
YQL
An API for the Web
Developer Talk Token Unicorn

Derek Gathright

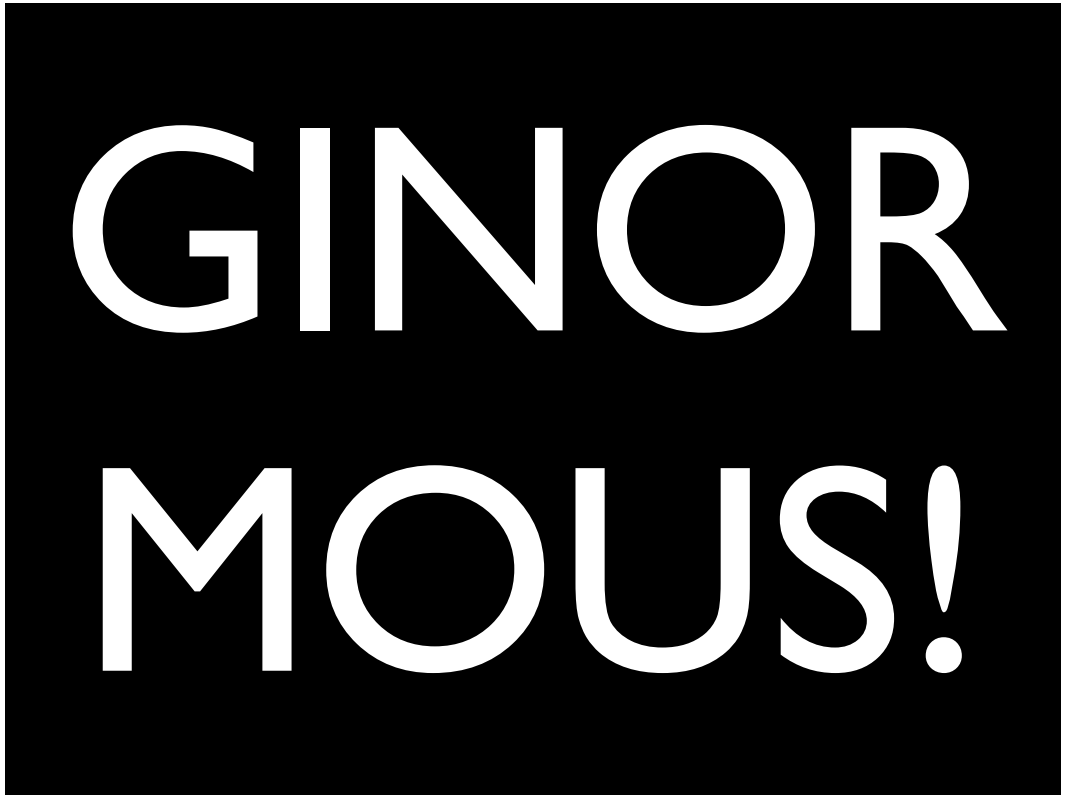
- Engineer @ Yahoo
- Tweet from @derek
- Blog @ http://derekville.net
- Everything else @ http://derek.io

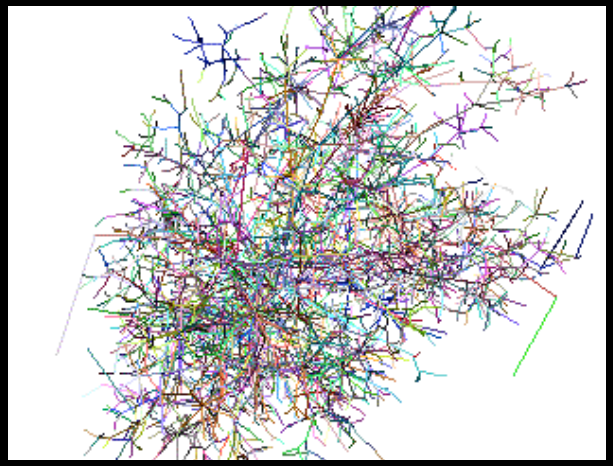
On December 4th, 1995

“Netscape and Sun announce JavaScript, a cross-platform object scripting language”

