Intro to APIs

Sam Maurer

October 7, 2015
CP 255 Urban Informatics
UC Berkeley

What’s an API?

- “Application Programming Interface”
- A **code-based interface** for outside developers to interact with a piece of software, anything from a code library to a website to a database system

Restaurant menu as API

REST API:

get_food()
required parameters:
    item name

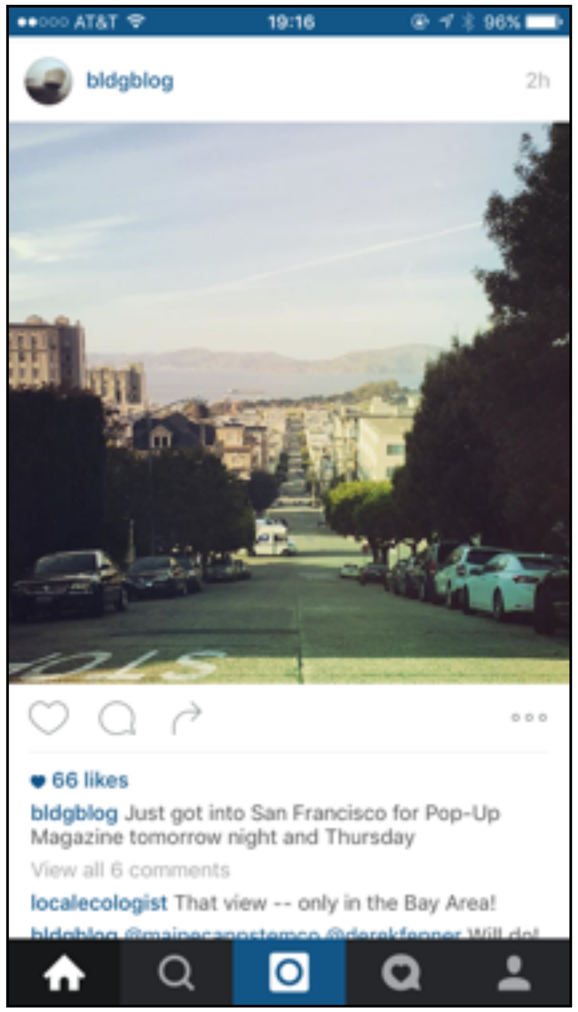
get_coffee()
optional parameters:
    type

get_food(item=
“lentil salad”)

get_coffee()

get_recipe(item=
“lentil salad”)

Real APIs in action



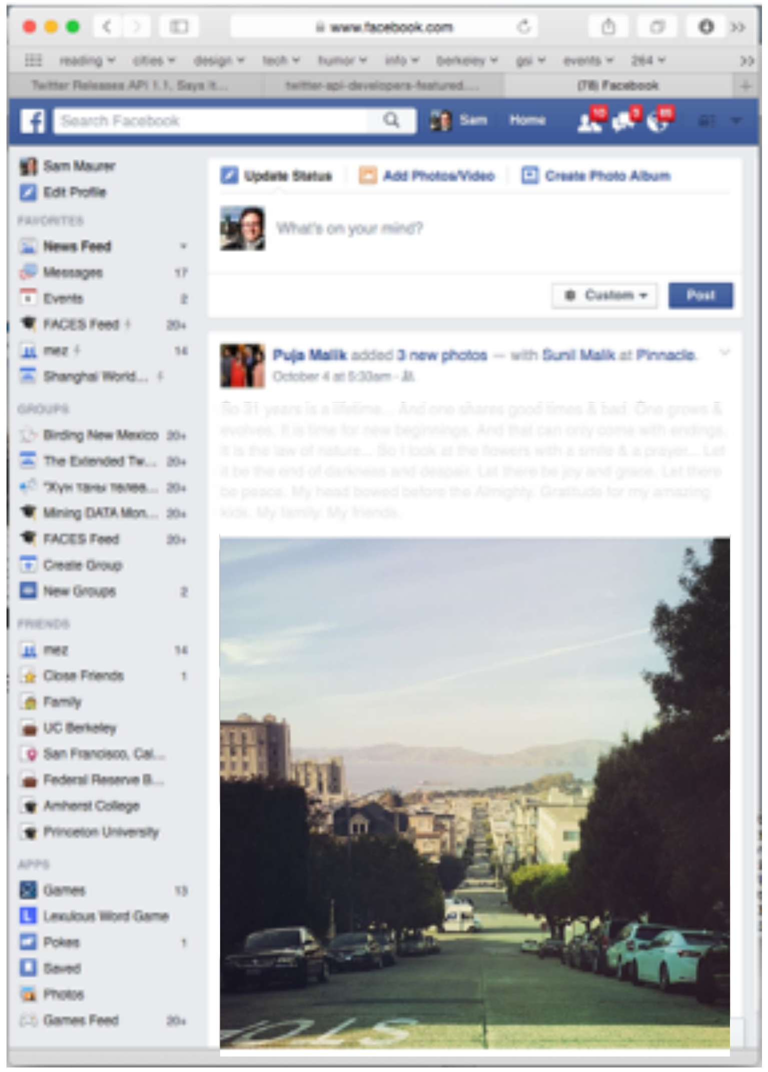
bldgblog Just got into San Francisco for Pop-Up Magazine tomorrow night and Thursday

