

1 Refactoring Activity
2 Paul ReFalo 11/19/17

3
4 For this activity I choose chapter 7's assignment for the ISS viewing tim
5 when completed it the first time that I was reusing code to in two places to
6 API succeeded or failed. I refactored this code to using one method instead
7 code in two places.

8 I also choose to refactor the way I was checking for the API success/fail
9 had originally wrote called the API with get and then I checked to see if I g
10 in text form from that response. If so, this was judged a successful API cal
11 this, I used the demonstration code the instructor gave which makes use of th
12 rather than checking for response.text. This seems much cleaner than the way
13 originally done it. Below you will see the refactored code followed by the o
14 Original code starts on ~ line 108.

15
16 ===== New / Refactored Code =====

17
18 # CS 519 assignment 7 - ISS.py "When can I see the ISS" by Paul ReFalo
19 # View | Tool Windows | Terminal
20 # python3 ISSrefactored.py "Portland, OR" 3

21
22 import sys
23 import requests
24 import json
25 from pprint import pprint
26 import time

27
28 # Global variables to hold lat and long
29 lat = 0
30 long = 0

31
32 address = sys.argv[1] # address of location from user input
33 n = int(sys.argv[2]) # number of desired results

34
35 # get Request function takes url, headers, query and returns API response if
36 def getRequest(requestURL, requestHeaders, requestQueryString):
37 error_count = 0
38 success = False
39
40 while not success and error_count < 3:
41 #response = requests.request("GET", requestURL)
42 getResponse = requests.get(requestURL, headers=requestHeaders, params=r
43 if getResponse.status_code < 400:
44 success = True
45 else:

```
46         print("==== API failed.  Retrying it now...standby. =====")
47         error_count += 1
48
49     if success:
50         return getResponse
51     else:
52         print("Too many errors, giving up")
53         sys.exit()
54
55 # use input from user to get Lat and Long of address
56 googleUrl = "https://maps.googleapis.com/maps/api/geocode/json"      # set up
57 googleQueryString = {"address":address}                               # config
58 googleHeaders = { }                                                  # no head
59
60 googleResponse = getRequest(googleUrl, googleHeaders, googleQueryString)
61 googleResults = json.loads(googleResponse.text) # use loads to get json text
62
63
64 for gr in googleResults["results"]:                                     # loop and extract la
65     lat = gr["geometry"]["location"]["lat"]
66     long = gr["geometry"]["location"]["lng"]
67
68 # use lat and long to get ISS viewing times
69 issUrl = "http://api.open-notify.org/iss-pass.json"                  # set up ISS url
70 issHeaders = { }                                                    # no headers needed
71 issQueryString = {"lat":lat, "lon":long, "n":n}                      # config queryString
72
73 issResponse = getRequest(issUrl, issHeaders, issQueryString)
74 issResults = json.loads(issResponse.text) # use loads to get json text
75
76 issResultsArray = []                                                 # set up array to hold ISS results
77 for issR in issResults["response"]:                                   # loop to get individual results
78     duration = issR["duration"]                                       # extract duration in seconds
79     risetime = issR["risetime"]                                       # extract risetime in seconds since e
80     risetime = time.strftime('%a %b %d %H:%M:%S %Y', time.localtime(risetime))
81
82     issResultsArray.append((duration, risetime)) # append to results array
83
84 print("From " + address + " you will be able to see the ISS on:")
85 for idx, e in enumerate(issResultsArray):                             # loop to display results in
86     if idx >= n:                                                       # break if length(issResultsA
87         break                                                         # this shouldn't happen or be
88     print(str(e[1]) + " for " + str(e[0]) + " seconds")              # print results
89
90
```

```
91 '''
92 Output demo:
93 MacBook-Pro:week-7-files paulrefalo$ python ISS.py "Santa Cruz, CA" 3
94 From Salem, OR you will be able to see the ISS on:
95 Sun Nov 19 19:57:16 2017 for 352 seconds
96 Sun Nov 19 21:30:39 2017 for 622 seconds