http://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html

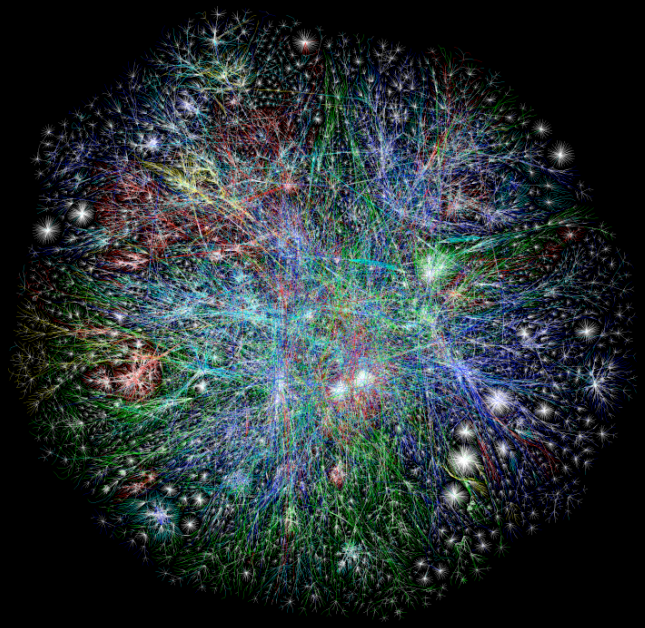
How big is the Web?


GINOR
MOUS!



The Internet: 1995

http://www.jevans.com/pubnetmap.html

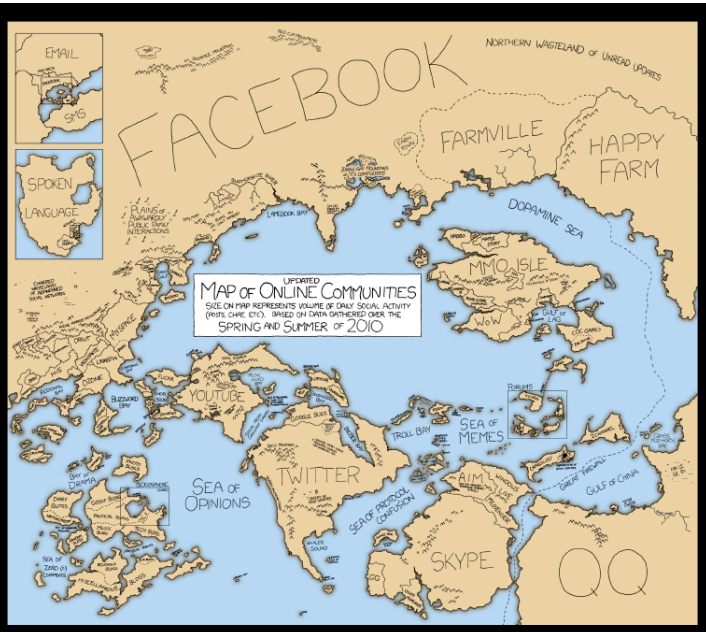


The Internet: 2003

http://www.opte.org/maps/

The Internet: 2007

http://xkcd.com/256/

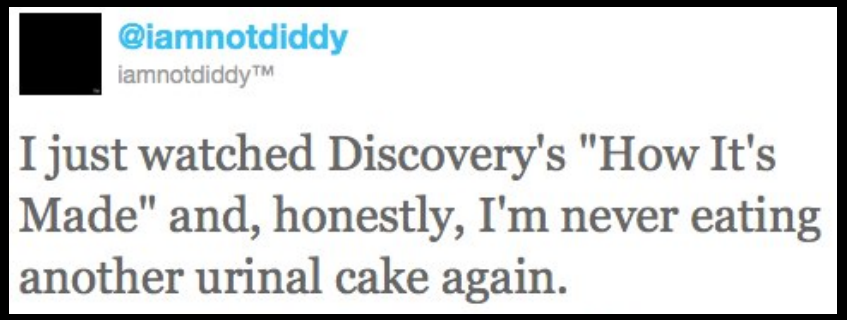


The Internet: 2010

http://xkcd.com/802/

It's impossible to measure the size of the web because it is constantly changing, growing, and morphing.

