APIs

CST 205

Application Programming Interfaces (APIs)

- A defined set of instructions you can send to a web service
- APIs allow developers to write programs that connect with 3rd party services
- Many of the biggest web applications have APIs: Facebook, Twitter, YouTube, Google Maps, and many more.
    - These APIs allow developers to connect to these applications to use the apps' data in their own development work.

Every API is different

- Important to read the documentation
- Test API *calls* to get the data
- Understand the structure of the data you get back
- Usually, you will get back so-called `JSON`

JavaScript Object Notation (`JSON`)

- A lightweight format used for data interchange
- Specified in RFC 4627 by Douglas Crockford
- `JSON` dethroned Extensible Markup Language (XML) as the preferred format for web services.
- `XML` is “heavier” (more verbose and structured)

What does `JSON` look like?

- Data is organized into attributes and values, each separated by a colon

```
{
  "people" :
  [
    {"firstName" : "Radia", "lastName" : "Perlman"},
    {"firstName" : "Konrad", "lastName" : "Zuse"},
    {"firstName" : "Barbara", "lastName" : "Liskov"}
  ]
}
```

This is a JavaScript array object called `people` containing three objects.

Python and `JSON`

- `JSON` looks similar to Python dictionaries and lists
- Python provides excellent support for `JSON`
- Add the following: `import json`
- Use `json.loads()` to convert a string to a `JSON` object
  - Use `json.load()` to load `JSON` from a file.

Example

- Say we want to load the `JSON` object from our file `cs_people.json` :

```
import json

data = json.load(open('cs_people.json'))

print(data['people'][0]['firstName'])

for person in data['people']:
    print(f'{person["firstName"]}', end="")
    print(f' {person["lastName"]} ', end="")
    print('is a famous Computer Scientist.', end="")
    print("")
```

Python and APIs

- We will be using the `Requests` library
    - Claimed to be “the only Non-GMO HTTP library for Python, safe for human consumption.”
- Many good examples in the `Requests` docs
- Install (with your virtual environment activated) with:

```
pip install requests
```

Let’s explore an API

- National Aeronautics and Space Administration (NASA) has a wide breadth of data (including) imagery via various APIs.

- We’ll try out NASA’s Astronomy Picture of the Day (APOD).

Example

```
import requests, json
from pprint import pprint

my_key = 'D8FJrAVDcE5RHJ29uwD5lRftLXMDO6Tw3iGnj19V'

payload = {
  'api_key': my_key,
  'start_date': '2017-03-09',
  'end_date': '2017-03-11'
}

endpoint = 'https://api.nasa.gov/planetary/apod'

try:
    r = requests.get(endpoint, params=payload)
    data = r.json()
    pprint(data)

except:
    print('please try again')
```

Request's JSON decoder

Data from example (abridged)


```
[{'copyright': 'Fabian Neyer',
  'date': '2017-03-09',
  'explanation': 'Only 11 million light-years away, Centaurus A is the closest...',
  'hdurl': 'https://apod.nasa.gov/apod/image/1703/ngc5128neyer_l1.jpg',
  'media_type': 'image',
  'service_version': 'v1',
  'title': 'Centaurus A',
  'url': 'https://apod.nasa.gov/apod/image/1703/ngc5128neyer1024_l1.jpg'},
 {'copyright': 'Bernhard Hubl',
  'date': '2017-03-10',
  'explanation': 'The galaxies of Abell 2666 lie far beyond the Milky Way...',
  'hdurl': 'https://apod.nasa.gov/apod/image/1703/abell2666-1hubl_full.jpg',
  'media_type': 'image',
  'service_version': 'v1',
  'title': 'Galaxy Cluster Abell 2666',
  'url': 'https://apod.nasa.gov/apod/image/1703/abell2666-1hubl_full1024.jpg'}]
```