Web Scraping with Python

March 31, 2017

Agenda

- Introduction
- Data sources
- Types of available data
- RESTful APIs
- Scraping
- Crawling
- Useful tools
- Integration

Collecting Data from the Web

Your Mission

- Get the data you need to the job you need to do.
- Data is designed for operations, not analysis.
- Data used in analysis usually needs to be denormalized.
- There can be many gatekeepers.

So what makes a good data source?

Publicly Available Datasets

- Amazon S3 Cloud Public Datasets
- UCI Machine Learning Repository
- Awesome Public Datasets
- More Datasets
- Kaggle
- Data.gov
- Sunlight Foundation

Strategy: look for academic data sets that implement techniques that you’re interested in - these may lead you to initial data or other primary sources.

In the end,
the best data
is *always* the data
you gather yourself.

Features of Data in the Wild

Common Data Formats

- CSV - stores tabular data in plain text where each row is a record and the values are delimited by commas.
- JSON - a data-interchange format that is easy for humans to read and write and for machines to parse and generate.
- XML - a markup language designed to carry data, with a focus on *what the data is*.
- HTML - a markup language designed to display data, with a focus on *how the data looks*.

Serialization

- Converting structured data into format to be shared, stored, or updated
- Original structure can be restored.
- Minimizes the size of the data so that it takes up less disk space when stored or bandwidth when shared.
- `.write()`

Parsing

- Processing input into meaningful structures to extract information.
- Examples:
  - A student parses a sentence into subject, verb, and object.
  - A compiler parses source code.
  - A CSV parser reads a stream according to rules (comma delimiters, quoting, etc) to extract the data in each line of a file.
- `.read()`

RESTful APIs

Application Programming Interface

Although computer scientists are used to APIs; most of the time APIs refer to *Web APIs* now - and this is essentially a data ingestion topic.

“In the simplest terms, APIs are sets of requirements that govern how one application can talk to another. APIs aren't at all new; whenever you use a desktop or laptop, APIs are what make it possible to move information between programs”

http://readwrite.com/2013/09/19/api-defined

Application Programming Interface

“These days, APIs are especially important because they dictate how developers can create new apps that tap into big Web services—social networks like Facebook or Pinterest, for instance, or utilities like Google Maps or Dropbox. The developer of a game app, for instance, can use the Dropbox API to let users store their saved games in the Dropbox cloud instead of working out some other cloud-storage option from scratch.

http://readwrite.com/2013/09/19/api-defined

Application Programming Interface

“Viewed more broadly, though, APIs make possible a sprawling array of Web-service "mashups," in which developers use mix and match APIs from the likes of Google or Facebook or Twitter to create entirely new apps and services. In many ways, the widespread availability of APIs for major services is what's made the modern Web experience possible.”

http://readwrite.com/2013/09/19/api-defined

goodreads
twitter
facebook
WEATHER UNDERGROUND
yelp
wordnik
SOUNDCLOUD
amazon

APIs

REST is a simple way to organize interactions between independent systems.

REST allows you to interact with minimal overhead with clients as diverse as mobile phones and other websites. In theory, REST is not tied to the web, but it's almost always implemented as such, and was inspired by HTTP. As a result, REST can be used wherever HTTP can.

So what is HTTP?