View all 6 comments

localecologist That view -- only in the Bay Area!

Instagram photo

$$\xrightarrow{\hspace{2em}} \boxed{\text{Facebook Graph API}} \xrightarrow{\hspace{2em}}$$

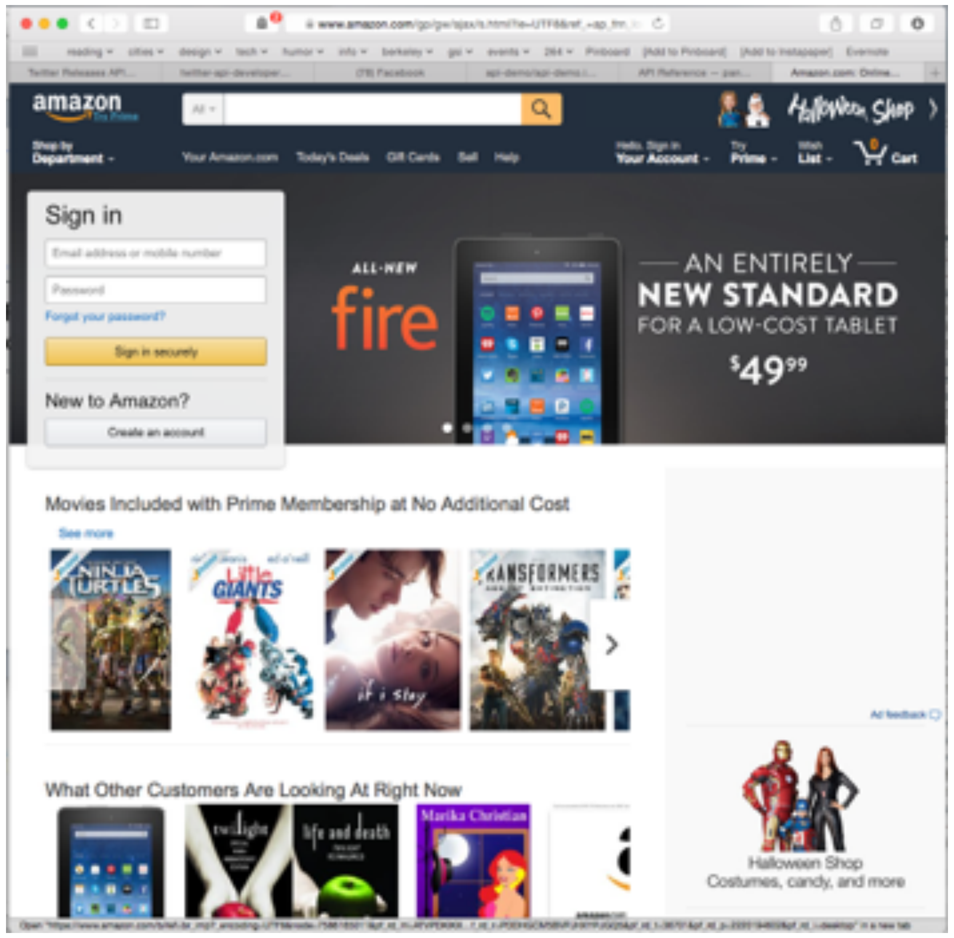


Puja Malik added 3 new photos — with Sunil Malik at Pinnacle.

