Part I: Web Services Using Python

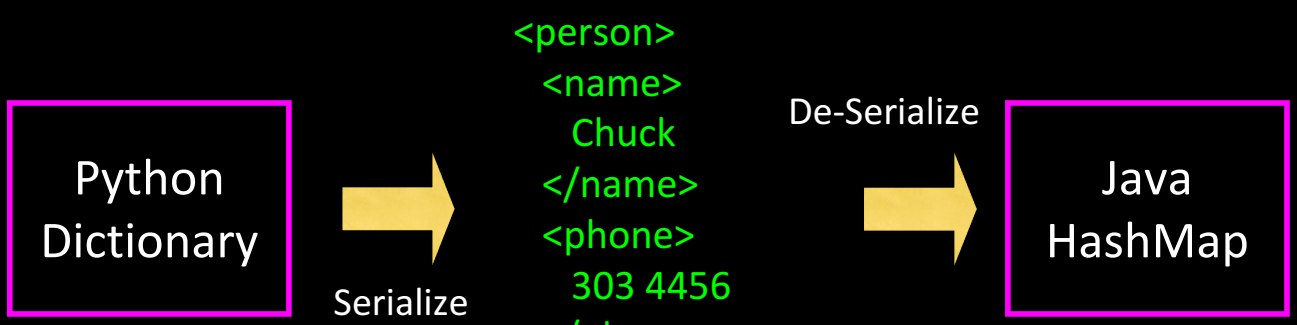
Recall: Data on the Web

- There are two commonly used formats: XML and JSON

Sending Data across the "Net"

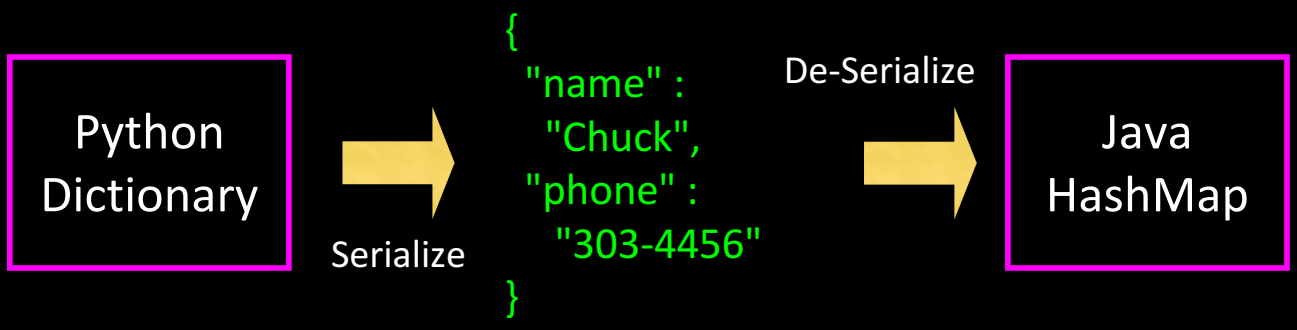
a.k.a. "Wire Protocol" - What we send on the "wire"

Agreeing on a "Wire Format"



XML

Agreeing on a "Wire Format"



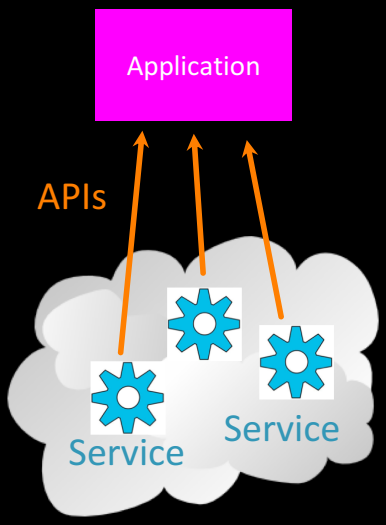
JSON

Service Oriented Approach

http://en.wikipedia.org/wiki/Service-oriented_architecture

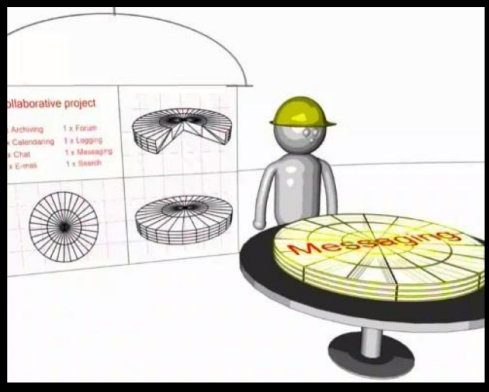
Service Oriented Approach

- Most non-trivial web applications use services
- They use services from other applications
  - Credit Card Charge
  - Hotel Reservation systems
- Services publish the “rules” applications must follow to make use of the service (API)



Multiple Systems

- Initially - two systems cooperate and split the problem
- As the data/service becomes useful - multiple applications want to use the information / application



http://www.youtube.com/watch?v=mj-kCFzF0ME

5:15

Web Services

http://en.wikipedia.org/wiki/Web_services

Application Program Interface

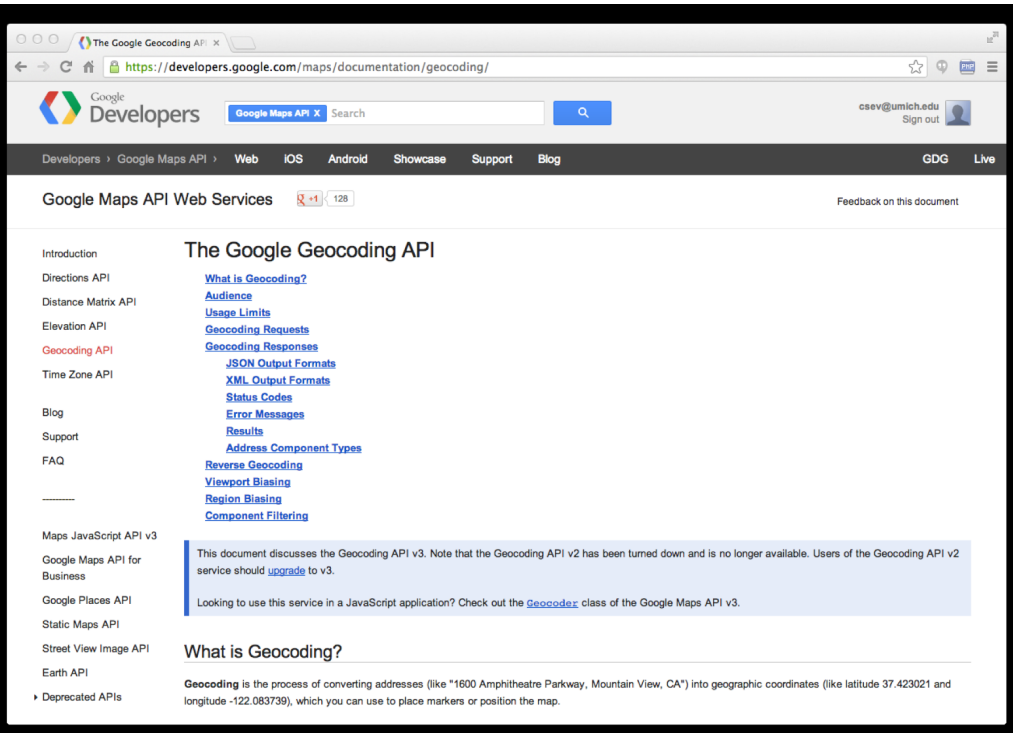
The API itself is largely abstract in that it specifies an interface and controls the behavior of the objects specified in that interface. The software that provides the functionality described by an API is said to be an "implementation" of the API. An API is typically defined in terms of the programming language used to build an application.

http://en.wikipedia.org/wiki/API

Web Service Technologies

- SOAP - Simple Object Access Protocol (software)
  - Remote programs/code which we use over the network
  - Note: Dr. Chuck does not like SOAP because it is overly complex
- REST - Representational State Transfer (resource focused)
  - Remote resources which we create, read, update and delete remotely

http://en.wikipedia.org/wiki/SOAP_(protocol)
http://en.wikipedia.org/wiki/REST



https://developers.google.com/maps/documentation/geocoding/

```
{
    "status": "OK",
     "results": [
        {
            "geometry": {
                "location_type": "APPROXIMATE",
                 "location": {
                    "lat": 42.2808256,
                     "lng": -83.7430378
                }
            },
            "address_components": [
                {
                    "long_name": "Ann Arbor",
                     "types": [
                        "locality",
                         "political"
                    ],
                    "short_name": "Ann Arbor"
                }
             ],
             "formatted_address": "Ann Arbor, MI, USA",
             "types": [
                "locality",
                "political"
            ]
        }
    ]
}
```

http://maps.googleapis.com/maps/api/geocode/json?sensor=false&address=Ann+Arbor%2C+MI

geojson.py

```
import urllib
import json

serviceurl = 'http://maps.googleapis.com/maps/api/geocode/json?'

while True:
    address = raw_input('Enter location: ')
    if len(address) < 1 : break

    url = serviceurl + urllib.urlencode({'sensor':'false',
          'address': address})
    print 'Retrieving', url
    uh = urllib.urlopen(url)
    data = uh.read()
    print 'Retrieved',len(data),'characters'

    try: js = json.loads(str(data))
    except: js = None
    if 'status' not in js or js['status'] != 'OK':
        print '==== Failure To Retrieve ===='
        print data
        continue

    print json.dumps(js, indent=4)

    lat = js["results"][0]["geometry"]["location"]["lat"]
    lng = js["results"][0]["geometry"]["location"]["lng"]
    print 'lat',lat,'lng',lng
    location = js['results'][0]['formatted_address']
    print location
```

```
Enter location: Ann Arbor, MI
Retrieving http://maps.googleapis.com/...
Retrieved 1669 characters
lat 42.2808256 lng -83.7430378
Ann Arbor, MI, USA
Enter location:
```

geojson.py

API Security and Rate Limiting

- The compute resources to run these APIs are not “free”
- The data provided by these APIs is usually valuable
- The data providers might limit the number of requests per day, demand an API “key”, or even charge for usage
- They might change the rules as things progress...

