Python Lesson 7: Data from the Internet

vanderbi.lt/py

Steve Baskauf

DiSC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Evaluation

- Please take a moment to go to the evaluation link at the top of the lesson page and fill out the form.

Intermediate Python and R

- Five sessions starting after spring break.
- More information at vanderbi.lt/py

- Beginner R lessons after spring break
- More information at vanderbi.lt/r

Main goals for today

- Conceptual understanding of HTTP and APIs
- Acquire **template code** for input of:
  - online CSV file as list of lists
  - online CSV file as list of dictionaries
  - JSON file from API as complex data structure
- Use the `requests` module to perform HTTP operations
- Apply the `csv` module `.reader()` and `.DictReader()` methods to processed body text from a remote server
- Apply the `requests` module `.json()` method to body text from a remote server

HTTP protocol

Hypertext Transfer Protocol (HTTP), used to carry out an interaction across the Internet. mediated by. Retrieving information using HTTP GET is called "dereferencing a URI". (People also say "resolving" a URI.)

Image from Clipart Panda

HTTP **GET**
http://dbpedia.org/resource/Bonobo
Accept: text/html

asking for a web page

Image from Clipart Kid

Client software

(a.k.a. the "machine")

In this case, the client is a web browser. It displays the returned body as a web page.

HTTP **Status:** 200 OK

Body:

```
<?xml version="1.0" encoding="UTF-8" ?><!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd"><html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:dbpprop="http://dbpedia.org/property/"
…
```

Web server

HTTP GET example

Getting generic files

- An HTTP GET request can just retrieve a particular file
- The file is returned in its normal text format
- If the client is a browser, it will try to render the file appropriately
- If the client is a Python script, the programmer decides what happens
- Examples:
  - HTML web page
  - CSV file

Try reading CSV files examples

APIs

- An **automated programming interface (API)** is a special kind of server on the Internet
- An **endpoint** is a URL that is used to carry out particular interactions
- The endpoint URL is usually a base API URL plus a specific resource URL (or subpath). Example:
    - `http://api.gbif.org/v1` base URL
    - `/occurrence/search` resource URL
    - `http://api.gbif.org/v1/occurrence/search` is full endpoint URL
- Any good API will have a developer guide to tell you how to use it

Sending data using GET

- URLs can specify parameters using a **query string**
- **Parameters** follow the `?` character
- Parameter **key:value pairs** typically are separated by an equals sign (`=`)
- Ampersands (`&`) are used to separate **multiple parameters**
- **URL encoding** is required for characters in the query that aren't allowed in URLs
- Example:

```
http://example.org/api?id=294&language=en
```

More on APIs

- Larger amounts of data (like whole files) can be sent using HTTP **POST** instead of GET
- Using an API may require **authentication**
  - sometimes for reading
  - almost always for writing data
- Sometimes you can specify the form of the response, but now nearly always **JSON**
- Retrieving large amounts often requires using **paging**. **Important**: see API etiquette notes!
- See notes from the **APIs/web scraping** Python class

Examples

- Data from Global Biodiversity Information Facility (GBIF) API
- Email me know if you want to try the Twitter API challenge problem (covered in API/web scraping lessons archive)
- Challenge problem 2.C. answer (super cartoon checker)
- Homework 2 answer (International Space Station)