October 4 at 5:33am

Posted to Facebook

Real APIs in action


amazon
Sign in
Forgot your password?
Sign in securely
New to Amazon?
Create an account
ALL-NEW fire
AN ENTIRELY NEW STANDARD FOR A LOW-COST TABLET
$49.99
Movies Included with Prime Membership at No Additional Cost
See more
What Other Customers Are Looking At Right Now
Halloween Shop
Costumes, candy, and more

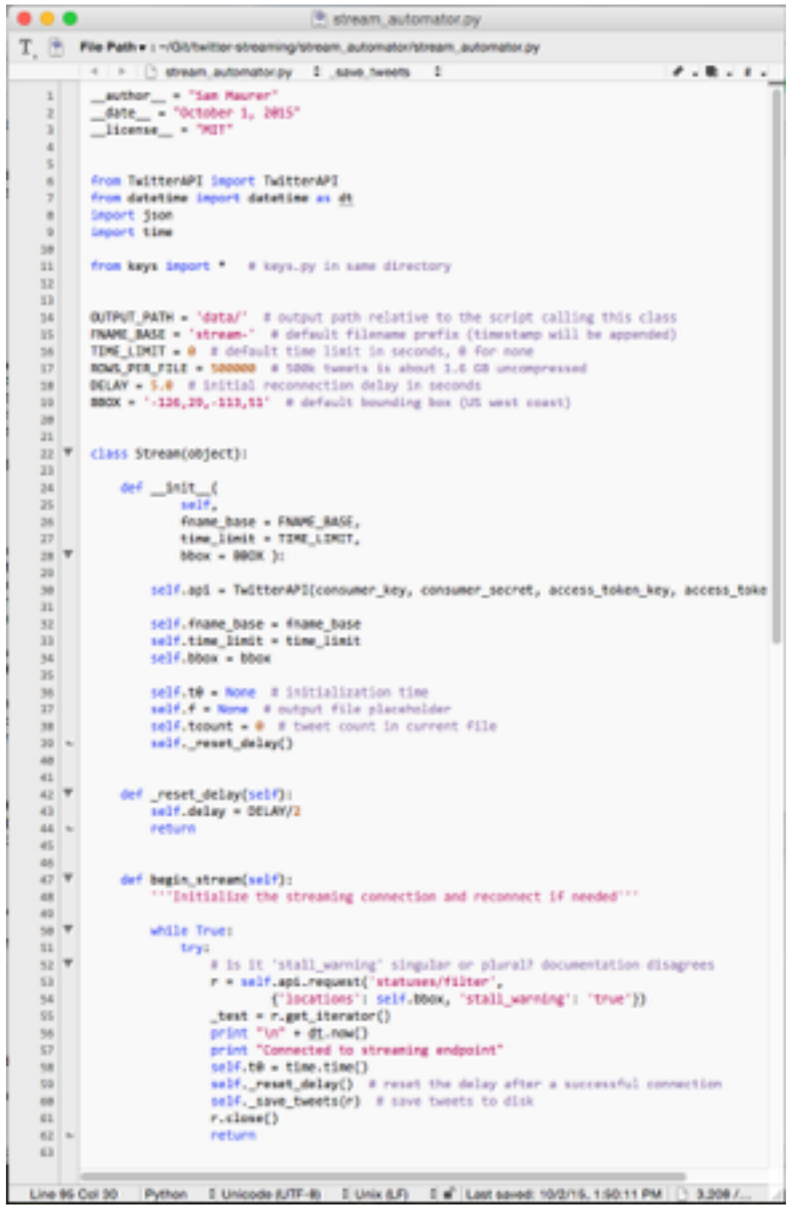
Discover beautiful things on Amazon, updated daily
SHOP
amazon
Amazon Fashion
THE 2015 COAT GUIDE
BRACE FOR THE CHILL IN FALL'S MOST FASHIONABLE—AND FUNCTIONAL—COATS
Try Amazon Prime Today and Get Unlimited Fast, Free Shipping
See more

