Homework 7: Live feed

Due Wednesday, November 12, 3p

What to hand in:

- `hw7exercises.py`: Exercises that use a RESTful API to get airport status information
- `hw7application.py`: An application that uses a RESTful API of your choice.
- `hw7writeup.pdf`: A description of what your application does and sample output (see part 2)
- If you used your app to make a webpage, a chart, or some other output, please attach that as well. And feel free to share on the Facebook group.

In this assignment you will be processing live data from the Web. Six exercises will guide you through the process of writing an application that uses a RESTful API. It will cover basic building blocks that are used in any application that uses data from the Web: constructing queries, downloading data via HTTP, handling errors, and extracting relevant information. After you've completed the exercises you will be asked to use a RESTful API of your choice to make your own application.

Part 0: Get the homework files

You'll need the homework files. Get them by typing the following lines in terminal:

```
cd ~
./getHW.sh 7
```

Don't forget that if you already have Eclipse open, you may need to right click on the homeworks project and "refresh" it, to make the new folder hw7 appear.

Finally, review grading comments/warnings from all previous homeworks and the related solutions. We have been pointing out bad habits that, if not fixed, will make this part of the quarter more difficult than it needs to be for you. Questions? *Please* come to office hours.

Part 1: Exercises.

Look at `hw7exercises.py`. It contains all of the instructions and guidance you need to do the exercises. A screenshot of what your program's output should look like is included on the last page of the assignment, but, now that we are using live data, your output *will* vary!

Part 2: Your own application.

Find a RESTful API that returns JSON-formatted data. If you would like a little more guidance, you can use the Flickr API (covered in class), or use an API being discussed on the Facebook group. You cannot use the API from the exercise.

You may also want to check Canvas for the list of API's we covered during the brainstorming session: https://canvas.uw.edu/courses/916351/pages/List%20of%20APIs. Not all are RESTful, so if you are unsure, please ask. Also, we *strongly* recommend that, for this first API assignment, you stay clear of APIs that require authentication and the complications it will introduce unless you have previously worked with oAuth.

Many services (like the Flickr API) require that you register for an API key, so you may have to go through a registration process before you can start using their API. **You are strongly encouraged to share any of your experiences with APIs on the HCDE310 Facebook group: what steps you had to follow to create an account (if any), what problems you came across, and what queries you used to gather interesting data. But please don't share complete code for using an API.**

The assignment is fairly open ended, but your application should have the following minimum requirements:

1) **Handle any errors due to HTTP or connection related exceptions** (e.g., use `try/except`).

2) **Have at least one function that takes in at least one string as a parameter and uses the string to invoke a call to a RESTful API of your choosing**. The function should return a dictionary or list with data from the API call.

3) **Have at least one function that extracts and outputs information of interest. This information may either be printed out (which is easiest), or written to a CSV-formatted file (which you can then plot in Google Docs)**. For example, a function that fulfills this requirement might take in a stock symbol as a parameter and prints its price. A more complicated function or set of functions might write out a file with the price of the stock over the past 30 days.

4) **Repeatedly call one or more of the functions in (2) *or* (3) using data from a list of values.** For example, your code might contain a list of your favorite restaurants in Seattle. Your program would then iterate over this list and print out a review for each restaurant.

Put your application in a new Python file called hw7application.py. Include that file, along with a PDF (hw7writeup.pdf) that contains a short description of what your program does, a sample screenshot of your program's output, and any plots you might have generated. If you generated any output files, include those as well.

Yes. This is like a mini-project, and would mostly fulfill the requirements (though might not be so awesome). For the final project, though, we encourage you to (1) think about how your system adds value – not required for this assignment – and (2) to push yourself to make something that goes beyond the requirements, both to learn a bit more and to have a better entry in your portfolio.