Usage Limits

The Google Geocoding API has the following limits in place:

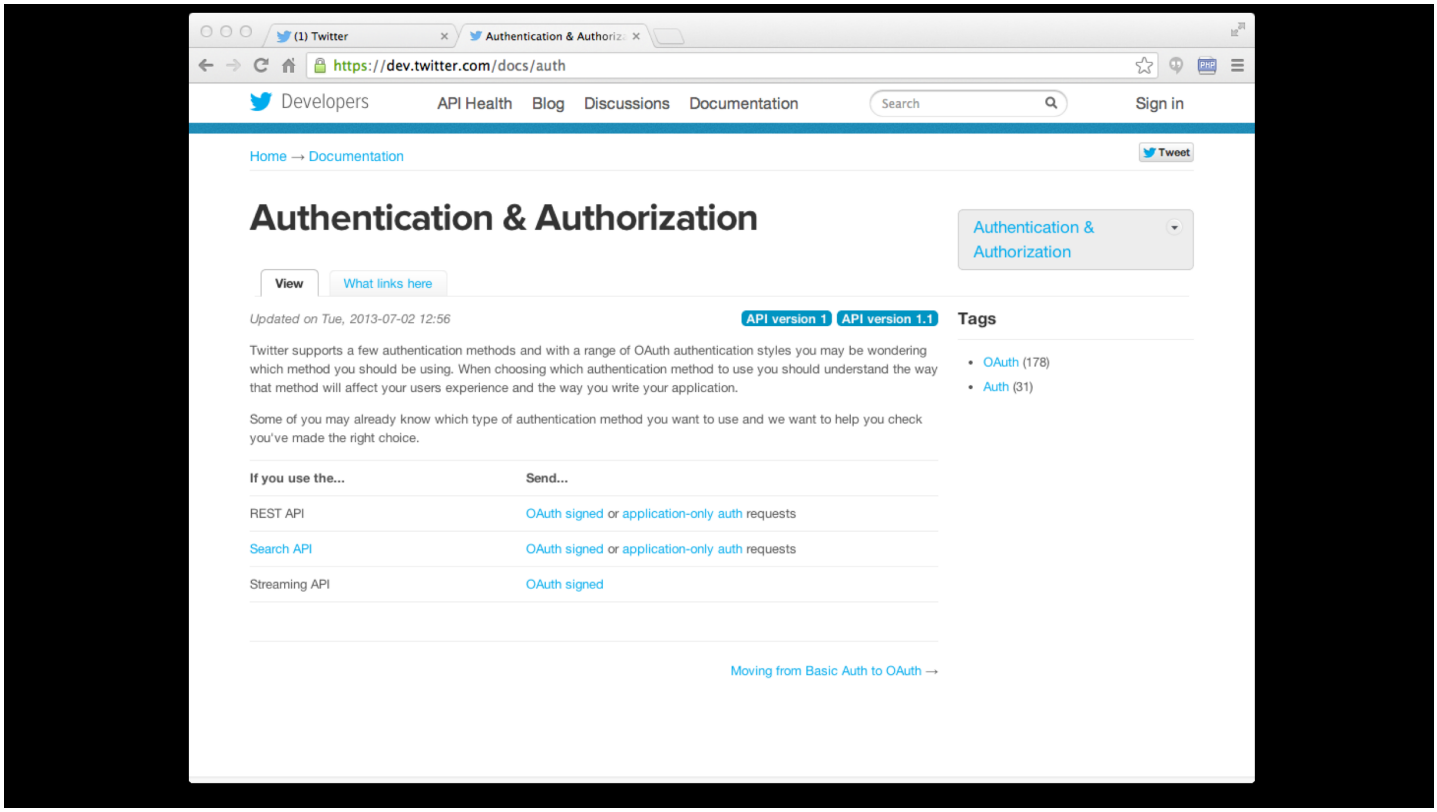
- 2,500 requests per day.

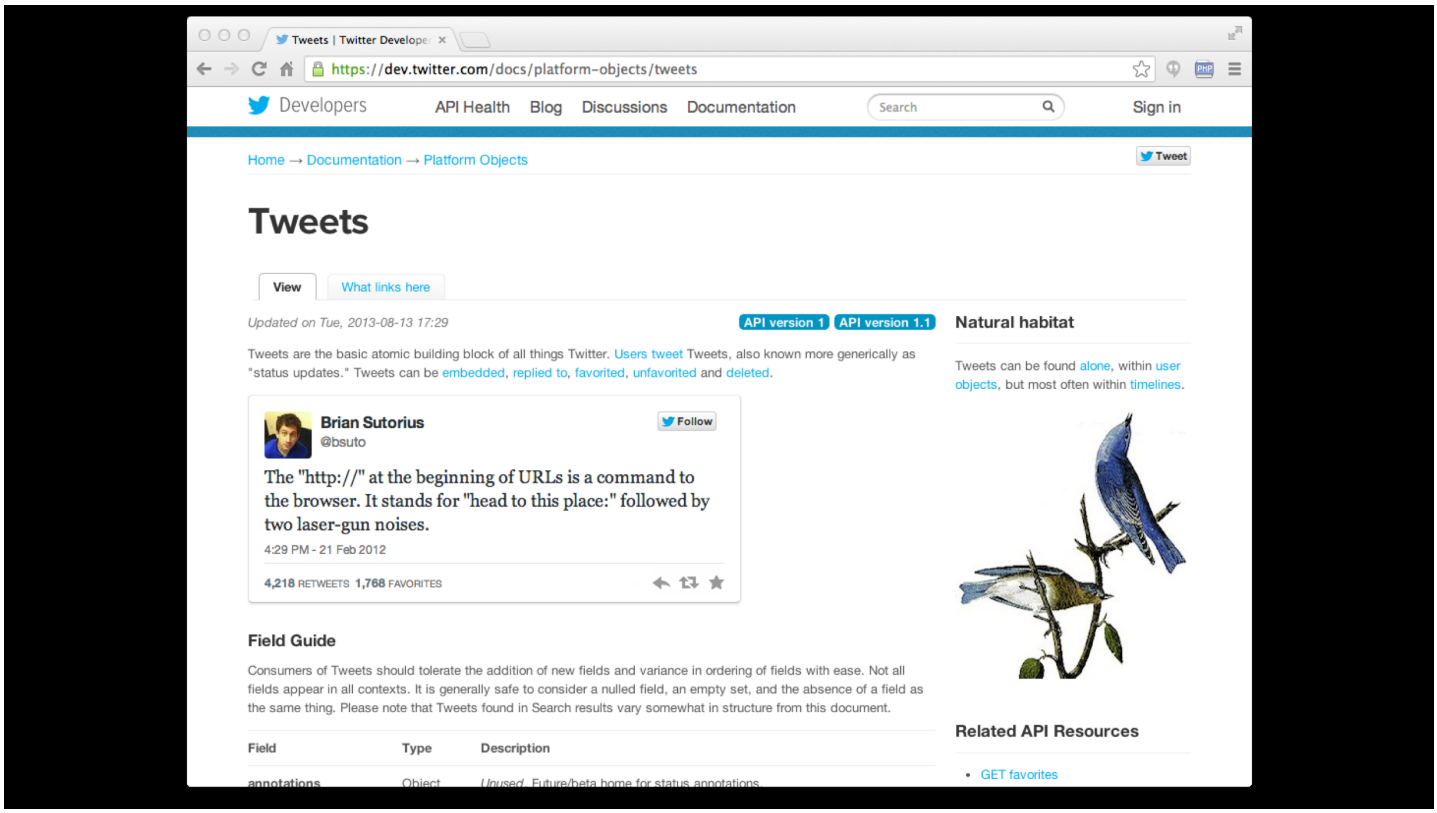
Google Maps API for Business customers have higher limits:

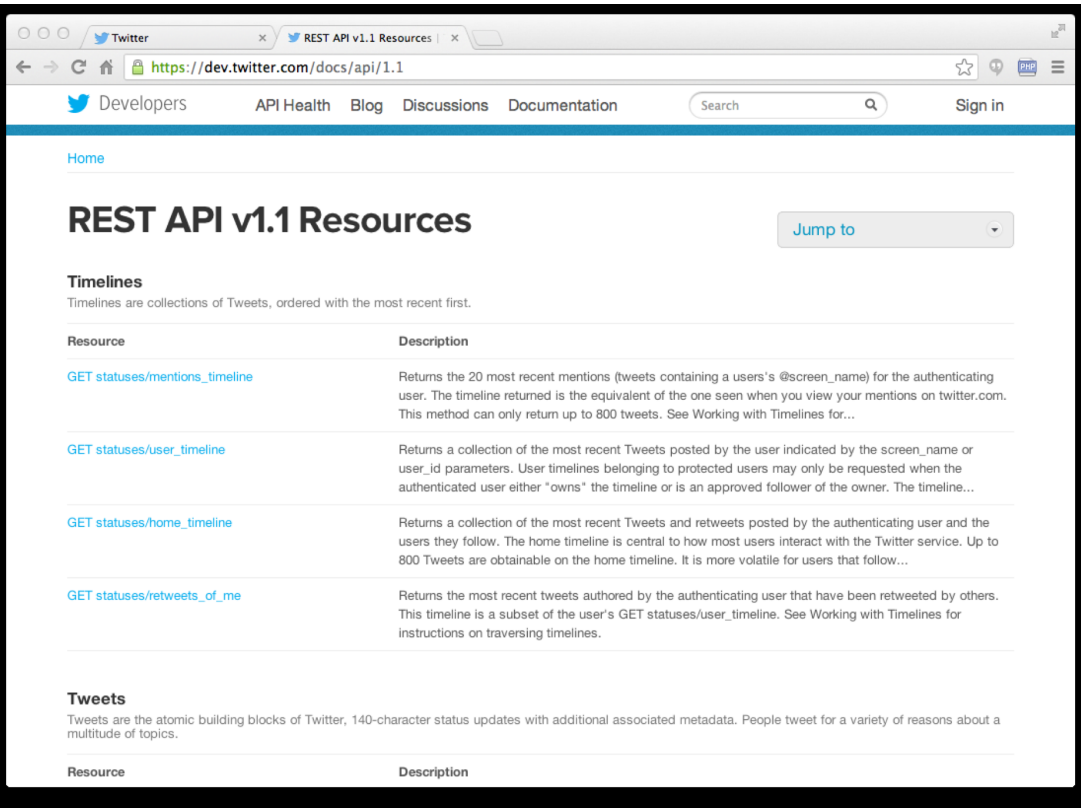
- 100,000 requests per day.

These limits are enforced to prevent abuse and/or repurposing of the Geocoding API, and may be changed in the future without notice. Additionally, we enforce a request rate limit to prevent abuse of the service. If you exceed the 24-hour limit or otherwise abuse the service, the Geocoding API may stop working for you temporarily. If you continue to exceed this limit, your access to the Geocoding API may be blocked.

The Geocoding API may only be used in conjunction with a Google map; geocoding results without displaying them on a map is prohibited. For complete details on allowed usage, consult the Maps API Terms of Service License Restrictions.


(1) Twitter
Authentication & Authoriz
https://dev.twitter.com/docs/auth
Developers
API Health
Blog
Discussions
Documentation
Search
Sign in
Home → Documentation
Tweet
Authentication & Authorization
Authentication & Authorization
View
What links here
Updated on Tue, 2013-07-02 12:56
API version 1
API version 1.1
Twitter supports a few authentication methods and with a range of OAuth authentication styles you may be wondering which method you should be using. When choosing which authentication method to use you should understand the way that method will affect your users experience and the way you write your application.
Some of you may already know which type of authentication method you want to use and we want to help you check you've made the right choice.
If you use the...
Send...
REST API
OAuth signed or application-only auth requests
Search API
OAuth signed or application-only auth requests
Streaming API
OAuth signed
Moving from Basic Auth to OAuth →
Tags
OAuth (178)
Auth (31)


Tweets | Twitter Develope
https://dev.twitter.com/docs/platform-objects/tweets
Developers
API Health
Blog
Discussions
Documentation
Search
Sign in
Home → Documentation → Platform Objects
Tweet
Tweets
View
What links here
Updated on Tue, 2013-08-13 17:29
API version 1
API version 1.1
Tweets are the basic atomic building block of all things Twitter. Users tweet Tweets, also known more generically as "status updates." Tweets can be embedded, replied to, favorited, unfavorited and deleted.
Brian Sutorius
@bsuto
Follow
The "http://" at the beginning of URLs is a command to the browser. It stands for "head to this place:" followed by two laser-gun noises.
4:29 PM - 21 Feb 2012
4,218 RETWEETS 1,768 FAVORITES
Field Guide
Consumers of Tweets should tolerate the addition of new fields and variance in ordering of fields with ease. Not all fields appear in all contexts. It is generally safe to consider a nulled field, an empty set, and the absence of a field as the same thing. Please note that Tweets found in Search results vary somewhat in structure from this document.
Field
Type
Description
annotations
Object
Unused. Future/beta home for status annotations.
Natural habitat
Tweets can be found alone, within user objects, but most often within timelines.
Related API Resources
GET favorites