Amazon Recommendations API

Real APIs in action

df = pd.DataFrame()

Pandas
API


stream_automator.py
File Path ▾ : ~/Git/twitter-streaming/stream_automator/stream_automator.py
stream_automator.py
_save_tweets

```python
__author__ = "Sam Maurer"
__date__ = "October 1, 2015"
__license__ = "MIT"


from TwitterAPI import TwitterAPI
from datetime import datetime as dt
import json
import time

from keys import *   # keys.py in same directory


OUTPUT_PATH = 'data/'  # output path relative to the script calling this class
FNAME_BASE = 'stream-'  # default filename prefix (timestamp will be appended)
TIME_LIMIT = 0  # default time limit in seconds, 0 for none
ROWS_PER_FILE = 500000  # 500k tweets is about 1.6 GB uncompressed
DELAY = 5.0  # initial reconnection delay in seconds
BBOX = '-126,29,-113,51'  # default bounding box (US west coast)


class Stream(object):

    def __init__(
            self,
            fname_base = FNAME_BASE,
            time_limit = TIME_LIMIT,
            bbox = BBOX ):

        self.api = TwitterAPI(consumer_key, consumer_secret, access_token_key, access_toke

        self.fname_base = fname_base
        self.time_limit = time_limit
        self.bbox = bbox

        self.t0 = None  # initialization time
        self.f = None  # output file placeholder
        self.tcount = 0  # tweet count in current file
        self._reset_delay()


    def _reset_delay(self):
        self.delay = DELAY/2
        return


    def begin_stream(self):
        '''Initialize the streaming connection and reconnect if needed'''

        while True:
            try:
                # is it 'stall_warning' singular or plural? documentation disagrees
                r = self.api.request('statuses/filter',
                        {'locations': self.bbox, 'stall_warning': 'true'})
                _test = r.get_iterator()
                print "\n" + dt.now()
                print "Connected to streaming endpoint"
                self.t0 = time.time()
                self._reset_delay()  # reset the delay after a successful connection
                self._save_tweets(r)  # save tweets to disk
                r.close()
                return

```

Line 95 Col 30 | Python | Unicode (UTF-8) | Unix (LF) | Last saved: 10/2/15, 1:50:11 PM | 3,208 /...

```
   depth  magnitude
0  48.82       4.50
1   9.52       4.70
2  11.18       2.97
3   4.09       2.65
4  26.07       4.60
```

Lessons

- An API is a **code-based interface** for outside developers to interact with a piece of software
- APIs create a **clear menu of access points**, shielding the inner workings of the software from outside view and allowing them to change as needed
- APIs are usually **transactional**: you submit a query and then receive something in return

Categories of APIs

APIs in a coding language

Functions:
my_function()

Arguments:
my_function(args="x")

Function returns a value

APIs over the web

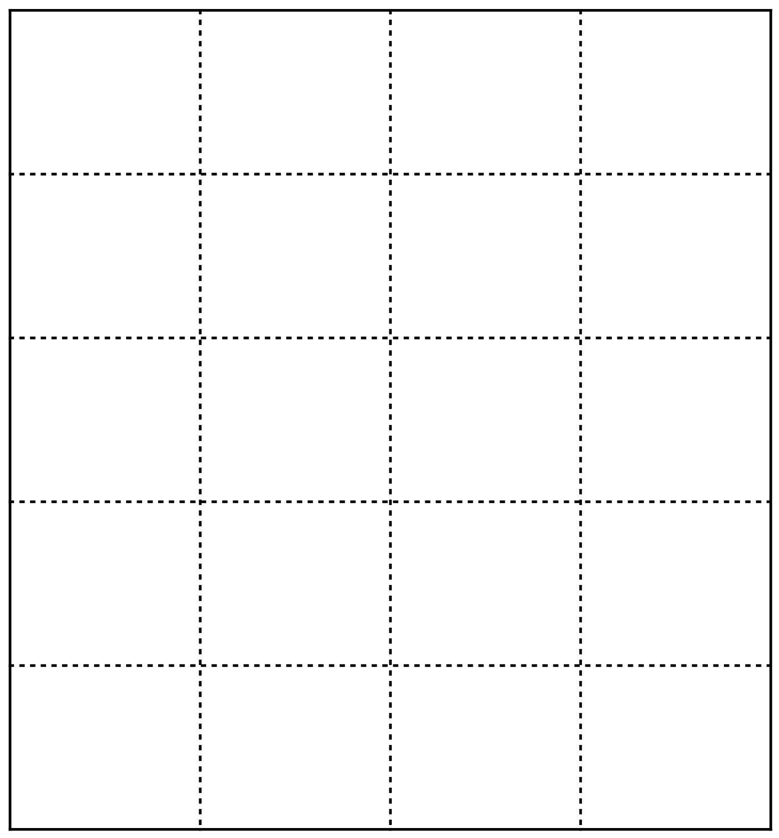
URL endpoints:
http://my.domain/endpoint

