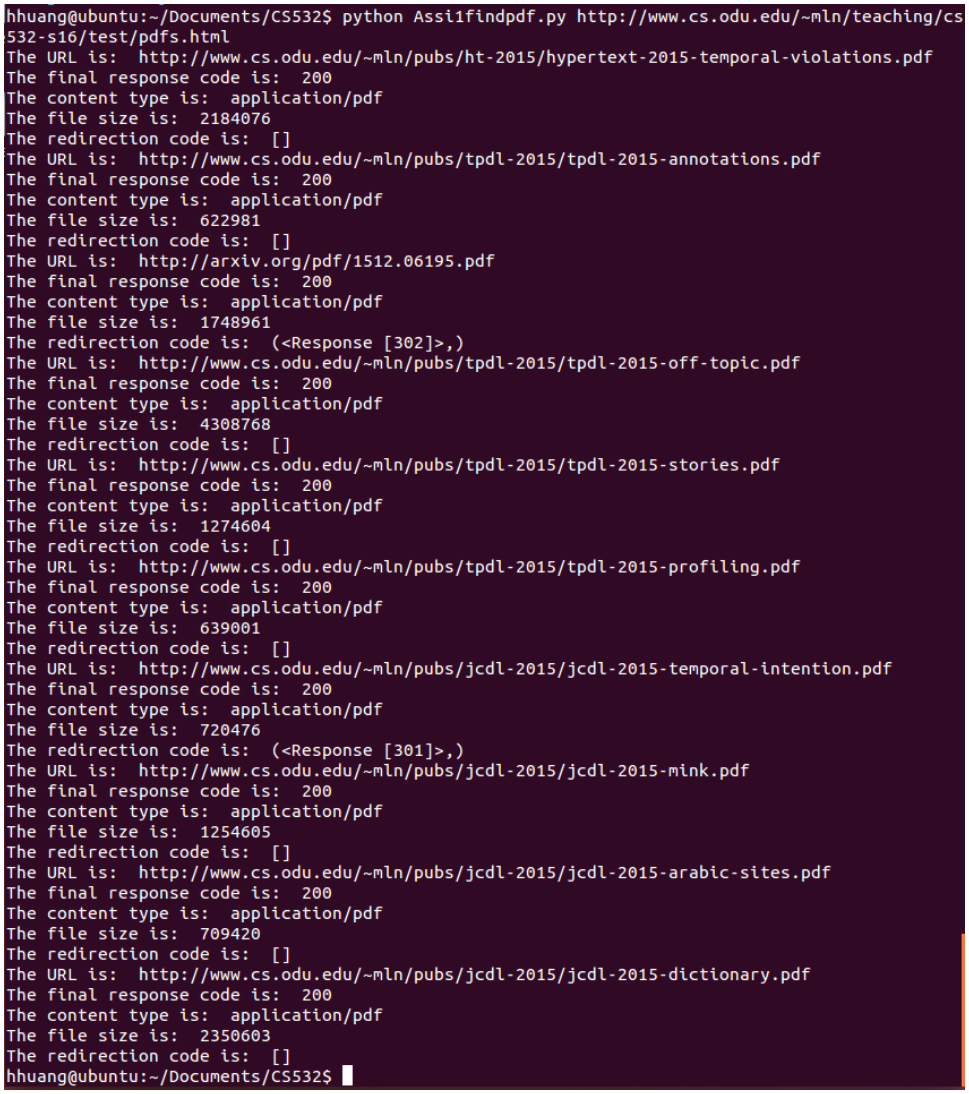
Write a Python program that:
1. takes as a command line argument a web page
2. extracts all the links from the page
3. lists all the links that result in PDF files, and prints out the bytes for each of the links. (note: be sure to follow all the redirects until the link terminates with a "200 OK".)
4. show that the program works on 3 different URIs, one of which needs to be:
http://www.cs.odu.edu/ mln/teaching/cs532-s16/test/pdfs.html

Answer

At this point, I would like to state that I received some help from my fellow class mate Zetan Li. He pointed me to a web page that made everything so much easier(`http://docs.python-requests.org/en/latest/api/#requests.Response`).