97 Sun Nov 19 23:07:01 2017 for 633 seconds
98
99 if API fails you might get:
100 ===== API failed. Retrying it now...standby. =====
101 ===== API failed. Retrying it now...standby. =====
102 From Salem, OR you will be able to see the ISS on:
103 Sun Nov 19 19:57:16 2017 for 352 seconds
104 Sun Nov 19 21:30:39 2017 for 622 seconds
105 Sun Nov 19 23:07:01 2017 for 633 seconds
106
107 ===== Original Code =====
108
109 # CS 519 Refactoring assignment - ISSrefactored.py "When can I see the ISS" b
110
111 import sys
112 import requests
113 import json
114 from pprint import pprint
115 import time
116
117 # Global variables to hold lat and long
118 lat = 0
119 long = 0
120
121 address = sys.argv[1] # address of location from user input
122 n = int(sys.argv[2]) # number of desired results
123
124 # use input from user to get Lat and Long of address
125 googleUrl = "https://maps.googleapis.com/maps/api/geocode/json" # set up
126 googleQueryString = {"address":address} # config
127 googleHeaders = { } # no head
128
129 googleAPI = False
130 for i in range(5): # give API a few chances to succeed. Abort script in
131     if googleAPI:
132         break
133     for attempt in range(5):
134         try:
135             response = requests.get(googleUrl, headers=googleHeaders, params=
```

```
136         googleResults = json.loads(response.text) # use loads to get jso
137
138         if googleResults["results"]:
139             googleAPI = True
140             break
141     except ConnectionError:
142         continue
143     else:
144         print("==== Google API failed. Retrying it now...standby. ====")
145         break
146 else:
147     print("Google geolocation failed. Try script again or check connecti
148     sys.exit()
149
150 #print(response.text)
151
152
153 for gr in googleResults["results"]: # loop and extract la
154     lat = gr["geometry"]["location"]["lat"]
155     long = gr["geometry"]["location"]["lng"]
156
157 # use lat and long to get ISS viewing times
158 issUrl = "http://api.open-notify.org/iss-pass.json" # set up ISS url
159 issHeaders = { } # no headers needed
160 issQueryString = {"lat":lat, "lon":long, "n":n} # config queryString
161
162 issAPI = False
163 for i in range(5): # give API a few chances to succeed. Abort script in
164     if issAPI:
165         break
166     for attempt in range(5):
167         try:
168             issResponse = requests.get(issUrl, headers=issHeaders, params=iss
169             issResults = json.loads(issResponse.text) # use loads to get jso
170
171             if issResults["response"]:
172                 issAPI = True
173                 break
174         except ConnectionError:
175             continue
176         else:
177             print("==== ISS API failed. Retrying it now...standby. ====")
178             break
179     else:
180         print("ISS Api failed. Try script again or check connection.")
```

```
181         sys.exit()
182
183 #print(issResponse.text)
184
185 issResultsArray = []                # set up array to hold ISS results
186 for issR in issResults["response"]: # loop to get individual results
187     duration = issR["duration"]      # extract duration in seconds
188     risetime = issR["risetime"]      # extract risetime in seconds since e
189     risetime = time.strftime('%a %b %d %H:%M:%S %Y', time.localtime(risetime))
190
191     issResultsArray.append((duration, risetime)) # append to results array
192
193 print("You will be able to see the ISS on:")
194 for idx, e in enumerate(issResultsArray): # loop to display results in
195     if idx >= n:                          # break if length(issResultsA
196         break                             # this shouldn't happen or be
197     print(str(e[1]) + " for " + str(e[0]) + " seconds") # print results
198
199
200
201 '''
202 Output demo:
203 MacBook-Pro:week-7-files paulrefalo$ python ISS.py "Santa Cruz, CA" 3
204 You will be able to see the ISS on:
205 Sat Nov 04 10:52:18 2017 for 625 seconds
206 Sat Nov 04 12:28:58 2017 for 582 seconds
207 Sun Nov 05 01:59:17 2017 for 103 seconds
208
209 if API fails you might get:
210 ===== Google API failed.  Retrying it now...standby. =====
211 ===== Google API failed.  Retrying it now...standby. =====
212 You will be able to see the ISS on:
213 Sat Nov 04 10:52:18 2017 for 625 seconds
214 Sat Nov 04 12:28:58 2017 for 582 seconds
215 Sun Nov 05 01:59:17 2017 for 103 seconds
216
217 '''
218
219 '''
```