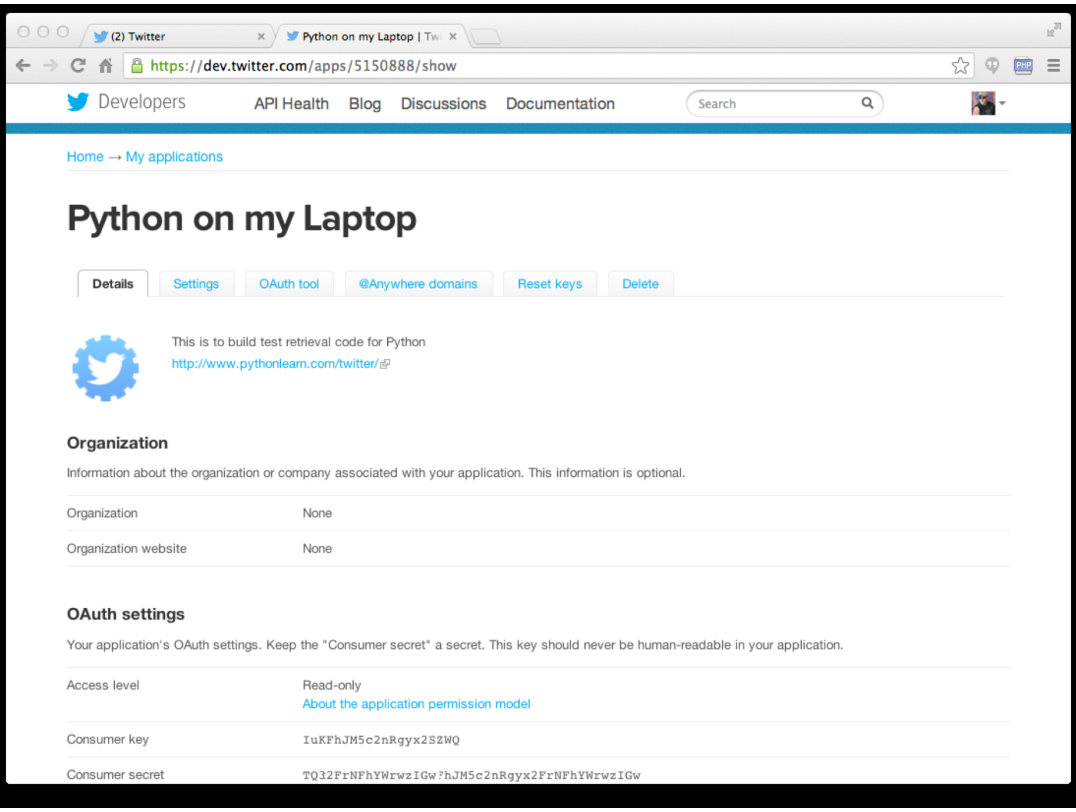


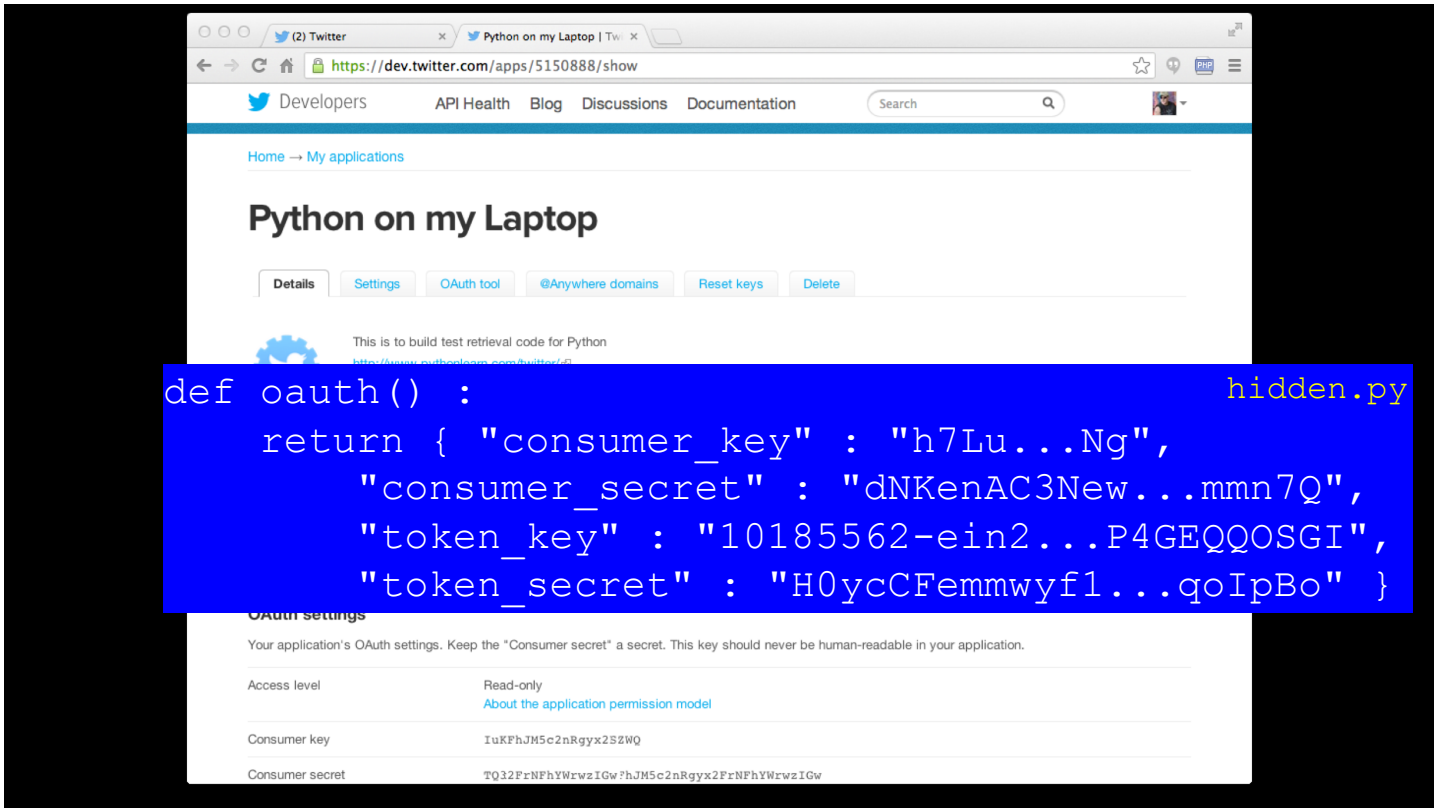
twitter2.py

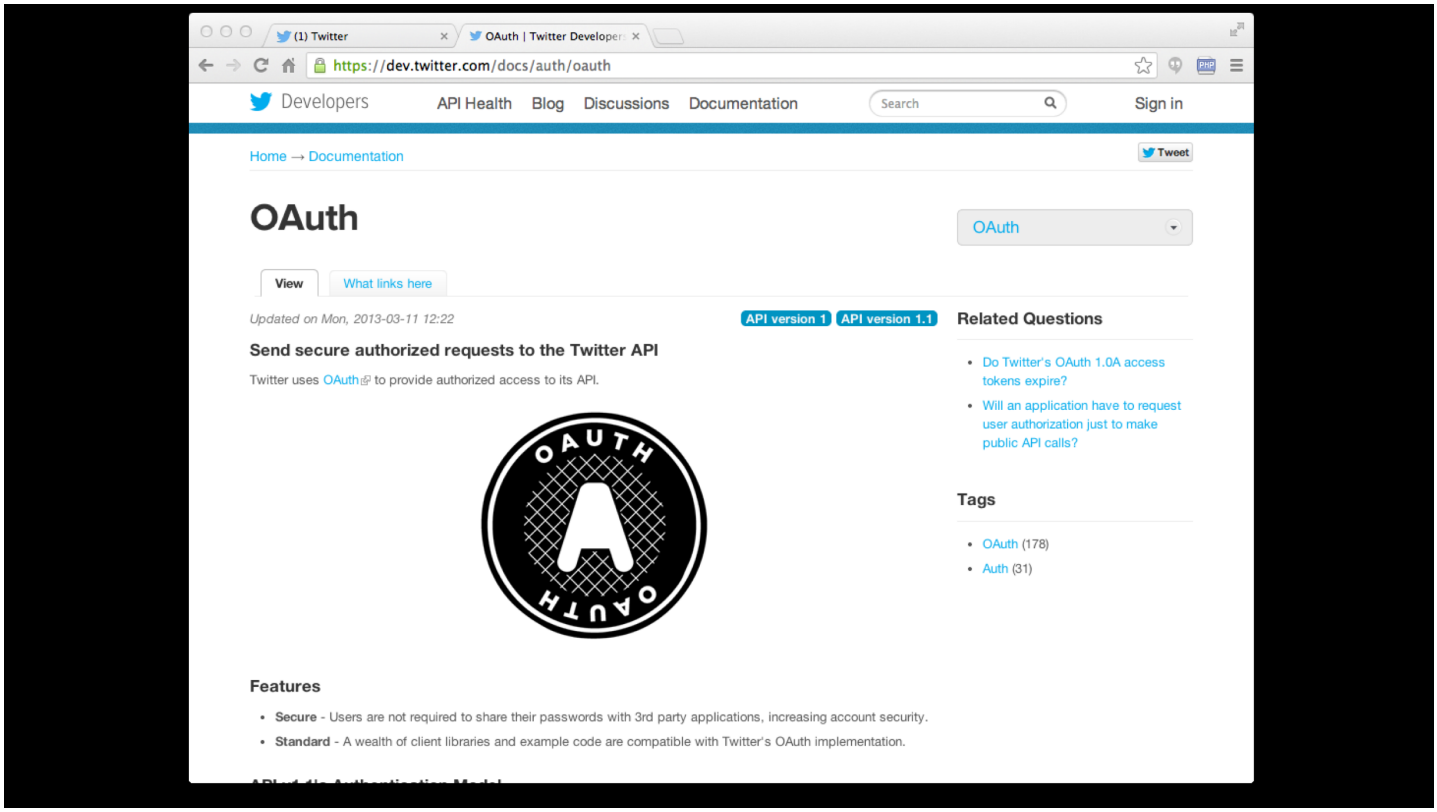
```
import urllib
import twurl
import json
TWITTER_URL = 'https://api.twitter.com/1.1/friends/list.json'
while True:
    print ''
    acct = raw_input('Enter Twitter Account:')
    if ( len(acct) < 1 ) : break
    url = twurl.augment(TWITTER_URL,
        {'screen_name': acct, 'count': '5'} )
    print 'Retrieving', url
    connection = urllib.urlopen(url)
    data = connection.read()
    headers = connection.info().dict
    print 'Remaining', headers['x-rate-limit-remaining']
    js = json.loads(data)
    print json.dumps(js, indent=4)
    for u in js['users'] :
        print u['screen_name']
        s = u['status']['text']
        print '  ',s[:50]
```

twitter2.py

```
Enter Twitter Account:drchuck
Retrieving https://api.twitter.com/1.1/friends ...
Remaining 14
{
    "users": [
        {
            "status": {
                "text": "@jazzychad I just bought one .__.",
                 "created_at": "Fri Sep 20 08:36:34 +0000 2013",
             },
             "location": "San Francisco, California",
             "screen_name": "leahculver",
             "name": "Leah Culver",
         },
         {
            "status": {
                "text": "RT @WSJ: Big employers like Google ...",
                 "created_at": "Sat Sep 28 19:36:37 +0000 2013",
             },
             "location": "Victoria Canada",
             "screen_name": "_valeriei",
             "name": "Valerie Irvine",
     ],
}
Leahculver
   @jazzychad I just bought one .__._
Valeriei
   RT @WSJ: Big employers like Google, AT&amp;amp;T are h
Ericbollens
   RT @lukew: sneak peek: my LONG take on the good &a
halherzog
  Learning Objects is 10. We had a cake with the LO,
```