I used import sys to enable me to take command line argument of a web page. The web page url is assigned to the object named weburl, argument 1 is used because argument 0 is always preserved for the file name. I send a get request for the source information of the web page. Then I convert whatever I got from the get request into unicode and run them through beautifulsoup. From the result I find all "a" tag to get all the hyperlinks of the page and assign the result in object named links. Then, from the "a" tagged link results I get every "herf" links and for each one of them, call the function getHeader(). In this function, I use request.get command again, which is so helpful in this assignment. Because the get command with the request library will automatically chase down the redirected links until it receives a 200 ok code. Then, I use a if statement to isolate the links that have a content type of application/pdf in its response header. For the ones that fit all the criteria, program will print out the url link, response codes, and file size.

The first web page I used to test against my program is: `http://www.cs.odu.edu/~mln/teaching/cs532-s16/test/pdfs.html`. From which my program found 10 url links that directed to pdf files.



```
hhuang@ubuntu:~/Documents/CS532$ python Assi1findpdf.py http://www.cs.odu.edu/~mln/teaching/cs
532-s16/test/pdfs.html
The URL is:  http://www.cs.odu.edu/~mln/pubs/ht-2015/hypertext-2015-temporal-violations.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  2184076
The redirection code is:  []
The URL is:  http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-annotations.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  622981
The redirection code is:  []
The URL is:  http://arxiv.org/pdf/1512.06195.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  1748961
The redirection code is:  (<Response [302]>,)
The URL is:  http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-off-topic.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  4308768
The redirection code is:  []
The URL is:  http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-stories.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  1274604
The redirection code is:  []
The URL is:  http://www.cs.odu.edu/~mln/pubs/tpdl-2015/tpdl-2015-profiling.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  639001
The redirection code is:  []
The URL is:  http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-temporal-intention.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  720476
The redirection code is:  (<Response [301]>,)
The URL is:  http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-mink.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  1254605
The redirection code is:  []
The URL is:  http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-arabic-sites.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  709420
The redirection code is:  []
The URL is:  http://www.cs.odu.edu/~mln/pubs/jcdl-2015/jcdl-2015-dictionary.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  2350603
The redirection code is:  []
hhuang@ubuntu:~/Documents/CS532$
```

Figure 5: The required web page

The second web page I used to test against my program is: `http://www.cs.odu.edu/`. From which my program found 4 url links that directed to pdf files.

```
hhuang@ubuntu:~/Documents/CS532$ python Assi1findpdf.py http://www.cs.odu.edu/
The URL is:  http://www.cs.odu.edu/studentappointmentinfo.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  636560
The redirection code is:  []
The URL is:  http://www.cs.odu.edu/StrategicPlan0515_2010.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  909323
The redirection code is:  []
The URL is:  http://www.cs.odu.edu/files/cs_systems_services.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  412031
The redirection code is:  []
The URL is:  http://www.cs.odu.edu/files/csintroductioninfo.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  564602
The redirection code is:  []
hhuang@ubuntu:~/Documents/CS532$
```

Figure 6: ODU Computer Science Department main page

The last web page I used to test against my program is: `https://graduate.cs.odu.edu/ms/Getting_Started`. From which my program also found 4 url links that directed to pdf files.

```
hhuang@ubuntu:~/Documents/CS532$ python Assi1findpdf.py https://graduate.cs.odu.edu/ms/Getting
_Started
The URL is:  https://graduate.cs.odu.edu/files/Spring2016-NewStudentInfo.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  691948
The redirection code is:  []
The URL is:  https://graduate.cs.odu.edu/files/newMSPhD-gathering-aug2015.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  2122916
The redirection code is:  []
The URL is:  https://graduate.cs.odu.edu/files/MSgathering-Fallcourses-mar2015.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  1952191
The redirection code is:  []
The URL is:  https://graduate.cs.odu.edu/files/newMSPhD-gathering-jan2015.pdf
The final response code is:  200
The content type is:  application/pdf
The file size is:  1368151
The redirection code is:  []
hhuang@ubuntu:~/Documents/CS532$
```

