Python for Informatics

Assignment 7

“Web Services”

Introduction: For this assignment, you can exercise 1 of 2 possible options. The first option is the “Google Geocoding” option. The second option is the “Alpha Vantage Stock Quote” option. If you complete and submit both options, you will receive 1.5 extra credit course points. The “Google Geocoding” option might require that you use your credit card to establish a developer’s account. They recently stopped requiring a credit card or an account, but they might reinstate their account and credit card policy at any time. If they do, Google will credit your account with $200 worth of service. You won’t need more than a small fraction of that amount to complete this assignment. In other words, whether Google requires an account along with a credit card or not, it shouldn’t cost you anything. Google has a vested interest in the developers that utilize their services, so they carefully guard your credit card information. However, since some students might not have a credit card, or might not feel comfortable developing software in association with Google, there is a second option for this assignment that does not require a credit card account—i.e., the “Alpha Vantage Stock Quote” option. If you are on the fence regarding which of the two options to choose, I recommend the Google Geocoding option—it’s perhaps the most fun and satisfying project in this course.

Option 1: “Google Geocoding”

Description:

1. Change the geojson.py program to print out the two-character country code from the retrieved data.
2. Add error checking—***try: except:***—so your program does not traceback if the country code is not there. Hint: “try” this code…

   ```
   names = {'wynken': 1, 'blynken': 5, 'nod': 42}

   print(names['Fred'])

   # Note that a keyerror is a kind of exception, however,

   # the exception you encounter in this assignment might not be a keyerror.
   ```

   As one more hint, if you look at the data carefully, you may find that the “country” tag is especially interesting.

3. Use the program to “try:” to retrieve the country code JSON element and make sure that it can handle locations that do not exist (such as “Pacific Ocean”). By the way, if the location does not exist, then the data structure element won’t either (that’s yet another hint!).
4. Your “except:” will handle non-existing locations by printing the message “There is no country code for that location.”.
5. Your program does not need to print the entire JSON structure.
6. Use the program to search for “San Diego” and “Pacific Ocean” and take a screenshot of the results.

You can find the documentation for the *Google Maps Platform Web Services Geocoding API* here: https://developers.google.com/maps/documentation/geocoding/start. The default API key should work fine. However, if you register with a credit card account to obtain an your own API key, you’ll also need to update the textbook Geocode example with your API key information. Regardless of how you obtain your API key, at the time that you enable the *Google Maps Platform* you’ll also need to pick the *Places API* as your “product”. If you click on the “Get started” button at the above URL, you will be guided through the step by step process of 1) Picking your product (*Places API*), 2) Selecting a project, and 3) Setting up your billing.

Deliverables:

Two files as attachments at our course shell assignment page. The first file should be the Python ***.py*** that you chose to modify. The second file should be a screenshot image file (.png or .jpg) demonstrating the correct execution of your program. Your screenshot image should show the correct response for a location that has a country code (such as "San Diego"), as well as a response (using the message stipulated above) for a location that does not have a country code (such as "Pacific Ocean").

Submission Deadline:

Please see the course schedule in our syllabus for all assignment submission deadlines.

Option 2: "Alpha-Vantage"

Description:

1. Change the geojson.py program to print out the "05. price" from the retrieved data.
2. Add error checking—***try: except:***—so your program does not traceback if the "05. price" is not there. You may not use an if/else construct for this logic.
3. Use the program to "try:" to retrieve the "05. price" JSON element and make sure that it can handle stock ticker symbols that do not exist (such as "FAKE").
4. Your "except:" will handle non-existing stock tickers symbols by printing a message such as "There is no price for that stock.".
5. Your program does not need to print the entire JSON structure.
6. Use the program to search for "WMT" and "FAKE" and take a screenshot of the results.

You can find the documentation for the *Alpha-Vantage Stock Time Series Quote Endpoint API* here: https://www.alphavantage.co/documentation/. Specifically, you'll be looking at the *Quote Endpoint API* instructions and examples. Use the Stock Time Series Quote Endpoint to obtain the latest price and volume information for the stock tickers "WMT" and "FAKE". After receiving the result of your API query, you'll want to retrieve the "Global Quote", "05. price" JSON element.

Deliverables:

Two files as attachments at our course shell assignment page. The first file should be the Python ***.py*** that you chose to modify. The second file should be a screenshot image file (.png or .jpg) demonstrating the correct execution of your program. Your screenshot image should show the correct response for stock ticker symbol “WMT”, as well as a response for the stock ticker symbol “FAKE” (which does not exist).

Submission Deadline:

Please see the course schedule in our syllabus for all assignment submission deadlines.

Peerwise Reminder:

This is not a reminder.