Data Foundations: The Web

Instructor: Anthony Rios

Outline

Introduction

https://www.youtube.com/watch?v=I6nu8N6FQqc

File IO on the Web

example.py

```
import urllib.request

fhand = urllib.request.urlopen('http://data.pr4e.org/romeo.txt')
for line in fhand:
        print(line.decode().strip())
```

B

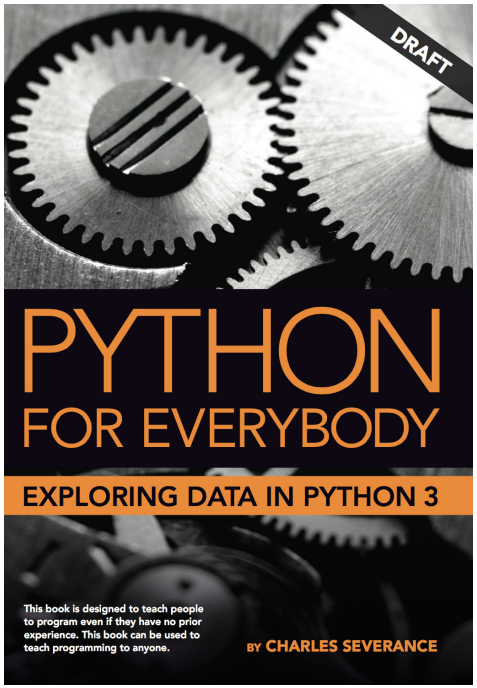
```
But soft what light through yonder window breaks
It is the east and Juliet is the sun
Arise fair sun and kill the envious moon
Who is already sick and pale with grief
```

Reading binary files using urllib

example.py

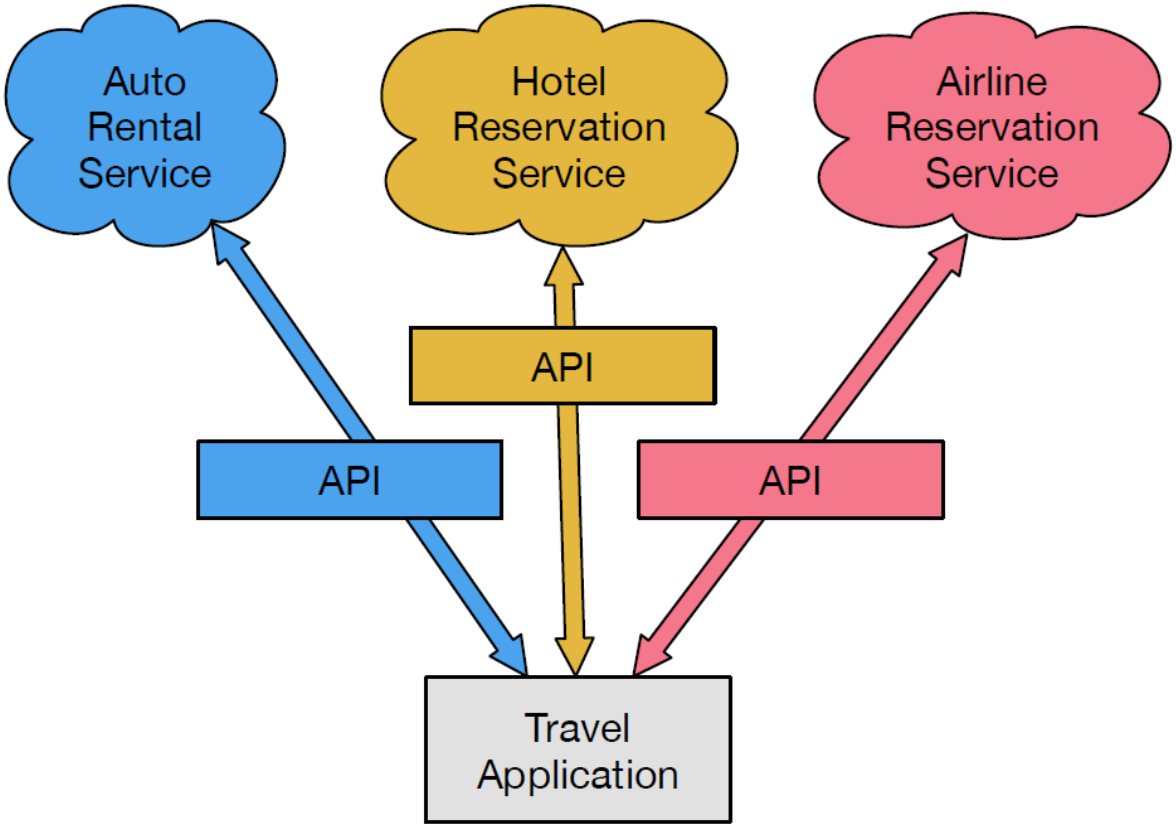
```
import urllib.request

img = urllib.request.urlopen('http://data.pr4e.org/cover3.jpg').read()
fhand = open('cover3.jpg', 'wb')
fhand.write(img)
fhand.close()
```