(2) Twitter
Python on my Laptop
https://dev.twitter.com/apps/5150888/show
Developers
API Health
Blog
Discussions
Documentation
Search
Home → My applications
Python on my Laptop
Details
Settings
OAuth tool
@Anywhere domains
Reset keys
Delete
This is to build test retrieval code for Python
def oauth() :
return { "consumer_key" : "h7Lu...Ng",
"consumer_secret" : "dNKenAC3New...mmn7Q",
"token_key" : "10185562-ein2...P4GEQQOSGI",
"token_secret" : "H0ycCFemmwyf1...qoIpBo" }
hidden.py
Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.
Access level
Read-only
About the application permission model
Consumer key
IuKFhJM5c2nRgyx2SZWQ
Consumer secret
TQ32FrNFhYWrwzIGw?hJM5c2nRgyx2FrNFhYWrwzIGw


(1) Twitter
OAuth | Twitter Developers
https://dev.twitter.com/docs/auth/oauth
Developers
API Health
Blog
Discussions
Documentation
Search
Sign in
Home → Documentation
Tweet
OAuth
OAuth
View
What links here
Updated on Mon, 2013-03-11 12:22
API version 1
API version 1.1
Send secure authorized requests to the Twitter API
Twitter uses OAuth to provide authorized access to its API.
OAUTH
OAUTH
Related Questions
Do Twitter's OAuth 1.0A access tokens expire?
Will an application have to request user authorization just to make public API calls?
Tags
OAuth (178)
Auth (31)
Features
Secure - Users are not required to share their passwords with 3rd party applications, increasing account security.
Standard - A wealth of client libraries and example code are compatible with Twitter's OAuth implementation.

twurl.py

```
import urllib
import oauth
import hidden

def augment(url, parameters) :
    secrets = hidden.oauth()
    consumer = oauth.OAuthConsumer(secrets['consumer_key'], secrets['consumer_secret'])
    token = oauth.OAuthToken(secrets['token_key'],secrets['token_secret'])
    oauth_request = oauth.OAuthRequest.from_consumer_and_token(consumer,
         token=token, http_method='GET', http_url=url, parameters=parameters)
    oauth_request.sign_request(oauth.OAuthSignatureMethod_HMAC_SHA1(), consumer, token)
    return oauth_request.to_url()
```

https://api.twitter.com/1.1/statuses/user_timeline.json?count=2&oauth_version=1.0&oauth_token=101...SGI&screen_name=drchuck&oauth_nonce=09239679&oauth_timestamp=1380395644&oauth_signature=rLK...BoD&oauth_consumer_key=h7Lu...GNg&oauth_signature_method=HMAC-SHA1

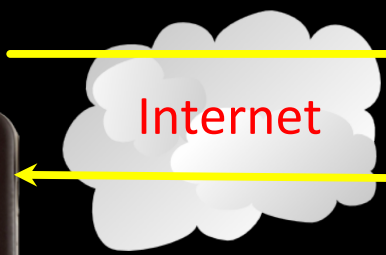
Summary

- Service Oriented Architecture - allows an application to be broken into parts and distributed across a network
- An Application Program Interface (API) is a contract for interaction
- Web Services provide infrastructure for applications cooperating (an API) over a network - SOAP and REST are two styles of web services
- XML and JSON are serialization formats

Part II: Networked Programs

Client

Server



Wikipedia

Internet

HTTP Request

Response GET

socket POST

HTML JavaScript

AJAX CSS

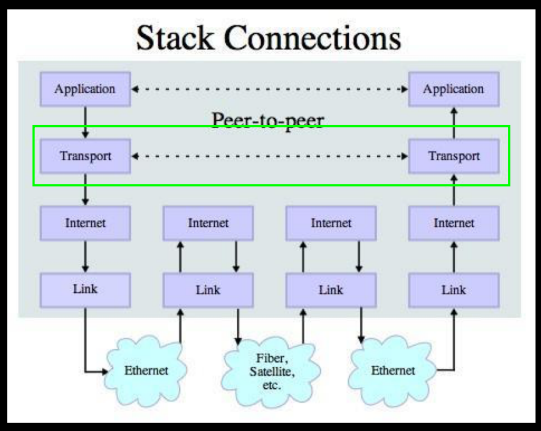
Python Data Store

Templates memcache

Network Architecture....

Transport Control Protocol (TCP)

- Built on top of IP (Internet Protocol)
- Assumes IP might lose some data - stores and retransmits data if it seems to be lost
- Handles "flow control" using a transmit window
- Provides a nice reliable pipe



Source: http://en.wikipedia.org/wiki/Internet_Protocol_Suite

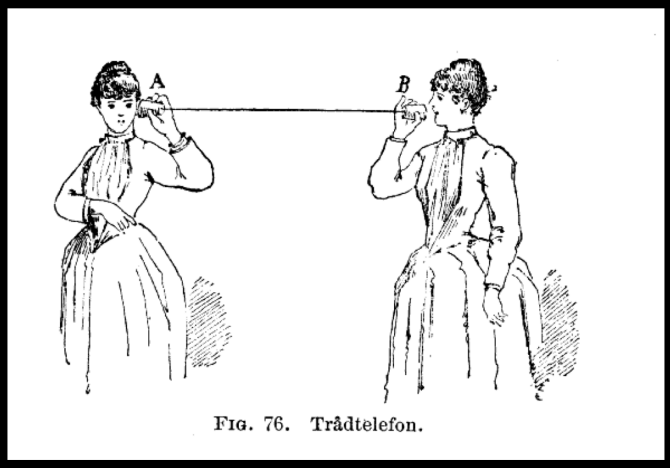


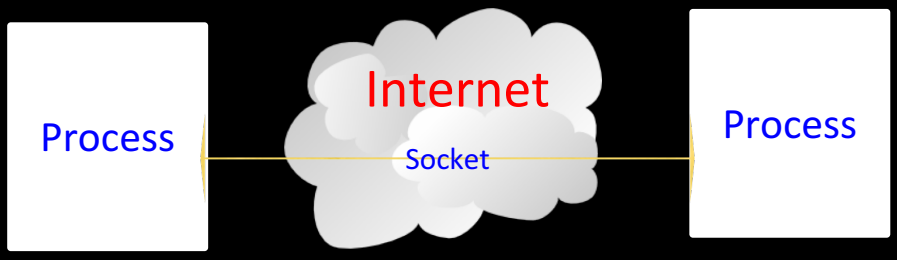
FIG. 76. Trådtelefon.

http://en.wikipedia.org/wiki/Tin_can_telephone

http://www.flickr.com/photos/kitcowan/2103850699/

TCP Connections / Sockets

"In computer networking, an Internet socket or network socket is an endpoint of a bidirectional inter-process communication flow across an Internet Protocol-based computer network, such as the Internet."

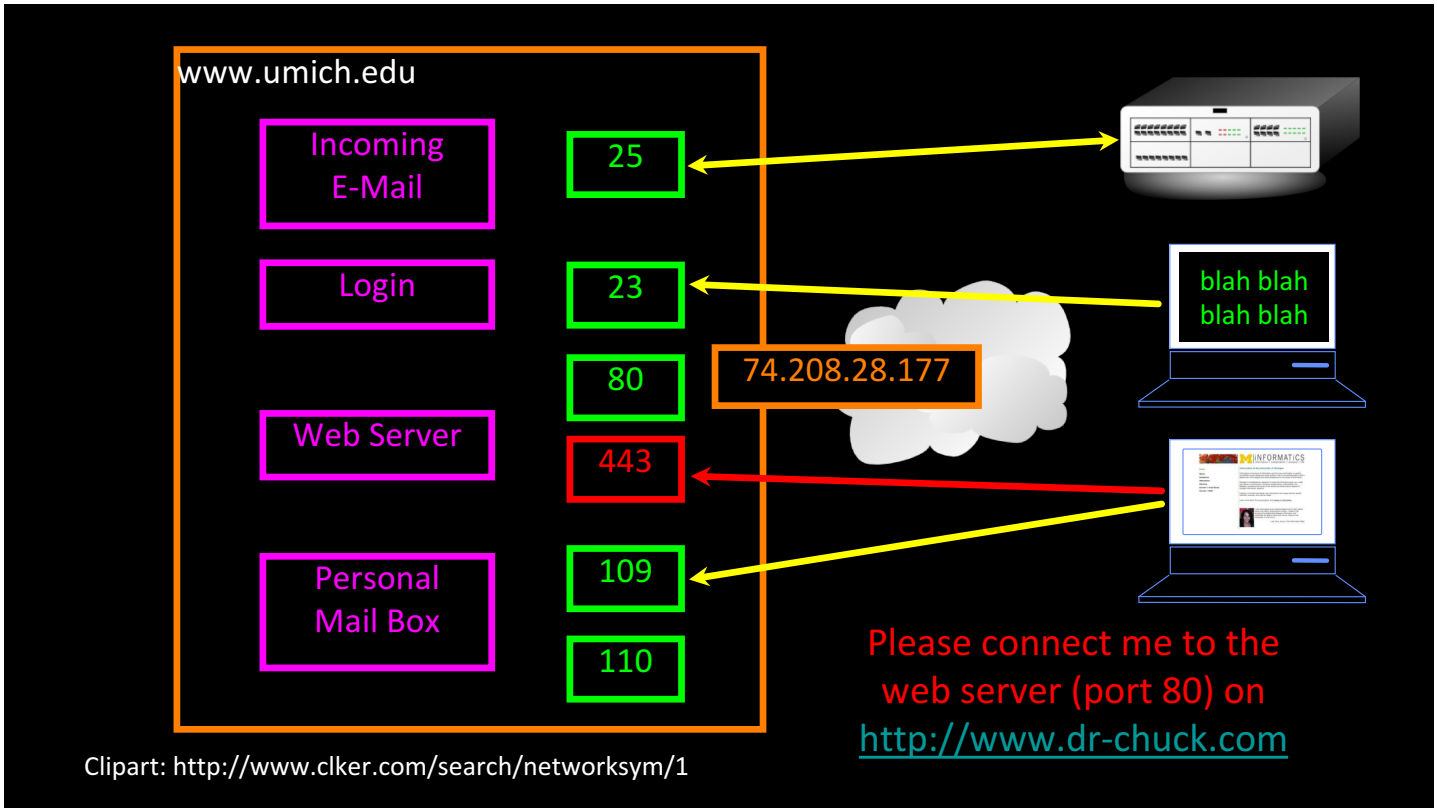


http://en.wikipedia.org/wiki/Internet_socket

TCP Port Numbers

- A port is an application-specific or process-specific software communications endpoint
- It allows multiple networked applications to coexist on the same server.
- There is a list of well-known TCP port numbers

