EmergingNotebook_slidesShow

May 20, 2020

Install and Verify Python

- Go to https://www.python.org/downloads and download the most recent version of Python 3.

Install Python modules that will be used in the course

- pip install requests
- pip install tabulate

Install Postman

- Postman is available as an application for Windows, Mac, and Linux operating systems. To install Postman, go to the apps page at https://www.getpostman.com/apps and click Download for your OS.
-

Install JSONView

- Install the JSONView Chrome extension.!
  - Click here to access the JSONView extension and add it to Chrome. After installation, you should see the JSONView icon in the top right corner of your Chrome browser

Test the JSONView extension

- Use the International Space Station Pass Predictions link below verify that your JSONView extension is working properly. You should be able to collapse and expand sections of the JSON data. http://api.open-notify.org/iss/v1/?lat=30.26715&lon=-97.74306

1 Demo -

Parsing JSON with a Python Application

Objectives

- Obtain a MapQuest API Key.
- Import necessary modules.
- Create API request variables and construct a URL.
- Add user input functionality.
- Add a quit feature so that the user can end the application.

- Display trip information for time, distance, and fuel usage.
- Iterate through the JSON data to extract and output the directions.
- Display error messages for invalid user input.

https://developer.mapquest.com/documentation/directions-api/route/get/

Note: If the above link no longer works, search for "MapQuest API Documentation".

2 Step2: Authenticating a RESTful request.

Authenticating a RESTful request is done in one of four ways. Describe each method below: ### - None: - The API resource is public and anybody can place the request. This is the method you have used up to this point.

2.0.1 - Basic HTTP:

- The username and password are passed to the server in an encoded string. This method is less common than token and OAuth authentication.

2.0.2 - Token:

```
- A secret key generally retrieved from the Web API developer portal.
```

2.0.3 - Open Authorization (OAuth):

- An open standard for retrieving an access token from an Identity Provider. The token is then passed with each API call.

```
For the MapQuest API, you will use token authentication.
```

3 Step 3: Get a MapQuest API key

Complete the following steps to get a MapQuest API key: 1. Go to: https://developer.mapquest.com/. 2. Click Sign Up at the top of the page. 3. Fill out the form to create a new account. For Company, enter Cisco Networking Academy Student. 4. After clicking Sign Me Up, you are redirected to the Manage Keys page. 5. Click Approve All Keys and expand My Application. 6. Copy your Consumer Key to Notepad for future use. This will be the key you use for the rest of this lab. Note: MapQuest may change the process for obtaining a key. If the steps above are no longer valid, search for "steps to generate mapquest api key".

3.0.1 Step 4: Importing modules for the application.

```
[ ]: # i have a module(file) i created called key.py and has a key variable that␣
     ↪contains my key
     from key import key

     import urllib.parse
     import requests
```

```
key = key
```

3.0.2 Step 5: Create variables for API request

Create variables to build the URL that will be sent in the request

```
[ ]: main_api = "https://www.mapquestapi.com/directions/v2/route?"
     orig = "Washington"
     dest = "Baltimaore"
     key = key
```

Combine the four variables to format the requested URL

```
[ ]: # Use the urlencode method to properly format the address value.
     # This function builds the parameters part of the URL and converts possible␣
      ↪special characters in the address value
     # (e.g. space into "+" and a comma into "%2C")

     url = main_api + urllib.parse.urlencode({"key": key, "from":orig, "to":dest})
```

Create a variable to hold the reply of the requested URL and print the returned JSON data

```
[ ]: json_data = requests.get(url).json()
     print(json_data)
```

Task 1

```
[ ]: #Replace 'your_api_key' with your MapQuest API key

     import urllib.parse
     import requests
     from key import key

     main_api = "https://www.mapquestapi.com/directions/v2/route?"
     orig = "Washington"
     dest = "Baltimaore"
     key = key

     url = main_api + urllib.parse.urlencode({"key": key, "from":orig, "to":dest})

     json_data = requests.get(url).json()
     print(json_data)
```

```
[ ]:
```

```
[ ]:
```

3.1 Task 2

[ ]:
```
import urllib.parse
import requests
from key import key

main_api = "https://www.mapquestapi.com/directions/v2/route?"
orig = "Washington"
dest = "Baltimore"
key = key

url = main_api + urllib.parse.urlencode({"key": key, "from":orig, "to":dest})
print("URL: " + (url))

json_data = requests.get(url).json()

##########################################################################
### EXTRA
##########################################################################

json_status = json_data["info"]["statuscode"]

if json_status == 0:
    print("API Status: " + str(json_status) + " = A successful route call.\n")
```

