Harvest data from Web APIs using the Python Requests library

Amanda Devine
25 July 2019
SI Carpentries Brown Bag

GitHub Repository: https://github.com/amdevine/cbb-python-requests (https://github.com/amdevine/cbb-python-requests)
Detailed Jupyter notebook: https://github.com/amdevine/cbb-python-requests/blob/master/harvest-data-apis-python-requests.ipynb (https://github.com/amdevine/cbb-python-requests/blob/master/harvest-data-apis-python-requests.ipynb)

Definitions

- **(REST) API**: Application Programming Interface. A special page on a website that provides structured data for other programs and applications.
- **GET Request**: An HTTP command to retrieve code and data from a website.
- **JSON**: JavaScript Object Notation. A common format of structuring data, analogous to a Python dictionary.
- **Base URL**: The "home" website URL for all API data.

NPS Base URL:

```
https://developer.nps.gov/api/v1
```

- **Endpoint**: The specific URL where the API page can be found.

  Parks Endpoint:
  `https://developer.nps.gov/api/v1/parks`

- **Parameter**: An additional criterion that is added to the endpoint to filter data returned.

  parkCode, stateCode, and limit parameters:
  `https://developer.nps.gov/api/v1/parks?parkCode=yell&stateCode=WY&limit=5`

- **API Key**: A string of characters assigned by the website to identify the user requesting data via the API.

  National Parks API Key:
  `https://developer.nps.gov/api/v1/parks?api_key=1mdaBewB37R0kUA2ZtfA6URe7PeUsig6jLQmSXyx`
  (not a real key!)

NPS Data API

Official source of data about natural areas managed by the National Park Service

- park information
- campground information
- alerts, events, news, educational resources, etc.

NPS API Keys: https://www.nps.gov/subjects/developer/get-started.htm (https://www.nps.gov/subjects/developer/get-started.htm)

NPS Data API documentation: https://www.nps.gov/subjects/developer/api-documentation.htm (https://www.nps.gov/subjects/developer/api-documentation.htm)

Python Requests library

Sample GET Request:

```
import requests
url = 'https://baseurl.com/endpoint'
params = {
    'field1': 'value1',
    'field2': 'value2',
}
r = requests.get(url, params).json()
```

Quickstart documentation: https://2.python-requests.org/en/master/user/quickstart/ (https://2.python-requests.org/en/master/user/quickstart/)

Setup

Import the `requests` and `pandas` libraries.

In [1]:

```
import requests
import pandas as pd
```

Save API Key as a constant or read it from a local file.

In [2]:

```
# API_KEY = '1mdaBewB37R0kUA2ZtfA6URe7PeUsig6jLQmSXyx'
with open('api_key_file.txt', 'r') as f:
    API_KEY = f.read().strip()
print("API Key: {}".format("API_KEY")) # Remove quotes to display actual API_KEY
```

```
API Key: API_KEY
```

Make a GET request to the API to retrieve data

This request returns data on up to 100 parks in Washington DC, Maryland, and Virginia.

In [3]:

```
url = 'https://developer.nps.gov/api/v1/parks'
params = {
    'api_key': API_KEY,
    'stateCode': 'DC,MD,VA', # Per the API documentation, separate multiple values with commas
    'fields': 'entranceFees',
    'limit': 100
}
r = requests.get(url, params)
```

`api_key` is a required parameter for all NPS Data API requests. `stateCode` filters parks based on two-letter US state abbreviations. `fields` specifies additional fields to return in addition to the default fields. `limit` specifies the maximum number of results to return.

`requests.get()` returns a variety of information about the web page retrieved.

```
In [4]: print("The response code is: {}".format(r.status_code))
        print("\nThe retrieved URL is: {}".format("r.url")) #Remove quotes to display URL
        print("\nThe first 300 characters of the retrieved text are:\n{}".format(r.text[:300]))
```

```
The response code is: 200

The retrieved URL is: r.url

The first 300 characters of the retrieved text are:
{"total":"80","data":[{"states":"DC","entranceFees":[{"cost":"0.0000","descriptio
n":"No Entrance Fee to enter park site.","title":"No Entrance Fee"}],"directionsInf
o":"The memorial is located at the corner of Vermont Avenue, 10th St, and U Street N
W, near the U Street\/African-American Civil War Mem
```

Work with retrieved data

Convert GET request object to dictionary

In [5]:

```
parks_data = r.json()

print("Top level keys:", list(parks_data))
print("\nAvailable keys in each entry:", list(parks_data['data'][0]))
```

```
Top level keys: ['total', 'data', 'limit', 'start']

Available keys in each entry: ['states', 'entranceFees', 'directionsInfo', 'direction
sUrl', 'url', 'weatherInfo', 'name', 'latLong', 'description', 'designation', 'parkCo
de', 'id', 'fullName']
```

Create a DataFrame

This code filters the retrieved data to states and associated lat/long coordinate for each park.