Figure 7: Essential Resources page for ODU Computer Science Master degree students

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$\longrightarrow$B
B$\longrightarrow$C
C$\longrightarrow$D
C$\longrightarrow$A
C$\longrightarrow$G
E$\longrightarrow$F
G$\longrightarrow$C
G$\longrightarrow$H
I$\longrightarrow$H
I$\longrightarrow$J
I$\longrightarrow$K
J$\longrightarrow$D
L$\longrightarrow$D
M$\longrightarrow$A
M$\longrightarrow$N
N$\longrightarrow$D
O$\longrightarrow$A
P$\longrightarrow$G

For the above graph, give the values for:
IN:
SCC:
OUT:
Tendrils:
Tubes:
Disconnected:

Answer

For this problem, I first created a graph to help me decide the node types.

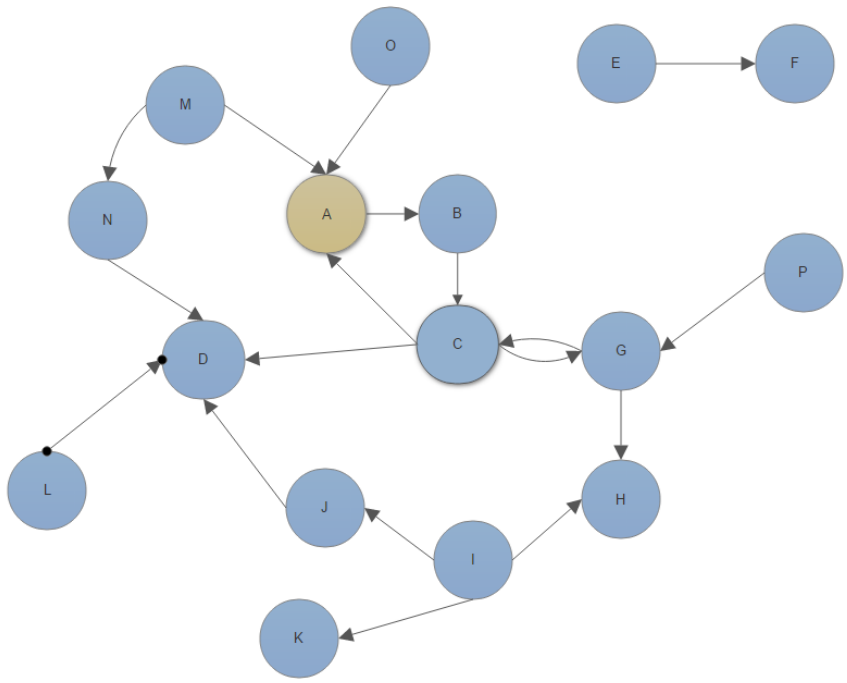


Figure 8: Problem 3 Graph

Now I apply the definitions of the node types:
IN: Nodes that only have links going out and connected directly to SCC nodes.
Out: Nodes that only have links going in and connected directly to SCC nodes.
SCC: Nodes that have links coming in from IN nodes and links going out to OUT nodes. Any SCC nodes can reach any other SCC nodes.
Tendrils: Nodes that are hanging from IN nodes or OUT node.
Tubes: Nodes that connect from IN to OUT without going through SCC.
Disconnected: Nodes that are not connected with main body of node networks.

The node types according to the graph:
IN: M
SCC: A B C G
OUT: D H
Tendrils: I J K L O P
Tubes: N
Disconnected: E F