[ ]:
```

```

3.1.1 Task 3

[ ]:
```
import urllib.parse
import requests
from key import key

main_api = "https://www.mapquestapi.com/directions/v2/route?"
key = key

##########################################################################
### EXTRA
##########################################################################

while True:
    orig = input("Starting Location: ")
    dest = input("Destination: ")
    url = main_api + urllib.parse.urlencode({"key": key, "from":orig, "to":
 ↪dest})
    print("URL: " + (url))
    json_data = requests.get(url).json()
    json_status = json_data["info"]["statuscode"]

    if json_status == 0:
        print("API Status: " + str(json_status) + " = A successful route call.
↪\n")
```

3.1.2 Task 4

```
[ ]: # 08_json-parse4.py
     import urllib.parse
     import requests
     from key import key

     main_api = "https://www.mapquestapi.com/directions/v2/route?"
     key = key

     while True:
         orig = input("Starting Location: ")
         ###########################################################################
         ### EXTRA
         ###########################################################################
         if orig == "quit" or orig == "q":
             break
         dest = input("Destination: ")
         if dest == "quit" or dest == "q":
             break
         ###########################################################################
         ### EXTRA
         ###########################################################################

         url = main_api + urllib.parse.urlencode({"key": key, "from":orig, "to":
     ↪dest})
         print("URL: " + (url))

         json_data = requests.get(url).json()
         json_status = json_data["info"]["statuscode"]

         if json_status == 0:
             print("API Status: " + str(json_status) + " = A successful route call.
     ↪\n")
```

```
[ ]:
```

3.1.3 Task 5

- Copy your URL into your web browser. If you collapse all the JSON data, you will see that there are two root dictionaries: route and info.

[ ]:
```
print(url)
```

Task 6

[ ]:
```
# 08_json-parse5.py

import urllib.parse
import requests
from key import key

main_api = "https://www.mapquestapi.com/directions/v2/route?"
key = key

while True:
    orig = input("Starting Location: ")
    if orig == "quit" or orig == "q":
        break

    dest = input("Destination: ")
    if dest == "quit" or dest == "q":
        break

    url = main_api + urllib.parse.urlencode({"key": key, "from":orig, "to":
 ↪dest})
    print("URL: " + (url))

    json_data = requests.get(url).json()
    json_status = json_data["info"]["statuscode"]

    if json_status == 0:
        print("API Status: " + str(json_status) + " = A successful route call.
 ↪\n")

        ␣
 ↪##########################################################################
        ### EXTRA
        ␣
 ↪##########################################################################

        print("Directions from " + (orig) + " to " + (dest))
        print("Trip Duration:   " + (json_data["route"]["formattedTime"]))
        print("Kilometers:      " + str("{:.2f}".
 ↪format((json_data["route"]["distance"])*1.61)))
```

```
        print("Fuel Used (Ltr): " + str("{:.2f}".
↪format((json_data["route"]["fuelUsed"])*3.78)))
        print("=============================================")
```

[ ]:

```
```

[ ]:

```
'''
OUTPUT
Starting Location: Washington
Destination: Baltimore
URL: https://www.mapquestapi.com/directions/v2/route?
↪key=your_api_key&to=Baltimore&from=Washington
API Status: 0 = A successful route call.

Directions from Washington to Baltimore
Trip Duration:    00:49:19
Kilometers:       61.32
Fuel Used (Ltr): 6.24
=============================================
Starting Location: q
>>>
'''
```

Task 7

[ ]:

```
# 08_json-parse6.py
import urllib.parse
import requests
from key import key

main_api = "https://www.mapquestapi.com/directions/v2/route?"
key = key

while True:
    orig = input("Starting Location: ")
    if orig == "quit" or orig == "q":
        break

    dest = input("Destination: ")
    if dest == "quit" or dest == "q":
        break

    url = main_api + urllib.parse.urlencode({"key": key, "from":orig, "to":
↪dest})
    print("URL: " + (url))

    json_data = requests.get(url).json()
    json_status = json_data["info"]["statuscode"]

    if json_status == 0:
        print("API Status: " + str(json_status) + " = A successful route call.
↪\n")
        print("Directions from " + (orig) + " to " + (dest))
        print("Trip Duration:   " + (json_data["route"]["formattedTime"]))
        print("Kilometers:      " + str("{:.2f}".
↪format((json_data["route"]["distance"])*1.61)))
        print("Fuel Used (Ltr): " + str("{:.2f}".
↪format((json_data["route"]["fuelUsed"])*3.78)))
        print("=============================================")

       ␣
↪###########################################################################
        ### EXTRA
       ␣
↪###########################################################################

        for each in json_data["route"]["legs"][0]["maneuvers"]:
            print((each["narrative"]) + " (" + str("{:.2f}".
↪format((each["distance"])*1.61) + " km)"))
        print("=============================================\n")
```