In [6]:

```python
parks_df = pd.DataFrame(parks_data['data'])
locations_df = parks_df[['parkCode', 'fullName', 'designation', 'states', 'latLong']]
locations_df.head(10)
```

Out[6]:

	parkCode	fullName	designation	states	latLong
0	afam	African American Civil War Memorial		DC	lat:38.916554, long:-77.025977
1	anac	Anacostia Park	Park	DC	lat:38.89644397, long:-76.96314236
2	anti	Antietam National Battlefield	National Battlefield	MD	lat:39.46763452, long:-77.73828017
3	appa	Appalachian National Scenic Trail	National Scenic Trail	CT,GA,MA,MD,ME,NC,NH,NJ,NY,PA,TN,VA,VT,WV	lat:40.41029575, long:-76.4337548
4	apco	Appomattox Court House National Historical Park	National Historical Park	VA	lat:37.38022164, long:-78.79856982
5	arho	Arlington House, The Robert E. Lee Memorial		VA	lat:38.8822021484375, long:-77.0734786987305
6	asis	Assateague Island National Seashore	National Seashore	MD,VA	lat:38.05593172, long:-75.24524611
7	balt	Baltimore National Heritage Area	National Heritage Area	MD	lat:39.2904968261719, long:-76.6284027099609
8	bawa	Baltimore-Washington Parkway	Parkway	MD	lat:39.02604289, long:-76.85410921
9	bepa	Belmont-Paul Women's Equality National Monument	National Monument	DC	lat:38.89231541, long:-77.00381882

Restructure/flatten data

Retrieved JSON data for an individual park's multiple entrance fees.

```
In [7]: parks_data['data'][2]['entranceFees']
```

```
Out[7]: [{'cost': '7.0000',
          'description': '3 day pass - $7.00 per bike or motorcycle \nThis is the entry fee t
        o the battlefield proper, museum, movie, and ranger programs.',
          'title': 'Antietam National Battlefield Entrance Fee'},
         {'cost': '15.0000',
          'description': '3 day vehicle pass.  This pass covers everyone in a vehicle, ie. fa
        mily.  The pass covers entry to the battlefield proper, museum, movie, and ranger pro
        grams.',
          'title': 'Antietam National Battlefield Entrance Fee'}]
```

For each park in the dataset, and for each entrance fee in that park, add some park and fee values as a dictionary to a new `entry_fee_data` list.

In [8]:

```
entry_fees_data = []
for park in parks_data['data']:
    for fee in park['entranceFees']:
        entry_fees_data.append({
            'parkCode': park['parkCode'],
            'fullName': park['fullName'],
            'designation': park['designation'],
            'fee_usd': fee['cost'],
            'fee_type': fee['title'],
            'fee_description': fee['description']
        })
print(entry_fees_data[:3])
```

```
[{'parkCode': 'afam', 'fullName': 'African American Civil War Memorial', 'designatio
n': '', 'fee_usd': '0.0000', 'fee_type': 'No Entrance Fee', 'fee_description': 'No En
trance Fee to enter park site.'}, {'parkCode': 'anac', 'fullName': 'Anacostia Park',
'designation': 'Park', 'fee_usd': '0.0000', 'fee_type': 'Entrance Fees', 'fee_descrip
tion': 'There are no entrance fees to this park.'}, {'parkCode': 'anti', 'fullName':
'Antietam National Battlefield', 'designation': 'National Battlefield', 'fee_usd':
'7.0000', 'fee_type': 'Antietam National Battlefield Entrance Fee', 'fee_descriptio
n': '3 day pass - $7.00 per bike or motorcycle \nThis is the entry fee to the battlef
ield proper, museum, movie, and ranger programs.'}]
```

Convert `entry_fee_data` to a DataFrame

In [9]:

```
entry_fees_df = pd.DataFrame(entry_fees_data)
entry_fees_df = entry_fees_df[['parkCode', 'fullName', 'designation', 'fee_usd', 'fee_type']]
entry_fees_df['fee_usd'] = entry_fees_df['fee_usd'].astype(float)
entry_fees_df.head(10)
```

Out[9]:

	parkCode	fullName	designation	fee_usd	fee_type
0	afam	African American Civil War Memorial		0.0	No Entrance Fee
1	anac	Anacostia Park	Park	0.0	Entrance Fees
2	anti	Antietam National Battlefield	National Battlefield	7.0	Antietam National Battlefield Entrance Fee
3	anti	Antietam National Battlefield	National Battlefield	15.0	Antietam National Battlefield Entrance Fee
4	appa	Appalachian National Scenic Trail	National Scenic Trail	0.0	Appalachian National Scenic Trail Entrance Fee
5	apco	Appomattox Court House National Historical Park	National Historical Park	0.0	Entrance Fee
6	arho	Arlington House, The Robert E. Lee Memorial		0.0	No Fee
7	asis	Assateague Island National Seashore	National Seashore	20.0	Assateague 7 day per vehicle pass
8	asis	Assateague Island National Seashore	National Seashore	20.0	Chincoteague National Wildlife Refuge Weekly Pass
9	balt	Baltimore National Heritage Area	National Heritage Area	0.0	Baltimore National Heritage Area

Export data as a tabular file

CSV file: `df_name.to_csv('output_file_name.csv', index=False)`

TSV file: `df_name.to_csv('output_file_name.tsv', sep='\t', index=False)`

In [10]:

```
locations_df.to_csv('parks_data.tsv', sep='\t', index=False)
entry_fees_df.to_csv('parks_entry_fees.tsv', sep='\t', index=False)
```

Additional API Resources

Full Requests documentation: https://2.python-requests.org/en/master/ (https://2.python-requests.org/en/master/)

List of US Federal Government APIs: https://catalog.data.gov/dataset?res_format=API (https://catalog.data.gov/dataset?res_format=API)

Repository of APIs: https://www.programmableweb.com/ (https://www.programmableweb.com/)