

A1

Kevin R. Clemmons

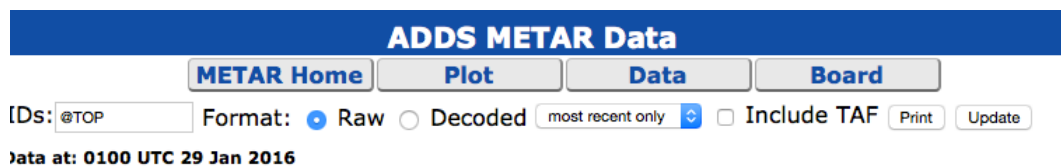
October 2015

1 Curl Demonstration

For the curl assignment, I used the aviation digital weather service to acquire a METAR for Norfolk International Airport which has the ICAO Code: KORF. The following curl command was used:

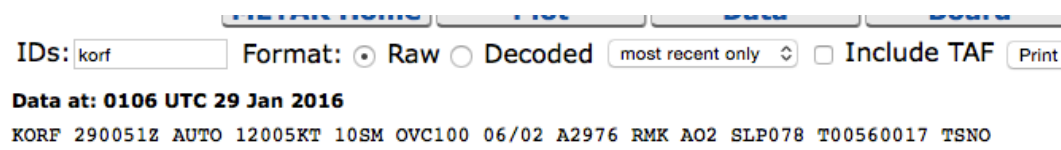
```
curl -d "ids=korf&format=rawdata=0&hrs=0" "https://www.aviationweather.gov/metar/data"> resul.html
```

1.1 Form Before



The screenshot shows the 'ADDs METAR Data' web form. At the top is a blue header with the title. Below it are four tabs: 'METAR Home', 'Plot', 'Data', and 'Board'. The 'Data' tab is selected. The form contains an 'IDs:' field with '@TOP' entered, a 'Format:' section with 'Raw' selected (indicated by a blue dot), 'Decoded' unselected, a 'most recent only' dropdown menu, an 'Include TAF' checkbox, and 'Print' and 'Update' buttons. At the bottom, it says 'Data at: 0100 UTC 29 Jan 2016'.

1.2 Form After



This screenshot shows the same form after submission. The 'IDs:' field now contains 'korf'. The 'Data at:' timestamp has updated to '0106 UTC 29 Jan 2016'. Below the form, the raw METAR data is displayed: 'KORF 290051Z AUTO 12005KT 10SM OVC100 06/02 A2976 RMK AO2 SLP078 T00560017 TSNO'. A horizontal line is drawn below the data.

The html file for the result of the curl command is located in the folder called curl

2 PDF Link Extractor

2.1 Libraries Used

- urllib2
- urllib
- urlparse
- sys
- httpplib
- errno
- socket
- bs4

2.2 Compilation

- Not Needed

2.3 Running

- Torun, enter the command: `python pdfLinks.py |url-name|`

Notes

2.3.1 This program was tested with the following links

- <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-088-introduction-to-c-memory-management-and-c-object-oriented-programming-january-iap-2010/lecture-notes/>
- <http://www.cs.toronto.edu/~suzanne/publications.html>
- <http://www.cs.odu.edu/~mln/teaching/cs532-s16/test/pdfs.html>
- <http://www.cs.odu.edu/~mln/>
- <http://www.macs.hw.ac.uk/~jim/112MH1/tuts/>

If you would like a demonstration of the test cases, enter the command `pythonpdfLinks.py -d`

A captured output of the above test cases is located in the file called `results.txt`, which can be found in the `pdfLinks` folder.

2.3.2 Sources

- Exception handling in lines 42-45
 - url: <http://stackoverflow.com/questions/20568216/python-handling-socket-error-errno-104-connection-reset-by-peer>
 - Accessed: 1/28/15
- Exception handline for requests
 - url: <http://stackoverflow.com/questions/666022/what-errors-exceptions-do-i-need-to-handle-with-urllib2-request-urlopen>

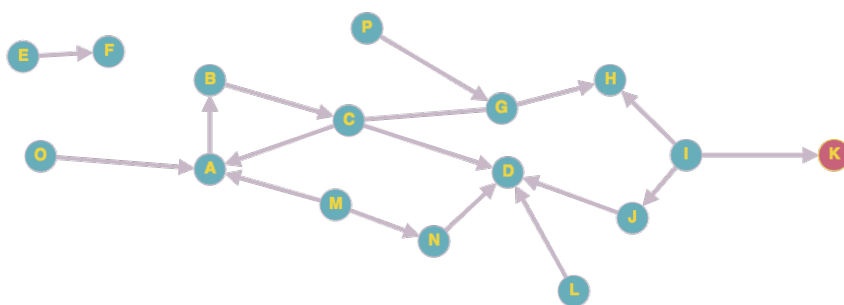
- Accessed: 1/24/15
- Handling link-redirections in line 62-77
 - url: <http://www.zacwitte.com/resolving-http-redirects-in-python>
 - Accessed: 1/28/15
- Handling incomplete links in line 152-153
 - url: <http://stackoverflow.com/questions/27397990/python-parsing-html-for-complete-links-urls>
 - * Accessed: 1/28/15

3 Graph

The graph below was generated by creating an adjacency matrix as seen below.

[illegible]

Once the matrix was created, the graph below was created from the following website: <http://graphonline.ru/en/>



3.1 Values for the Graph

The following vertexes make up the components of the graph.

- IN: O, P, L
- SCC: C
- OUT: K,H,G
- Tendrils: M
- Tubes: B,C,G,H
- Disconnected: E,I