Rest API

https://www.youtube.com/watch?v=pHFWGN-upGM



API

When an application makes a set of services in its API available over the web, we call these **web services**.

Advantages of service architectures:

- We always maintain only **one copy** of the data.
- The **owners** of the data can set rules about the use of their data.

API: Geocoding Case Study

```
>>> import urllib.request, urllib.parse
>>> serviceurl = 'http://py4e-data.dr-chuck.net/json?'
>>> api_key = 42
>>> address = "Ann Arbor, MI"
>>> parms = {'address':address}
>>> parms['key'] = api_key
>>> url = serviceurl + urllib.parse.urlencode(parms)
>>> uh = urllib.request.urlopen(url)
>>> data = uh.read().decode()
>>> print(data) # Returns JSON
{"results" : [{ "address_components" : [{
              "long_name" : "700", ...
```

API: Web Forms

The previous code – with the exception of the api_key – can be used to submit web forms.

```
>>> serviceurl = 'http://duckduckgo.com/html/?'

>>> query = "Python"

>>> parms = {'q':query}

>>> url = serviceurl + urllib.parse.urlencode(parms)

>>> html_page = urllib.request.urlopen(url).read().decode()
```

`http://duckduckgo.com/html/`

Exercise 1

Write code to query http://duckduckgo.com/html/ with the key query "data science". Parse the resulting page by returning all the unique web URLs. Return only the base URLs (http://duckduckgo.com, www.duckduckgo.com, ...)

hint: Use re.findall()


15 minutes

BeautifulSoup: Parsing Wikipedia Tables

```
>>> from bs4 import BeautifulSoup

>>> import urllib

>>> url = 'https://en.wikipedia.org/wiki/List_of_Asian_countries_by_area'

>>> html = urllib.request.urlopen(url).read()

>>> soup = BeautifulSoup(html,'lxml')

>>> print(soup.prettify())
```

Parsing Wikipedia Tables

cat example.py

```
from bs4 import BeautifulSoup
import urllib
url = 'https://en.wikipedia.org/wiki/List_of_Asian_countries_by_area'
html = urllib.request.urlopen(url).read()
soup = BeautifulSoup(html,'lxml')
myTable = soup.find('table',{'class':'wikitable sortable'})
tbody = myTable.find('tbody')
rows = tbody.find_all('tr')
for row in rows:
        cols = row.find_all('td')
        cols = [ele.text.strip() for ele in cols]
        print(cols)
```

Parsing Wikipedia Tables

python example.py

```
['18', 'Japan', '377,930', '']
['19', 'Vietnam', '331,212', '']
['20', 'Malaysia', '330,803', '']
['21', 'Oman', '309,500', '']
...
```

Exercise 2

Find another table on Wikipedia and repeat the process in the previous slide to parse that table.