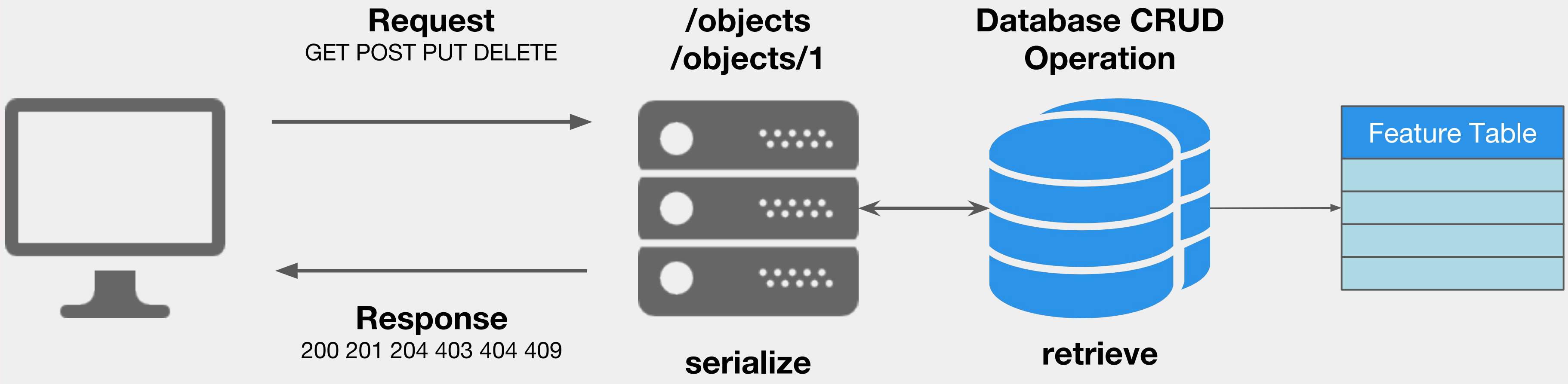
HTTP Basics

- HyperText Transfer Protocol
- Foundation of data communication on the web
- Send request, receive response
- HTTP Request Methods
    - GET
    - HEAD
    - POST
    - PUT
    - DELETE

HTTP Basics

- User Agent String - browser, OS, and other system info.
- HTTP Status Codes
  - 1xx - Informational
  - 2xx - Success
  - 3xx - Redirection
  - 4xx - Client Error
  - 5xx - Server Error
- TLS - successor to SSL that provides protocol for secure communications.

REST API Feature Interaction



Request
GET POST PUT DELETE

/objects
/objects/1

Database CRUD
Operation

Feature Table

Response
200 201 204 403 404 409

serialize

retrieve

Scraping

What is Web Scraping?

- Automated extraction of specific information from a web page.
- Often a page's text content, but it may also include:
  - Headers
  - Date the page was published
  - Links are present on the page
  - Any other specific information the page contains
- Objective: extracting specific information from a page

What Makes Scraping Difficult?

- Need to determine what information you want
- Need custom scraper for each site
- Different pages have different structure
- Page structure/content changes periodically
- Javascript can make scraping difficult
- Potential legal issues

HTML Structure

```
<html>

    <head> </head>

        <body>

            Content (<p>, <div>, <table> ,<ul>, <ol>, etc.)

        </body>

</html>
```

- HTML Attributes - title, href, size, alt, etc.

Crawling

What is Web Crawling?

- Traversal of a website's link network
- Saving or indexing all the pages in that network
- Obtain information about link networks within and between websites.

What Makes Crawling Difficult?

- Need to know the site structure in advance
- Determining depth of crawl
- Latency/bandwidth variations
- Site mirrors and duplicate pages
- Spider/crawler traps

From Scraping to Crawling

- Different Objectives
  - Scraping - extracting specific information from a page.
  - Crawling - obtain information about link networks within and between websites.

- Possible to crawl a site and scrape pages.
- Need to know specific content we want from each page .
- Need to have information about site structure in advance.

Tools

Requests.py

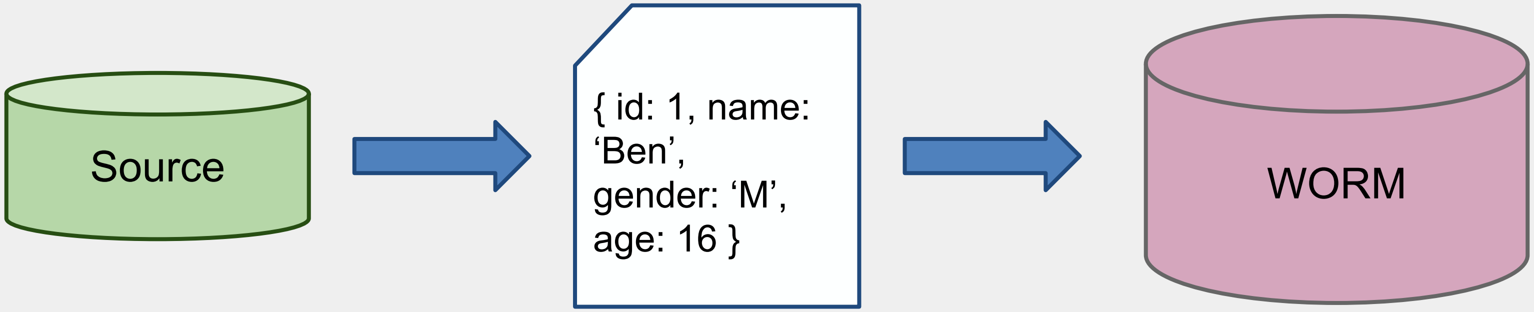
- Elegant, simple HTTP library for Python
- How it works:
    - Make a request to a web page (get, post, put, etc.)
    - Receive a response from server
    - Read content of server response
        - Headers
        - Cookies
        - Content
        - Etc.