[ ]:

```
```

[ ]:

```
'''
OUTPUT
Starting Location: Washington
Destination: Baltimore
URL: https://www.mapquestapi.com/directions/v2/route?
 ↪key=your_api_key&to=Baltimore&from=Washington
API Status: 0 = A successful route call.

Directions from Washington to Baltimore
Trip Duration:   00:49:19
Kilometers:      61.32
Fuel Used (Ltr): 6.24
=============================================
Start out going north on 6th St/US-50 E/US-1 N toward Pennsylvania Ave/US-1 Alt␣
 ↪N. (1.28 km)
Turn right onto New York Ave/US-50 E. Continue to follow US-50 E (Crossing into␣
 ↪Maryland). (7.51 km)
Take the Balt-Wash Parkway exit on the left toward Baltimore. (0.88 km)
Merge onto MD-295 N. (50.38 km)
Turn right onto W Pratt St. (0.86 km)
Turn left onto S Calvert St/MD-2. (0.43 km)
Welcome to BALTIMORE, MD. (0.00 km)
=============================================

Starting Location: Washington
Destination: Beijing
URL: https://www.mapquestapi.com/directions/v2/route?
↪to=Beijing&key=your_api_key&from=Washington
Starting Location: Washington
Destination: Balt
URL: https://www.mapquestapi.com/directions/v2/route?
↪to=Balt&key=your_api_key&from=Washington
Starting Location: Washington
Destination:
URL: https://www.mapquestapi.com/directions/v2/route?
↪to=&key=your_api_key&from=Washington
Starting Location: q
>>>
'''
```

3.1.4 Task 8

[ ]:
```python
# 08_jsont-parse7.py

import urllib.parse
import requests
from key import key

main_api = "https://www.mapquestapi.com/directions/v2/route?"
key = key

while True:
    orig = input("Starting Location: ")
    if orig == "quit" or orig == "q":
        break

    dest = input("Destination: ")
    if dest == "quit" or dest == "q":
        break

    url = main_api + urllib.parse.urlencode({"key": key, "from":orig, "to":
↪dest})
    print("URL: " + (url))

    json_data = requests.get(url).json()
    json_status = json_data["info"]["statuscode"]
    if json_status == 0:
        print("API Status: " + str(json_status) + " = A successful route call.
↪\n")
        print("Directions from " + (orig) + " to " + (dest))
        print("Trip Duration:   " + (json_data["route"]["formattedTime"]))
        print("Kilometers:      " + str("{:.2f}".
↪format((json_data["route"]["distance"])*1.61)))
        print("Fuel Used (Ltr): " + str("{:.2f}".
↪format((json_data["route"]["fuelUsed"])*3.78)))
        print("=============================================")


        for each in json_data["route"]["legs"][0]["maneuvers"]:
            print((each["narrative"]) + " (" + str("{:.2f}".
↪format((each["distance"])*1.61) + " km)"))
        print("=============================================\n")

    ##########################################################################
    ### EXTRA
    ##########################################################################

    elif json_status == 402:
        ␣
↪print("\n*****************************************************************")
        print("Status Code: " + str(json_status) + "; Invalid user inputs for␣
↪one or both locations.")
        ␣
↪print("*****************************************************************\n")
    else:
        ␣
↪print("\n***********************************************************************")
        print("Status Code: " + str(json_status) + "; Refer to:")
        print("https://developer.mapquest.com/documentation/directions-api/
↪status-codes")
        ␣
↪print("***********************************************************************\n")
```

[ ]:

```
[ ]: """
OUTPUT
Starting Location: Washington
Destination: Beijing
URL: https://www.mapquestapi.com/directions/v2/route?
↪from=Washington&to=Beijing&key=your_api_key
```

```
****************************************************************
Staus Code: 402; Invalid user inputs for one or both locations.
****************************************************************

Starting Location: Washington
Destination: Balt
URL: https://www.mapquestapi.com/directions/v2/route?
 ↪from=Washington&to=Balt&key=your_api_key

************************************************************************
Staus Code: 602; Refer to:
https://developer.mapquest.com/documentation/directions-api/status-codes
************************************************************************

Starting Location: Washington
Destination:
URL: https://www.mapquestapi.com/directions/v2/route?
 ↪from=Washington&to=&key=your_api_key

************************************************************************
Staus Code: 611; Refer to:
https://developer.mapquest.com/documentation/directions-api/status-codes
************************************************************************

Starting Location: q
>>>
"""
```

Task 9 (FULL CODE)