http://en.wikipedia.org/wiki/TCP_and_UDP_port

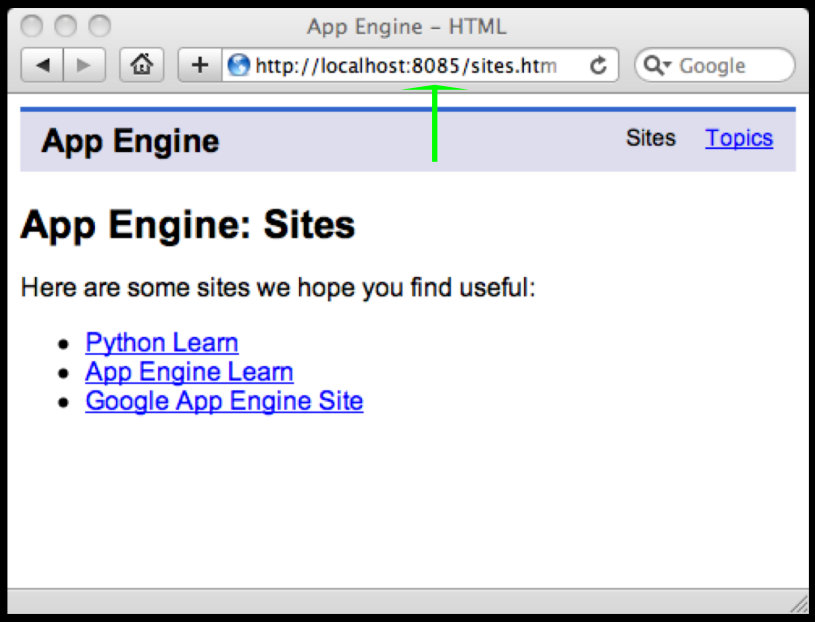


Clipart: http://www.clker.com/search/networksym/1

Common TCP Ports

- Telnet (23) - Login
- SSH (22) - Secure Login
- HTTP (80)
- HTTPS (443) - Secure
- SMTP (25) (Mail)
- IMAP (143/220/993) - Mail Retrieval
- POP (109/110) - Mail Retrieval
- DNS (53) - Domain Name
- FTP (21) - File Transfer

http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers



Sometimes we see the port number in the URL if the web server is running on a "non-standard" port.

Sockets in Python

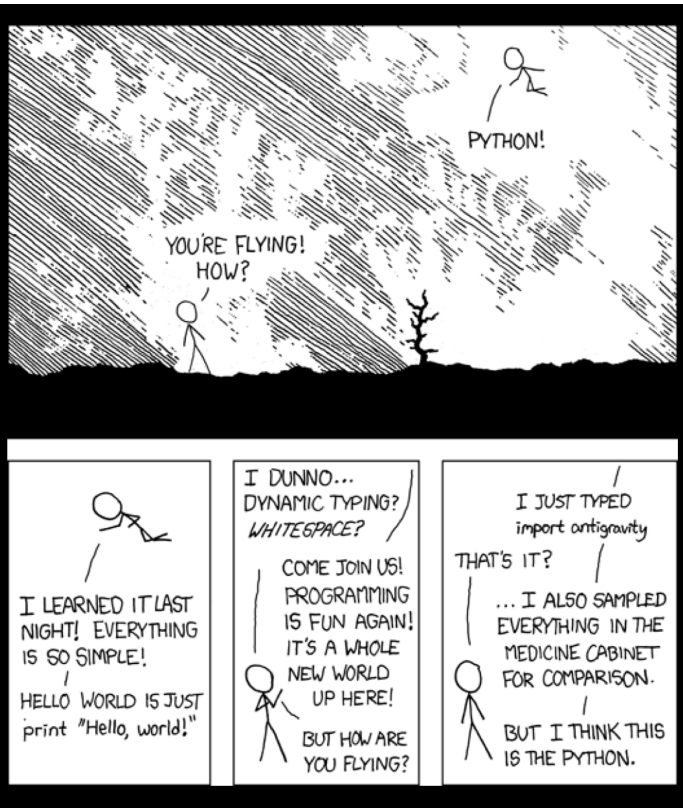
- Python has built-in support for TCP Sockets

```
import socket
mysock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect( ('www.py4inf.com', 80) )
```

Host

Port

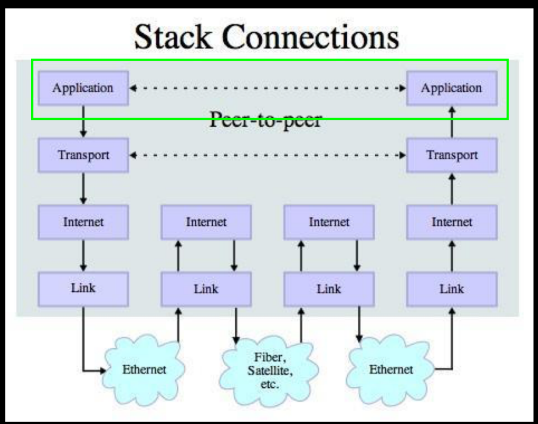
http://docs.python.org/library/socket.html



http://xkcd.com/353/

Application Protocol

- Since TCP (and Python) gives us a reliable socket, what do we want to do with the socket? What problem do we want to solve?
- Application Protocols
  - Mail
  - World Wide Web



Source: http://en.wikipedia.org/wiki/Internet_Protocol_Suite

HTTP - Hypertext Transport Protocol

- The dominant Application Layer Protocol on the Internet
- Invented for the Web - to Retrieve HTML, Images, Documents, etc
- Extended to be data in addition to documents - RSS, Web Services, etc..
- Basic Concept - Make a Connection - Request a document - Retrieve the Document - Close the Connection

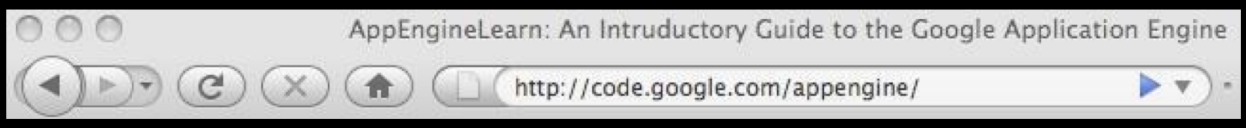
http://en.wikipedia.org/wiki/Http

HTTP

The HyperText Transport Protocol is the set of rules to allow browsers to retrieve web documents from servers over the Internet

What is a Protocol?

- A set of rules that all parties follow for so we can predict each other's behavior
- And not bump into each other
  - On two-way roads in USA, drive on the right-hand side of the road
  - On two-way roads in the UK, drive on the left-hand side of the road



http://www.dr-chuck.com/page1.htm

protocol | host | document

http://www.youtube.com/watch?v=x2GylLq59rI

1:17 - 2:19

Getting Data From The Server

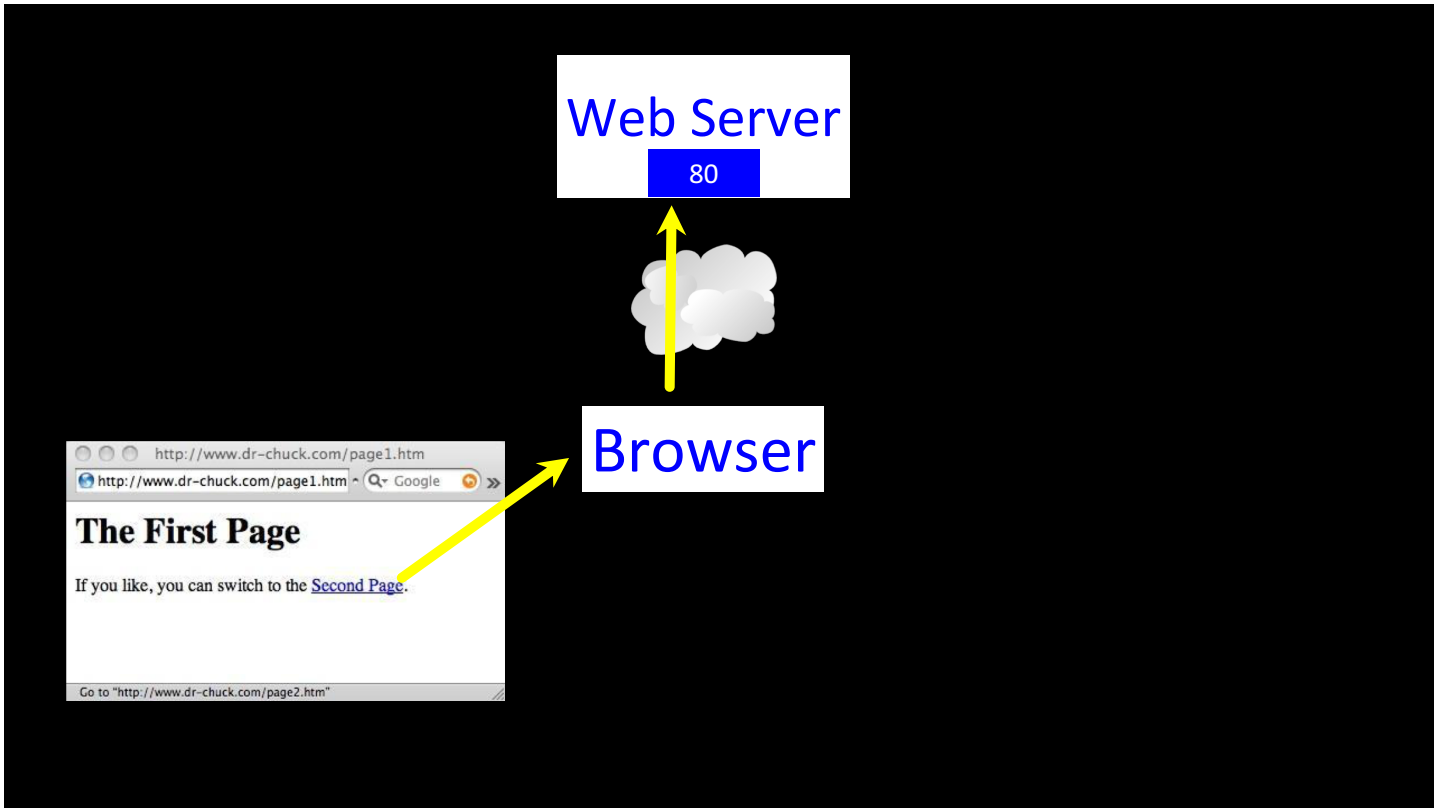
- Each time the user clicks on an anchor tag with an href= value to switch to a new page, the browser makes a connection to the web server and issues a "GET" request - to GET the content of the page at the specified URL
- The server returns the HTML document to the browser, which formats and displays the document to the user