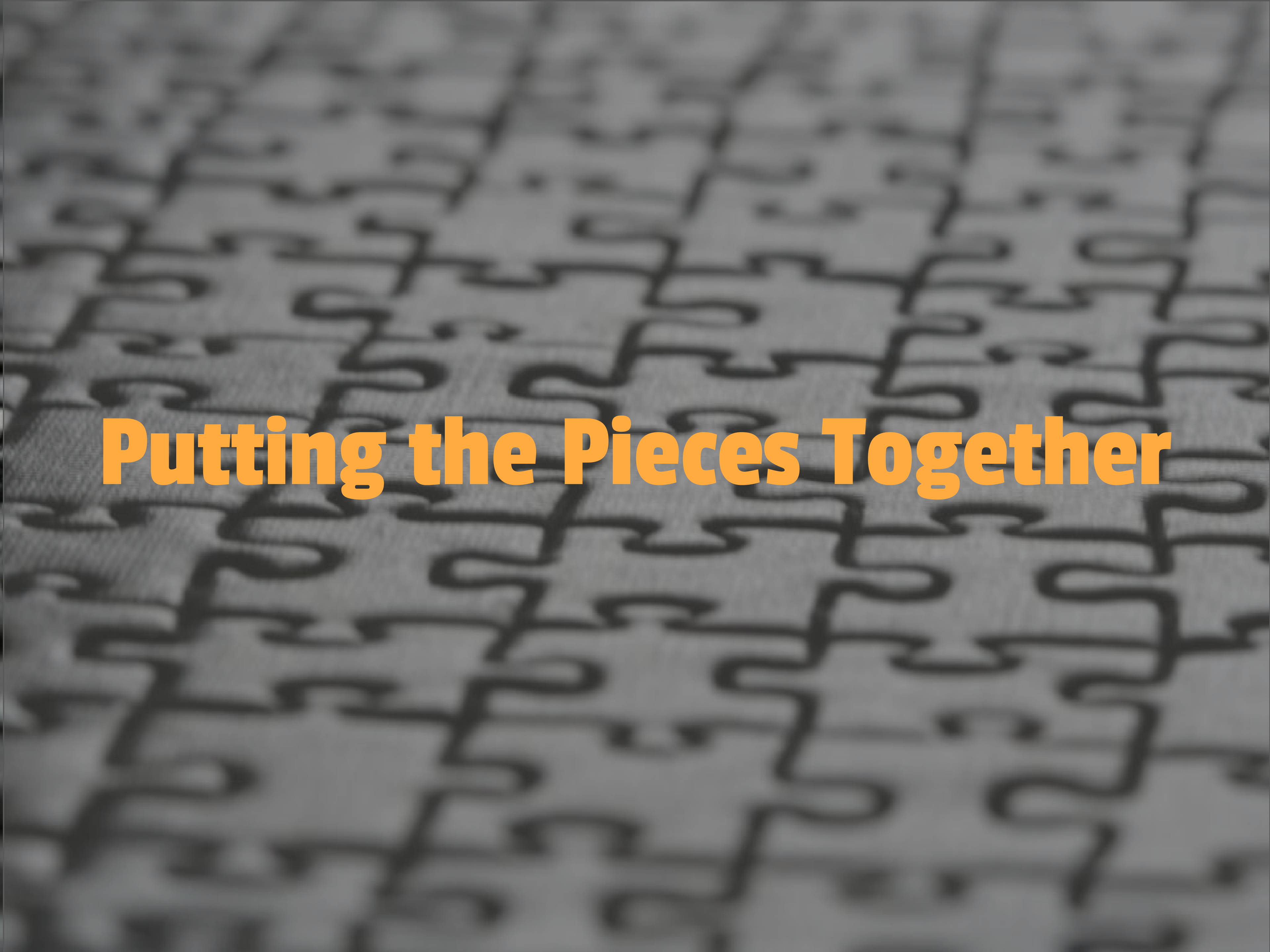
Scrapy

- Open source framework for crawling websites and extracting structured data.
- Spiders - define how a certain site (or group of sites) will be scraped.
- Selectors - select certain parts of the HTML document.
- Items - objects that serve as simple containers used to collect the scraped data.
- Scrapy Shell - debug scraping code quickly without having to run spider.
- Pipelines, extractors, and more!