[ ]:
```
# FULL CODE

import urllib.parse
import requests
from key import key

main_api = "https://www.mapquestapi.com/directions/v2/route?"
key = key

while True:
    orig = input("Starting Location: ")
    if orig == "quit" or orig == "q":
        break

    dest = input("Destination: ")
    if dest == "quit" or dest == "q":
        break

    url = main_api + urllib.parse.urlencode({"key": key, "from":orig, "to":
↪dest})
    print("URL: " + (url))

    json_data = requests.get(url).json()
    json_status = json_data["info"]["statuscode"]

    if json_status == 0:
        print("API Status: " + str(json_status) + " = A successful route call.
↪\n")
        print("Directions from " + (orig) + " to " + (dest))
        print("Trip Duration:   " + (json_data["route"]["formattedTime"]))
        print("Kilometers:      " + str("{:.2f}".
↪format((json_data["route"]["distance"])*1.61)))
        print("Fuel Used (Ltr): " + str("{:.2f}".
↪format((json_data["route"]["fuelUsed"])*3.78)))
        print("=============================================")
        for each in json_data["route"]["legs"][0]["maneuvers"]:
            print((each["narrative"]) + " (" + str("{:.2f}".
↪format((each["distance"])*1.61) + " km)"))
        print("=============================================\n")
    elif json_status == 402:
        ␣
↪print("\n**********************************************************************")
        print("Status Code: " + str(json_status) + "; Invalid user inputs for␣
↪one or both locations.")
        ␣
↪print("**********************************************************************\n")
    else:
        ␣
↪print("\n************************************************************************")
        print("Status Code: " + str(json_status) + "; Refer to:")
        print("https://developer.mapquest.com/documentation/directions-api/
↪status-codes")
        ␣
↪print("************************************************************************\n")
```

[ ]:

```
[ ]: """
Starting Location: Washington
Destination: Baltimore
```

```
URL: https://www.mapquestapi.com/directions/v2/route?
 ↪key=your_api_key&from=Washington&to=Baltimore
API Status: 0 = A successful route call.

Directions from Washington to Baltimore
Trip Duration:   00:49:19
Kilometers:      61.32
Fuel Used (Ltr): 6.24
=============================================
Start out going north on 6th St/US-50 E/US-1 N toward Pennsylvania Ave/US-1 Alt␣
 ↪N. (1.28 km)
Turn right onto New York Ave/US-50 E. Continue to follow US-50 E (Crossing into␣
 ↪Maryland). (7.51 km)
Take the Balt-Wash Parkway exit on the left toward Baltimore. (0.88 km)
Merge onto MD-295 N. (50.38 km)
Turn right onto W Pratt St. (0.86 km)
Turn left onto S Calvert St/MD-2. (0.43 km)
Welcome to BALTIMORE, MD. (0.00 km)
=============================================

Starting Location: Moscow
Destination: Beijing
URL: https://www.mapquestapi.com/directions/v2/route?
 ↪key=your_api_key&from=Moscow&to=Beijing
API Status: 0 = A successful route call.

Directions from Moscow to Beijing
Trip Duration:   84:31:10
Kilometers:      7826.83
Fuel Used (Ltr): 793.20
=============================================
Start out going west on             /Kremlin Embankment. (0.37 km)
Turn slight right onto ramp. (0.15 km)
Turn slight right onto           . (0.23 km)
[output omitted]
Turn left onto   /E. Guangchang Rd. (0.82 km)
   /E. Guangchang Rd becomes   /E. Chang'an Str. (0.19 km)
Welcome to BEIJING. (0.00 km)
=============================================

Starting Location: Washington
Destination: Beijing
URL: https://www.mapquestapi.com/directions/v2/route?
 ↪key=your_api_key&from=WashingtonTurn+right+onto+%E5%89%8D%E9%97%A8%E8%A5%BF%E5%A4%A7%E8%A1%
 ↪+%281.01+km%29&to=Beijing

****************************************************************
```

```
Staus Code: 402; Invalid user inputs for one or both locations.
****************************************************************

Starting Location: Washington
Destination: Balt
URL: https://www.mapquestapi.com/directions/v2/route?
 ↪key=your_api_key&from=Washington&to=Balt

************************************************************************
Staus Code: 602; Refer to:
https://developer.mapquest.com/documentation/directions-api/status-codes
************************************************************************

Starting Location: Washington
Destination:
URL: https://www.mapquestapi.com/directions/v2/route?
 ↪key=your_api_key&from=Washington&to=

************************************************************************
Staus Code: 611; Refer to:
https://developer.mapquest.com/documentation/directions-api/status-codes
************************************************************************

Starting Location: q
>>>
"""
```