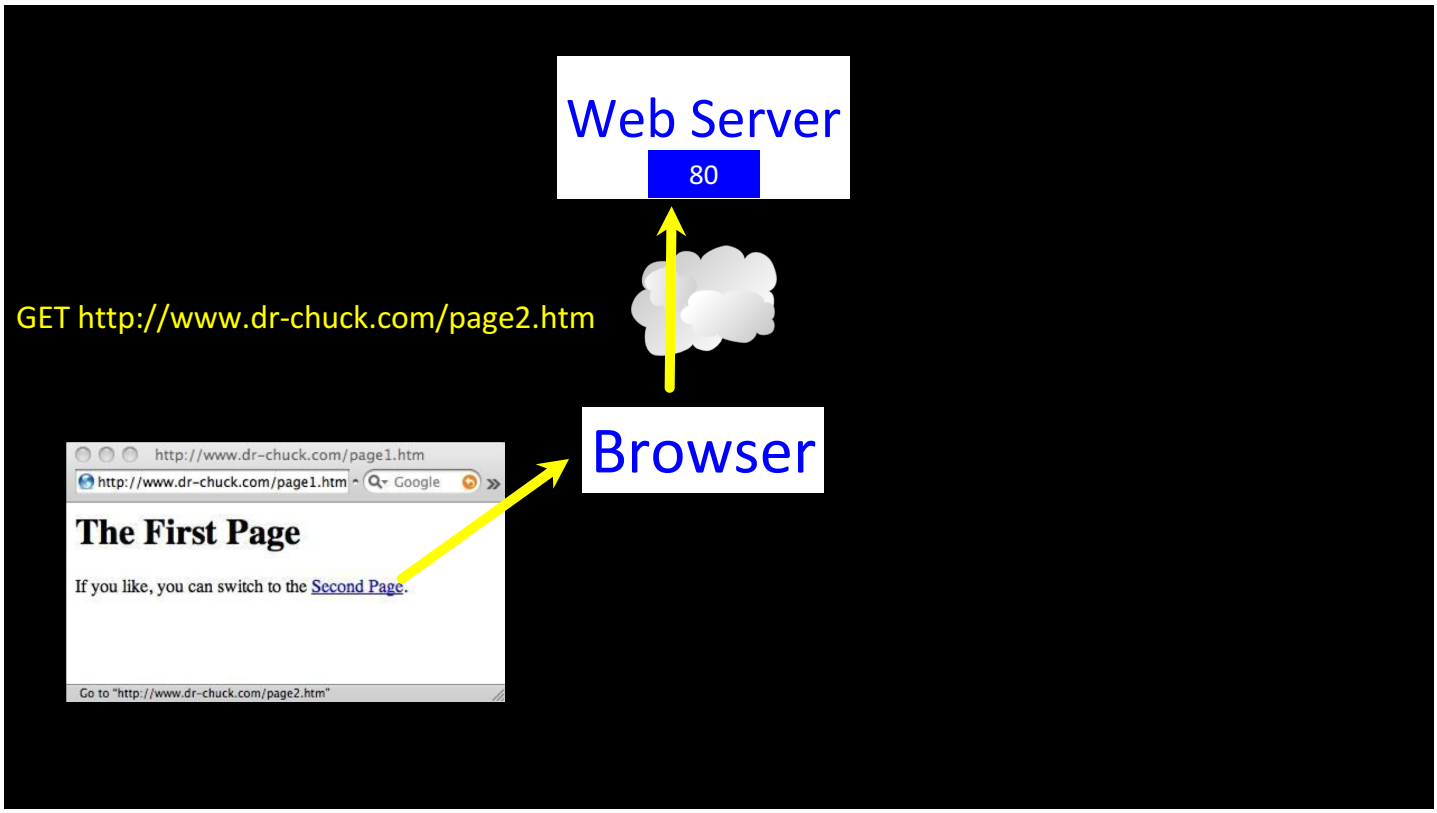
Making an HTTP request

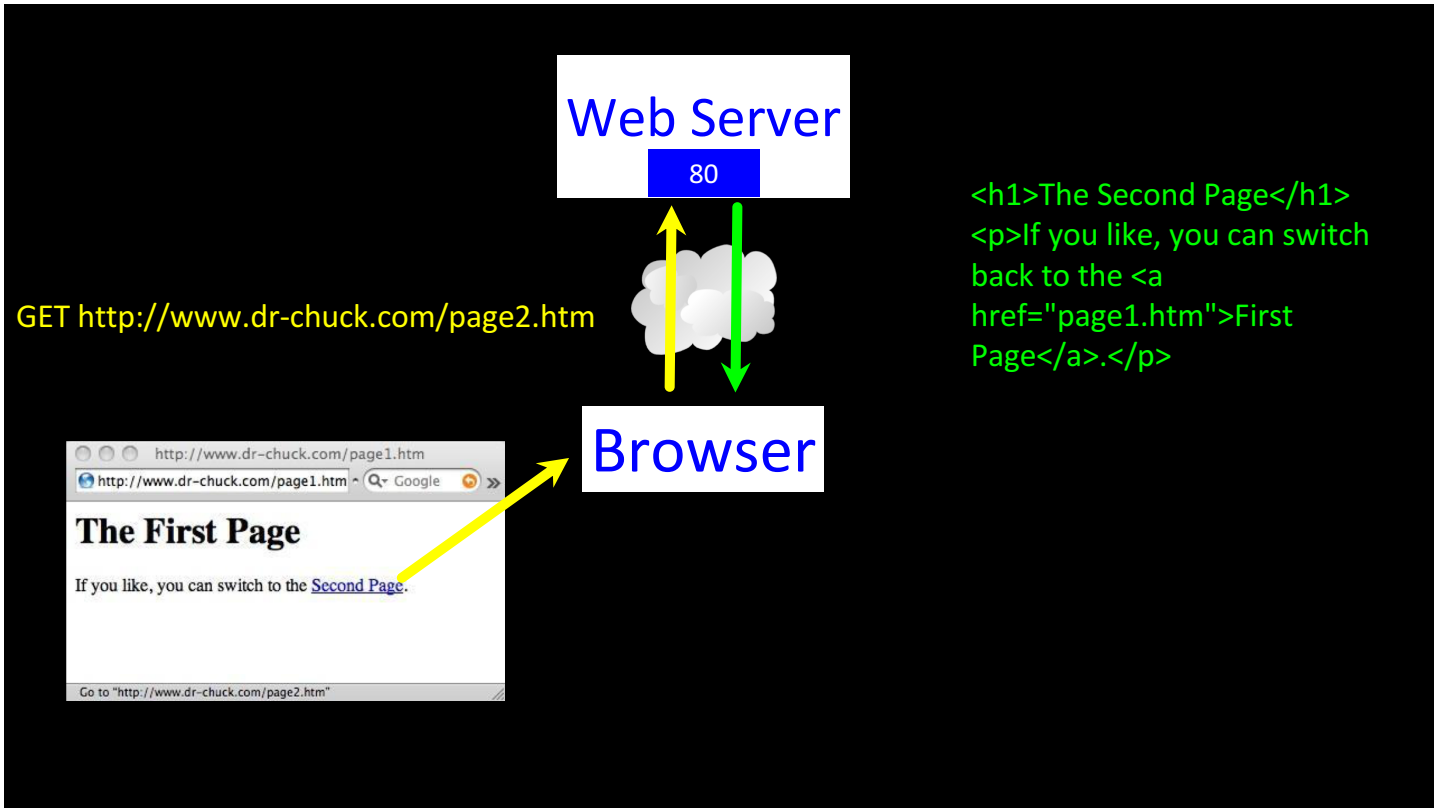
- Connect to the server like www.dr-chuck.com
  - a "hand shake"
- Request a document (or the default document)
  - GET http://www.dr-chuck.com/page1.htm
  - GET http://www.mlive.com/ann-arbor/
  - GET http://www.facebook.com

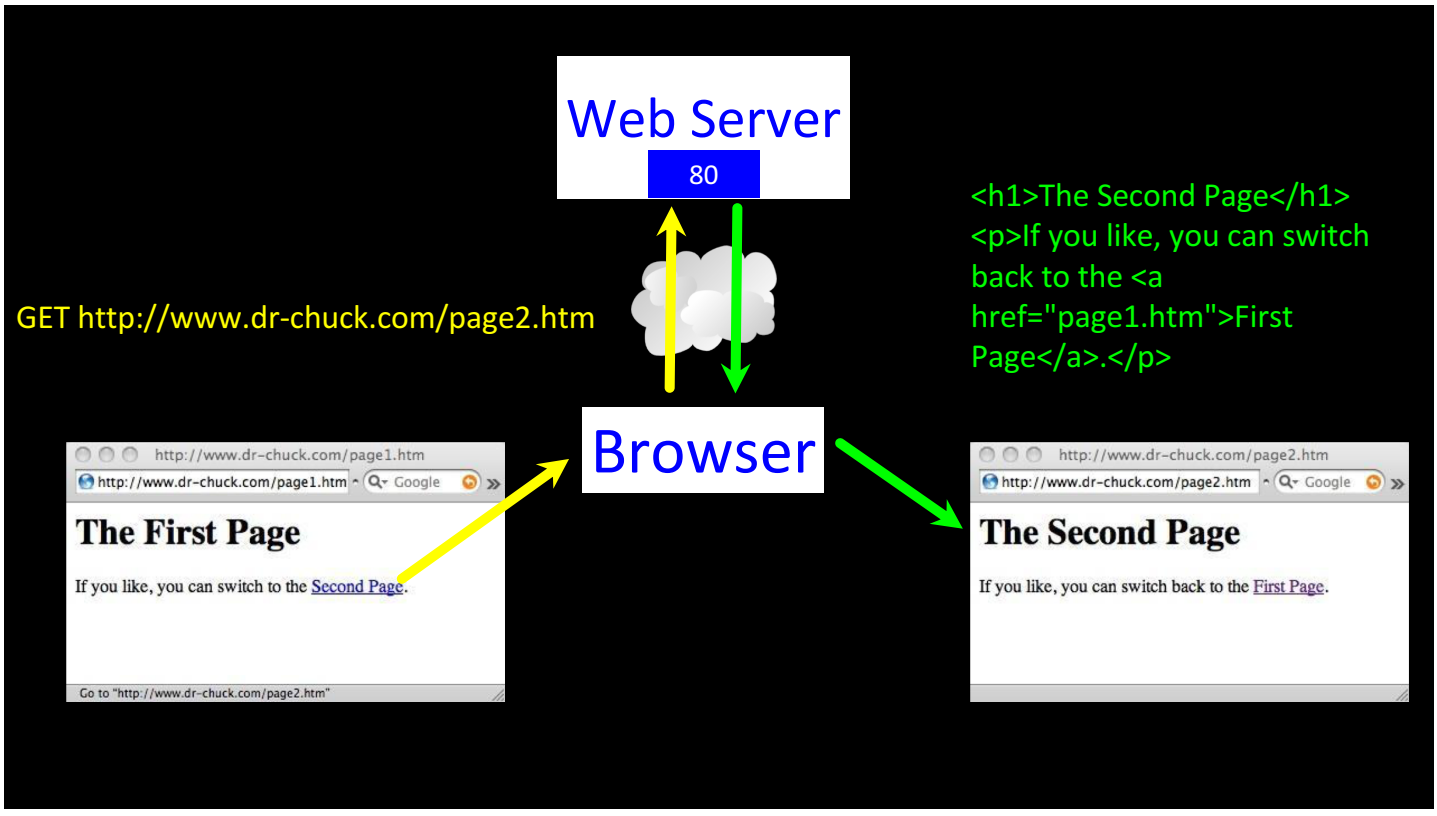

http://www.dr-chuck.com/page1.htm
http://www.dr-chuck.com/page1.htm
Google
The First Page
If you like, you can switch to the Second Page.
Go to "http://www.dr-chuck.com/page2.htm"


http://www.dr-chuck.com/page1.htm
http://www.dr-chuck.com/page1.htm
Google
The First Page
If you like, you can switch to the Second Page.
Go to "http://www.dr-chuck.com/page2.htm"
Browser


Web Server
80
Browser
http://www.dr-chuck.com/page1.htm
http://www.dr-chuck.com/page1.htm
Google
The First Page
If you like, you can switch to the Second Page.
Go to "http://www.dr-chuck.com/page2.htm"

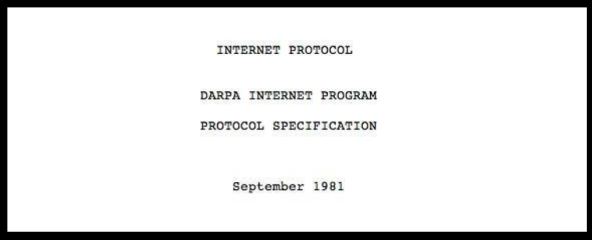

Web Server
80
GET http://www.dr-chuck.com/page2.htm
Browser
http://www.dr-chuck.com/page1.htm
http://www.dr-chuck.com/page1.htm
Google
The First Page
If you like, you can switch to the Second Page.
Go to "http://www.dr-chuck.com/page2.htm"


Web Server
80
GET http://www.dr-chuck.com/page2.htm
<h1>The Second Page</h1>
If you like, you can switch back to the First Page.
Browser
http://www.dr-chuck.com/page1.htm
The First Page
If you like, you can switch to the Second Page.