Sample Exercise Output

Note: your output will vary because we are using live data. For example, when we did this two years ago, the NOAA weather data was down (too much traffic from Hurricane Sandy), so less information was available. Portions of the service were also unavailable during the government shutdown.

```
-------1a--------
format=json
-------1b--------
http://services.faa.gov/airport/status/SEA?format=json
-------2--------
{"delay":"false","IATA":"SEA","state":"Washington","name":"Seattle-Tacoma
International","weather":{"visibility":10.00,"weather":"Mostly
Cloudy","meta":{"credit":"NOAA's National Weather Service","updated":"12:53 PM
Local","url":"http://weather.gov/"},"temp":"56.0 F (13.3 C)","wind":"South at
4.6mph"},"ICAO":"KSEA","city":"Seattle","status":{"reason":"No known delays for
this
airport.","closureBegin":"","endTime":"","minDelay":"","avgDelay":"","maxDelay"
:"","closureEnd":"","trend":"","type":""}}
-------3--------
{
  "IATA": "SEA",
  "ICAO": "KSEA",
  "city": "Seattle",
  "delay": "false",
  "name": "Seattle-Tacoma International",
  "state": "Washington",
  "status": {
    "avgDelay": "",
    "closureBegin": "",
    "closureEnd": "",
    "endTime": "",
    "maxDelay": "",
    "minDelay": "",
    "reason": "No known delays for this airport.",
    "trend": "",
    "type": ""
  },
  "weather": {
    "meta": {
      "credit": "NOAA's National Weather Service",
      "updated": "12:53 PM Local",
```

```
      "url": "http://weather.gov/"
    },
    "temp": "56.0 F (13.3 C)",
    "visibility": 10.0,
    "weather": "Mostly Cloudy",
    "wind": "South at 4.6mph"
  }
}
-------4--------
Airport: Seattle-Tacoma International (Seattle, Washington)
Reason: No known delays for this airport.
-------5a--------
{
  "IATA": "SEA",
  "ICAO": "KSEA",
  "city": "Seattle",
  "delay": "false",
  "name": "Seattle-Tacoma International",
  "state": "Washington",
  "status": {
    "avgDelay": "",
    "closureBegin": "",
    "closureEnd": "",
    "endTime": "",
    "maxDelay": "",
    "minDelay": "",
    "reason": "No known delays for this airport.",
    "trend": "",
    "type": ""
  },
  "weather": {
    "meta": {
      "credit": "NOAA's National Weather Service",
      "updated": "12:53 PM Local",
      "url": "http://weather.gov/"
    },
    "temp": "56.0 F (13.3 C)",
    "visibility": 10.0,
    "weather": "Mostly Cloudy",
    "wind": "South at 4.6mph"
  }
}
-------5b--------
Airport: San Francisco International (San Francisco, California)
Reason: No known delays for this airport.
-------5c--------
Airport: Seattle-Tacoma International (Seattle, Washington)
Reason: No known delays for this airport.
Airport: General Edward Lawrence Logan International (Boston, Massachusetts)
Reason: No known delays for this airport.
Airport: John F Kennedy International (New York, New York)
Reason: No known delays for this airport.
Airport: Ronald Reagan Washington National (Washington, District of Columbia)
```

```
Reason: No known delays for this airport.
-------6a--------
We failed to reach a server
Reason Not Found
-------6b--------
Error trying to retrieve airport: xy
None
{u'status': {u'minDelay': u'', u'maxDelay': u'', u'trend': u'', u'reason': u'No
known delays for this airport.', u'closureEnd': u'', u'avgDelay': u'',
u'closureBegin': u'', u'endTime': u'', u'type': u''}, u'ICAO': u'KSEA',
u'name': u'Seattle-Tacoma International', u'city': u'Seattle', u'IATA': u'SEA',
u'delay': u'false', u'state': u'Washington', u'weather': {u'wind': u'South at
4.6mph', u'weather': u'Mostly Cloudy', u'meta': {u'url': u'http://weather.gov/',
u'credit': u"NOAA's National Weather Service", u'updated': u'12:53 PM Local'},
u'temp': u'56.0 F (13.3 C)', u'visibility': 10.0}}
-------6c--------
-------6d--------
Airport: Chicago OHare International (Chicago, Illinois)
Reason: No known delays for this airport.
Airport: Los Angeles International (Los Angeles, California)
Reason: No known delays for this airport.
Airport: John F Kennedy International (New York, New York)
Reason: No known delays for this airport.
Airport: Dallas/Ft Worth International (Dallas-Ft Worth, Texas)
Reason: No known delays for this airport.
Error trying to retrieve airport: NON-
```