Query parameters:
?args=x

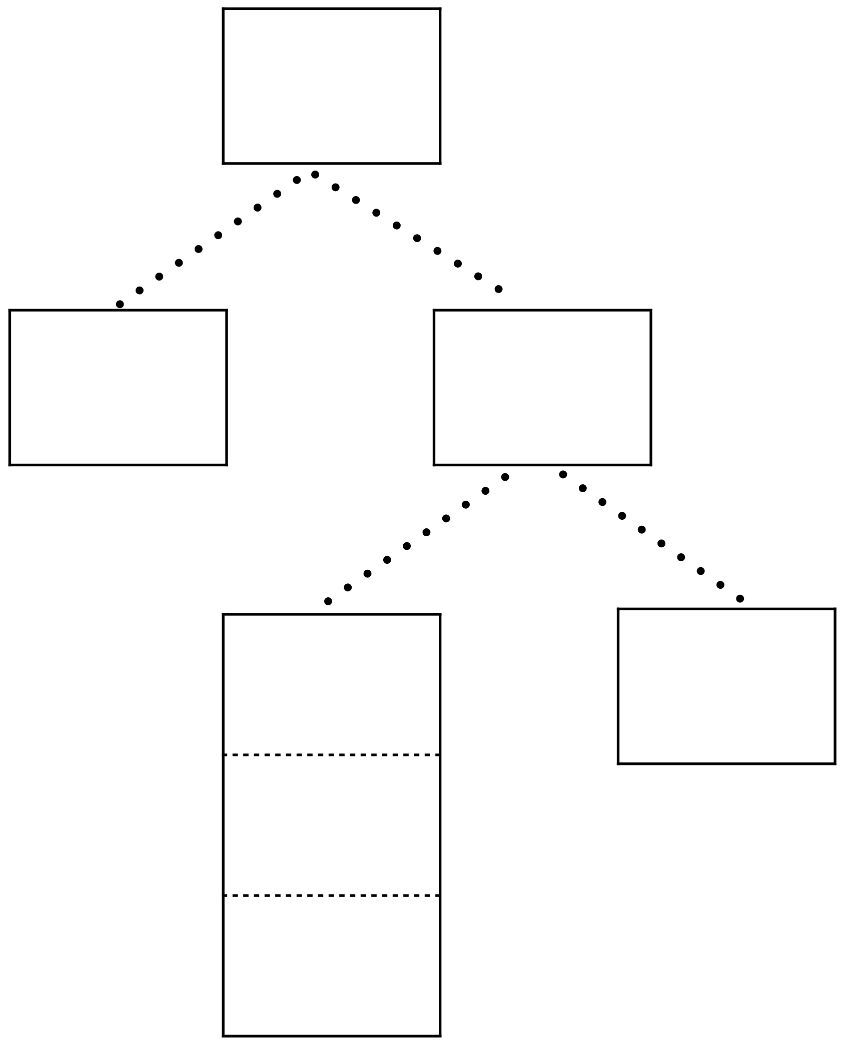
Web request returns a value

Data formats

Table
(CSV, DataFrame)



Nested arrays
(JSON, XML)



More lessons

- APIs that operate over the web have **URL endpoints** and **query parameters**, which are similar to functions and arguments
- They usually return data as **nested arrays** in JSON format, which you can reassemble into tables

Demo!