Web Server
80
GET http://www.dr-chuck.com/page2.htm
<h1>The Second Page</h1>
If you like, you can switch back to the First Page.
Browser
http://www.dr-chuck.com/page1.htm
The First Page
If you like, you can switch to the Second Page.
http://www.dr-chuck.com/page2.htm
The Second Page
If you like, you can switch back to the First Page.

Internet Standards

- The standards for all of the Internet protocols (inner workings) are developed by an organization
- Internet Engineering Task Force (IETF)
- www.ietf.org
- Standards are called “RFCs” - “Request for Comments”



INTERNET PROTOCOL

DARPA INTERNET PROGRAM

PROTOCOL SPECIFICATION

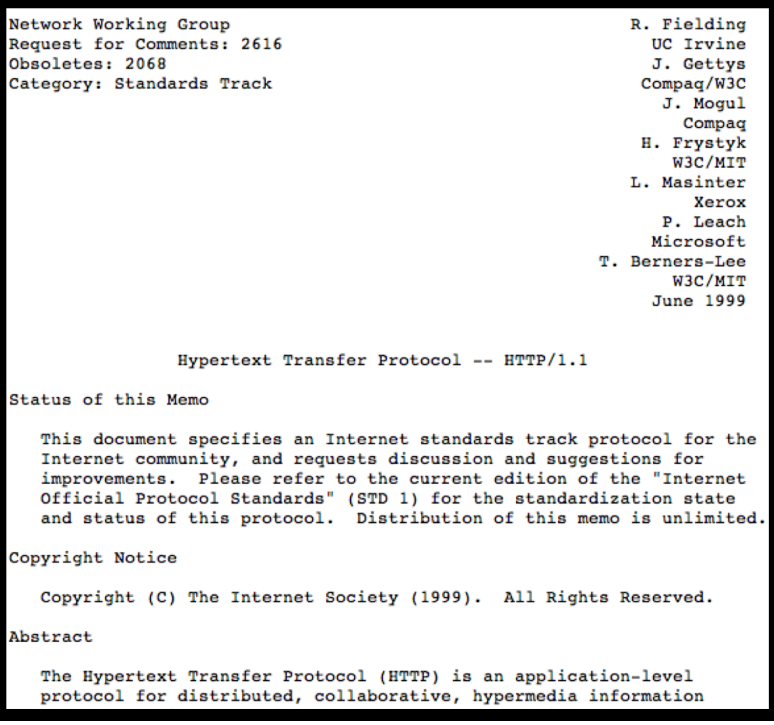
September 1981

The internet protocol treats each internet datagram as an independent entity unrelated to any other internet datagram. There are no connections or logical circuits (virtual or otherwise).

The internet protocol uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

Source: http://tools.ietf.org/html/rfc791

http://www.w3.org/Protocols/rfc2616/rfc2616.txt



Network Working Group
Request for Comments: 2616
Obsoletes: 2068
Category: Standards Track

R. Fielding, UC Irvine
J. Gettys, Compaq/W3C
J. Mogul, Compaq
H. Frystyk, W3C/MIT
L. Masinter, Xerox
P. Leach, Microsoft
T. Berners-Lee, W3C/MIT
June 1999

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

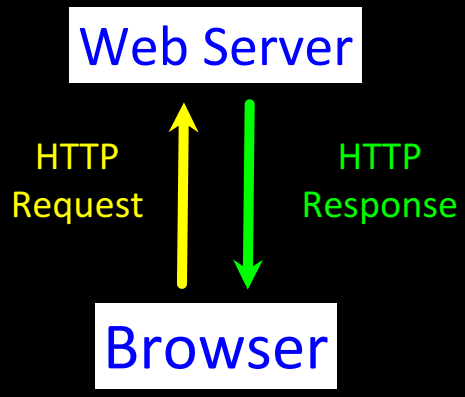
The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information

```
5 Request

   A request message from a client to a server includes, within the
   first line of that message, the method to be applied to the resource,
   the identifier of the resource, and the protocol version in use.

        Request       = Request-Line              ; Section 5.1
                        *(( general-header        ; Section 4.5
                         | request-header         ; Section 5.3
                         | entity-header ) CRLF)  ; Section 7.1
                        CRLF
                        [ message-body ]          ; Section 4.3

5.1 Request-Line

   The Request-Line begins with a method token, followed by the
   Request-URI and the protocol version, and ending with CRLF. The
   elements are separated by SP characters. No CR or LF is allowed
   except in the final CRLF sequence.

        Request-Line   = Method SP Request-URI SP HTTP-Version CRLF
```

Making an HTTP request

- Connect to the server like www.dr-chuck.com
  - a “hand shake”
- Request a document (or the default document)
  - GET http://www.dr-chuck.com/page1.htm
  - GET http://www.mlive.com/ann-arbor/
  - GET http://www.facebook.com

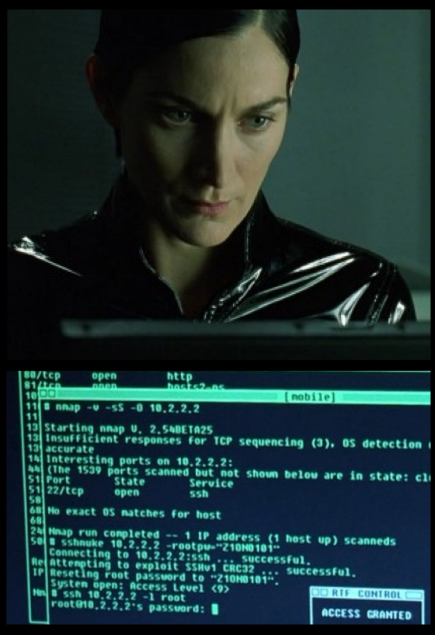
“Hacking” HTTP




```
$ telnet www.dr-chuck.com 80
Trying 74.208.28.177...
Connected to www.dr-chuck.com.
Escape character is '^]'.
GET http://www.dr-chuck.com/page1.htm HTTP/1.0

<h1>The First Page</h1>
<p>If you like, you can switch to the
<a href="http://www.dr-chuck.com/page2.htm">Second Page</a>.
</p>
```

Port 80 is the non-encrypted HTTP port

Accurate Hacking in the Movies



- Matrix Reloaded
- Bourne Ultimatum
- Die Hard 4
- …

http://nmap.org/movies.html

```
$ telnet www.dr-chuck.com 80
Trying 74.208.28.177...
Connected to www.dr-chuck.com.Escape character is '^]'.
GET http://www.dr-chuck.com/page1.htm HTTP/1.0

<h1>The First Page</h1>
<p>If you like, you can switch to the
<a href="http://www.dr-chuck.com/page2.htm">Second
Page</a>.</p>
Connection closed by foreign host.
```

```
si-csev-mbp:tex csev$ telnet www.umich.edu 80
Trying 141.211.144.190...
Connected to www.umich.edu.Escape character is '^]'.
GET /
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"><html
xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en"><head><title>University of Michigan</title><meta
name="description" content="University of Michigan is one of
the top universities of the world, a diverse public institution
of higher learning, fostering excellence in research. U-M
provides outstanding undergraduate, graduate and professional
education, serving the local, regional, national and
international communities." />
```

```
...
<link rel="alternate stylesheet" type="text/css"
href="/CSS/accessible.css" media="screen" title="accessible"
/><link rel="stylesheet" href="/CSS/print.css"
media="print,projection" /><link rel="stylesheet"
href="/CSS/other.css"
media="handheld,tty,tv,braille,embossed,speech,aural" />...
<dl><dt><a
href="http://ns.umich.edu/htdocs/releases/story.php?id=8077">
<img src="/Images/electric-brain.jpg" width="114" height="77"
alt="Top News Story" /></a><span
class="verbose">:</span></dt><dd><a
href="http://ns.umich.edu/htdocs/releases/story.php?id=8077">Sc
ientists harness the power of electricity in the
brain</a></dd></dl>
```

As the browser reads the document, it finds other URLs that must be retrieved to produce the document.

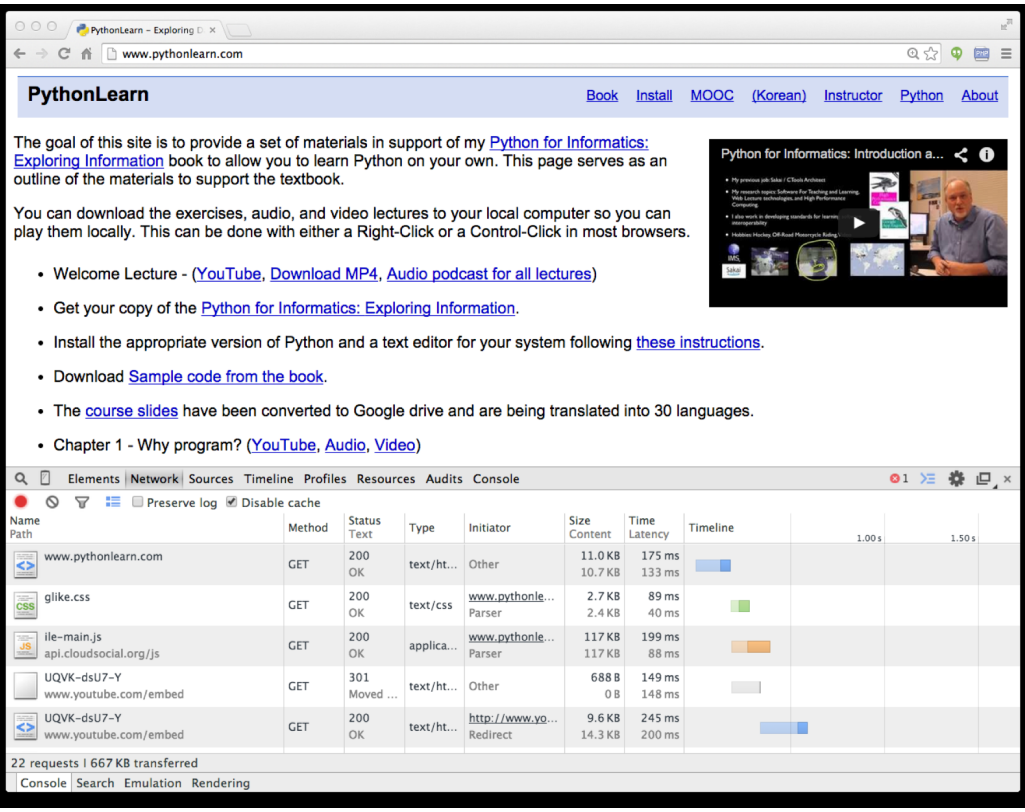
The big picture...


```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<title>University of Michigan</title>
....
```

```
@import "/CSS/graphical.css"/**/;
p.text strong, .verbose, .verbose p, .verbose h2{text-indent:-
876em;position:absolute}
p.text strong a{text-decoration:none}
p.text em{font-weight:bold;font-style:normal}
div.alert{background:#eee;border:1px solid
red;padding:.5em;margin:0 25%}
a img{border:none}
.hot br, .quick br, dl.feature2 img{display:none}
div#main label, legend{font-weight:bold}
```

A browser debugger reveals detail...