Databases and Database Tools

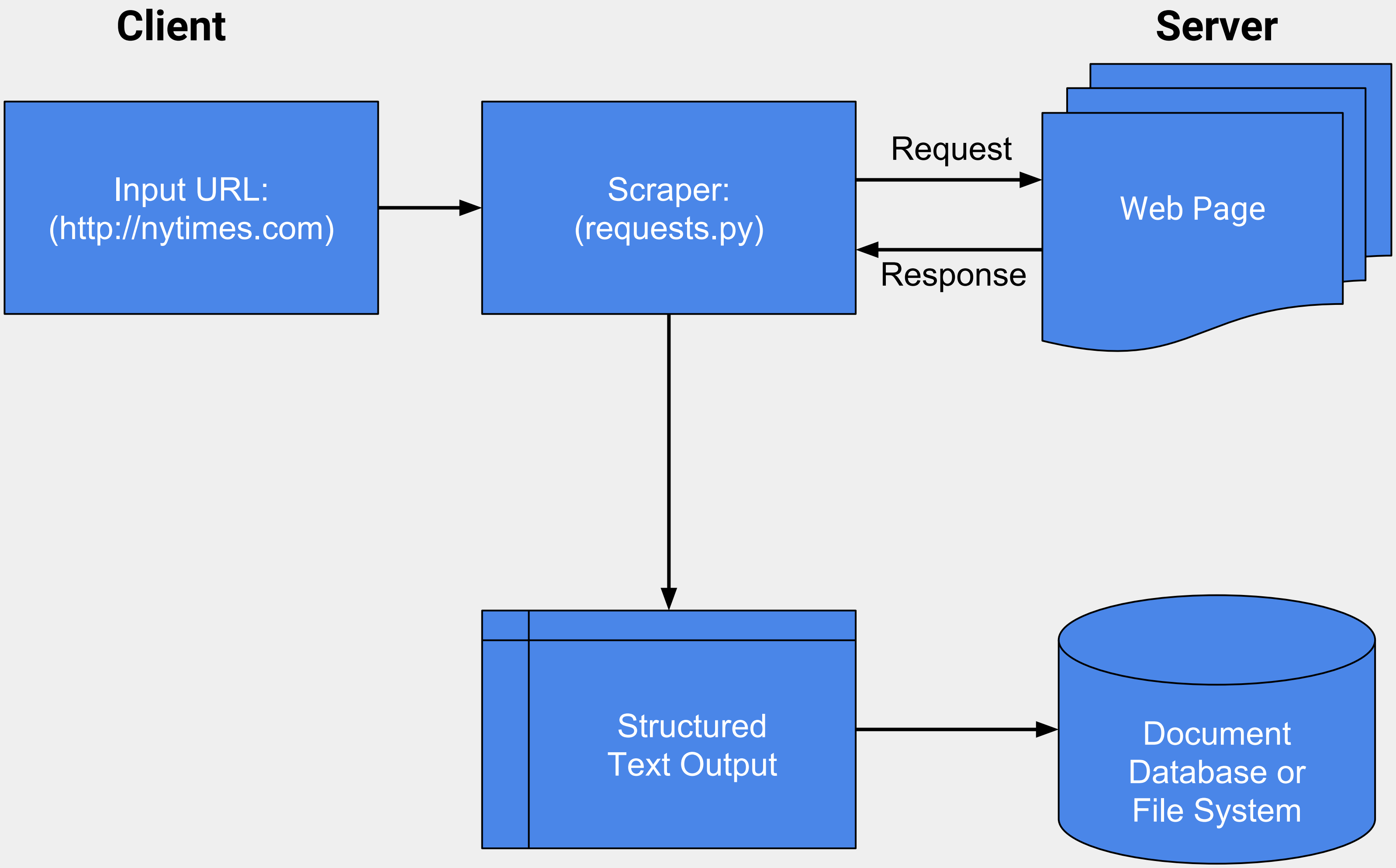
- PostgreSQL
- Postgres App
- Pgadmin
- SQLite - lightweight, self-contained SQL database engine.
- Psycopg2
- Postico
- Postman
- JetBrains Database Navigator





Putting the Pieces Together

Basic Workflow



Client

Server

Input URL:
(http://nytimes.com)

Scraper:
(requests.py)

Request

Response

Web Page

Structured
Text Output

Document
Database or
File System

Being a Good Citizen

- Robot.txt files - tell you what the site does and does not allow from crawlers.
- Rate limiting - limiting the frequency at which you ping a website.
  - Too much traffic too quickly may bring down a smaller website.
  - Larger websites may block your IP address.