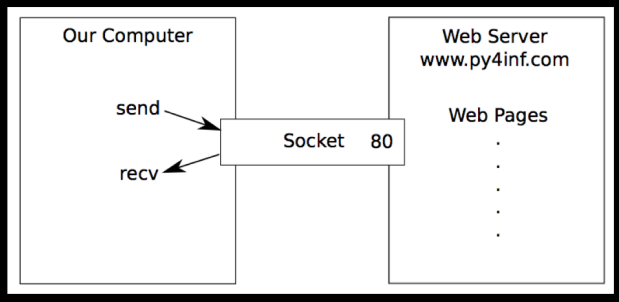
- Most browsers have a developer mode so you can watch it in action
- It can help explore the HTTP request-response cycle
- Some simple-looking pages involve lots of requests:
  - HTML page(s)
  - Image files
  - CSS Style Sheets
  - JavaScript files



Lets Write a Web Browser!

An HTTP Request in Python

```
import socket
mysock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect(('www.py4inf.com', 80))

mysock.send('GET http://www.py4inf.com/code/romeo.txt HTTP/1.0\n\n')

while True:
    data = mysock.recv(512)
    if ( len(data) < 1 ) :
        break
    print data
mysock.close()
```




```
HTTP/1.1 200 OK
Date: Sun, 14 Mar 2010 23:52:41 GMT
Server: Apache
Last-Modified: Tue, 29 Dec 2009 01:31:22 GMT
ETag: "143c1b33-a7-4b395bea"
Accept-Ranges: bytes
Content-Length: 167
Connection: close
Content-Type: text/plain

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
```

HTTP Header

```
while True:
    data = mysock.recv(512)
    if ( len(data) < 1 ) :
        break
    print data
```

HTTP Body

Making HTTP Easier With urllib

Using urllib in Python

Since HTTP is so common, we have a library that does all the socket work for us and makes web pages look like a file

```
import urllib
fhand = urllib.urlopen('http://www.py4inf.com/code/romeo.txt')

for line in fhand:
    print line.strip()
```

http://docs.python.org/library/urllib.html

urllib1.py

```
import urllib
fhand = urllib.urlopen('http://www.py4inf.com/code/romeo.txt')
for line in fhand:
    print line.strip()
```

```
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
```

http://docs.python.org/library/urllib.html

urllib1.py

Like a file...

```
import urllib
fhand = urllib.urlopen('http://www.py4inf.com/code/romeo.txt')

counts = dict()
for line in fhand:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1
print counts
```

urlwords.py

Reading Web Pages

```
import urllib
fhand = urllib.urlopen('http://www.dr-chuck.com/page1.htm')
for line in fhand:
    print line.strip()
```

```
<h1>The First Page</h1>
<p>
If you like, you can switch to the <a
href="http://www.dr-chuck.com/page2.htm">Second
Page</a>.
</p>
```

urllib2.py

Going from one page to another...

```
import urllib
fhand = urllib.urlopen('http://www.dr-chuck.com/page1.htm')
for line in fhand:
    print line.strip()
```

```
<h1>The First Page</h1>
<p>
If you like, you can switch to the <a
href="http://www.dr-chuck.com/page2.htm">Second
Page</a>.
</p>
```

Google

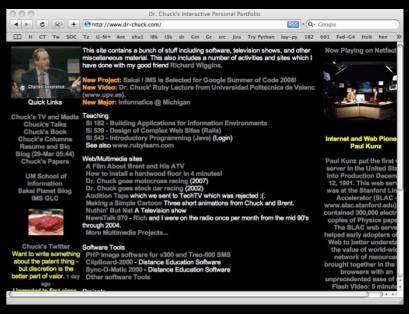
```
import urllib
fhand = urllib.urlopen('http://www.dr-chuck.com/page1.htm')
for line in fhand:
    print line.strip()
```

Parsing HTML
(a.k.a. Web Scraping)

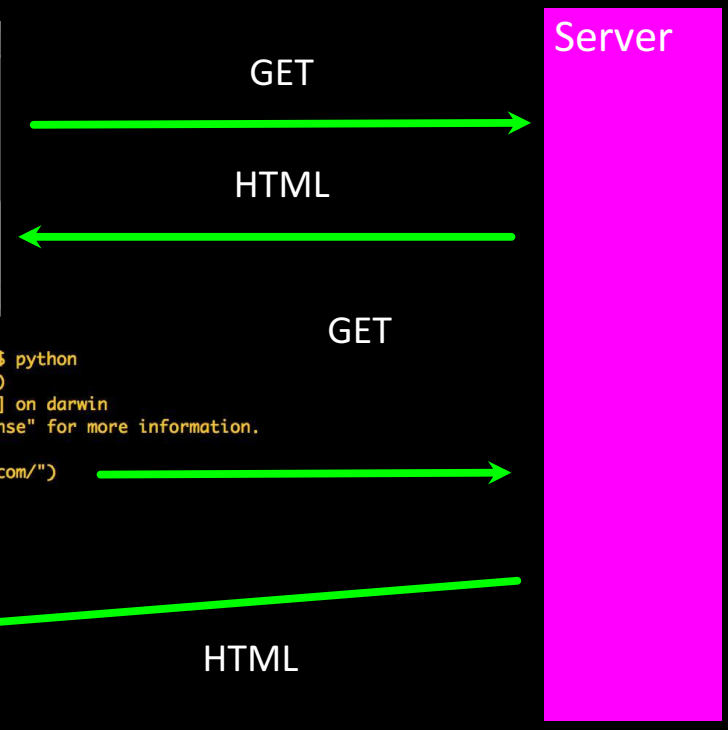
What is Web Scraping?

- When a program or script pretends to be a browser and retrieves web pages, looks at those web pages, extracts information, and then looks at more web pages.
- Search engines scrape web pages - we call this “spidering the web” or “web crawling”

http://en.wikipedia.org/wiki/Web_scraping
http://en.wikipedia.org/wiki/Web_crawler



```
charles-severances-macbook-air:Scraping csev$ python
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import urllib
>>> f = urllib.urlopen("http://www.dr-chuck.com/")
>>> contents = f.read()
>>> f.close()
>>> print len(contents)
95328
>>> print contents[0:30]
<html>
<head>
  <title>Dr. C
>>>
```



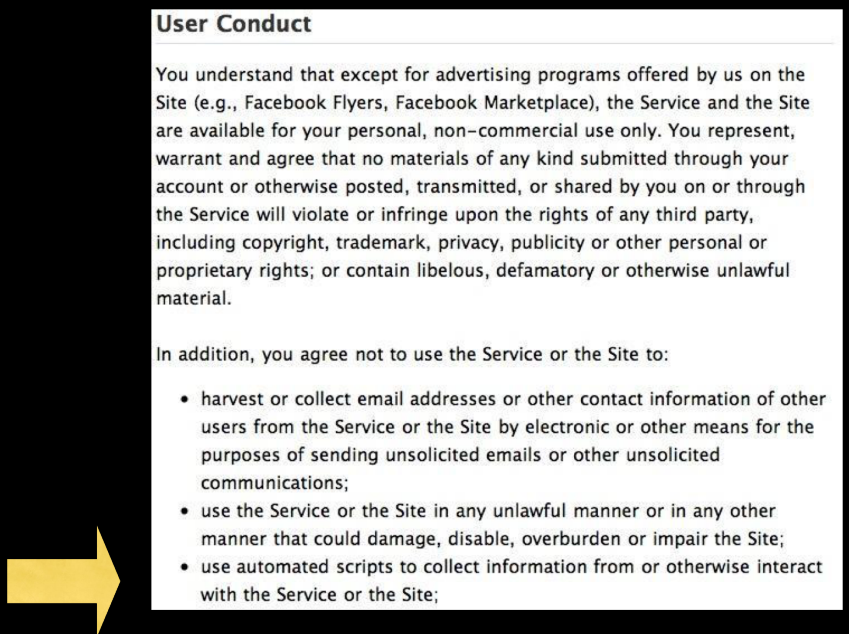
Why Scrape?

- Pull data - particularly social data - who links to who?
- Get your own data back out of some system that has no “export capability”
- Monitor a site for new information
- Spider the web to make a database for a search engine

Scraping Web Pages

- There is some controversy about web page scraping and some sites are a bit snippy about it.
  - Google: facebook scraping block
- Republishing copyrighted information is not allowed
- Violating terms of service is not allowed

http://www.facebook.com/terms.php



User Conduct

You understand that except for advertising programs offered by us on the Site (e.g., Facebook Flyers, Facebook Marketplace), the Service and the Site are available for your personal, non-commercial use only. You represent, warrant and agree that no materials of any kind submitted through your account or otherwise posted, transmitted, or shared by you on or through the Service will violate or infringe upon the rights of any third party, including copyright, trademark, privacy, publicity or other personal or proprietary rights; or contain libelous, defamatory or otherwise unlawful material.

In addition, you agree not to use the Service or the Site to:

- harvest or collect email addresses or other contact information of other users from the Service or the Site by electronic or other means for the purposes of sending unsolicited emails or other unsolicited communications;
- use the Service or the Site in any unlawful manner or in any other manner that could damage, disable, overburden or impair the Site;
- use automated scripts to collect information from or otherwise interact with the Service or the Site;

The Easy Way - Beautiful Soup

- You could do string searches the hard way
- Or use the free software called BeautifulSoup from www.crummy.com

http://www.crummy.com/software/BeautifulSoup/

Place the BeautifulSoup.py file in the same folder as your Python code...

```
import urllib
from BeautifulSoup import *

url = raw_input('Enter - ')

html = urllib.urlopen(url).read()
soup = BeautifulSoup(html)

# Retrieve a list of the anchor tags
# Each tag is like a dictionary of HTML attributes

tags = soup('a')

for tag in tags:
   print tag.get('href', None)
```

urllinks.py

```
<h1>The First Page</h1>
<p>If you like, you can switch to the<a
href="http://www.dr-
chuck.com/page2.htm">Second
Page</a>.</p>
```

```
html = urllib.urlopen(url).read()
soup = BeautifulSoup(html)

tags = soup('a')
for tag in tags:
   print tag.get('href', None)
```

```
python urllinks.py
Enter - http://www.dr-chuck.com/page1.htm
http://www.dr-chuck.com/page2.htm
```

Summary

- The TCP/IP gives us pipes / sockets between applications
- We designed application protocols to make use of these pipes
- HyperText Transport Protocol (HTTP) is a simple yet powerful protocol
- Python has good support for sockets, HTTP, and HTML parsing

Acknowledgements / Contributions

Initial Development: Charles Severance, University of Michigan School of Information

… Insert new Contributors here

...