Every second:
Twitter gets 600 new tweets



Every minute:
YouTube +35 hours of video

Every month: Facebook gets 2.5 billion new photos

Yeah, lots of it is garbage

But there's still a ton of interesting stuff out there.

So how do you access it, programmatically?

It is easy enough to ‘scrape’ the web (using cURL, wget, etc...), but how do you parse it?

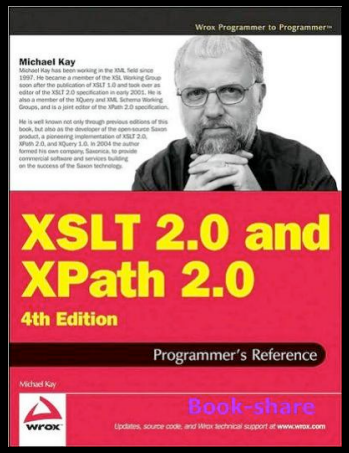
XPath + DOM Traversal = Yay!

Regular Expressions = Double Yay!

"If a regular expression is longer than 2 inches, find another method"

- Douglas Crockford


/^((?>[a-zA-Z\d!#$%&'*+\-/=?^_`{|}~]+\x20*|"((?=[\x01-\x7f])[^"\\]|\\[\x01-\x7f])*"\x20*)*(?<angle><))?((?!\.)(?>\.?[a-zA-Z\d!#$%&'*+\-/=?^_`{|}~]+)+|"((?=[\x01-\x7f])[^"\\]|\\[\x01-\x7f])*")@(((?!-)[a-zA-Z\d\-]+(?<!-)\.)+[a-zA-Z]{2,}|\[(((?(?<!\[)\.)(25[0-5]|2[0-4]\d|[01]?\d?\d)){4}|[a-zA-Z\d\-]*[a-zA-Z\d]:((?=[\x01-\x7f])[^\\\[\]]|\\[\x01-\x7f])+)\])(?(angle)>)$/



4.4 lbs &1,368 pages.
No thanks!

The Point?

The Web has a ton of data, but no easy, **hackday**-able way to access it.

THE WEB
NEEDS
AN API!

APIs are awesome.
You get (mostly) whatever data you want in (mostly) whatever format you want and in a (mostly) easy to parse structure.

Example...

http://api.twitter.com/1/users/show.json?screen_name=derek


```
{
  profile_image_url: http://a1.twimg.com/profile_images/295683345/self1_normal.
  description: "Engineer at Yahoo, PHP Developer (ZCE), JavaScripter, Palm fanb
  profile_background_color: "022330",
  url: http://derekville.net/,
- status: {
    in_reply_to_screen_name: null,
    in_reply_to_status_id_str: null,
    contributors: null,
    retweet_count: 0,
    in_reply_to_user_id_str: null,
    created_at: "Thu Dec 02 06:10:23 +0000 2010",
    id_str: "10214172507766784",
    retweeted: false,
    source: "web",
    geo: null,
    in_reply_to_user_id: null,
    in_reply_to_status_id: null,
    truncated: false,
    place: null,
    coordinates: null,
    favorited: false,
    id: 10214172507766784,
    text: "Working on my #timesopen slidedeck.  Did I just find a way to work
  },
  screen_name: "derek",
  lang: "en",
  statuses_count: 6170,
  time_zone: "Pacific Time (US & Canada)",
  created_at: "Mon Jul 17 16:09:19 +0000 2006",
  profile_text_color: "333333",
  location: "Los Angeles, CA",
  id_str: "1974",
  follow_request_sent: false,
```

Companies discovered that if you build APIs, developers will come in droves

And it saves you from having to dance around on stage like a monkey

(if you are confused, Google "monkey dance")

Yahoo, Google,
Facebook, Twitter,
Microsoft, NY Times, ...

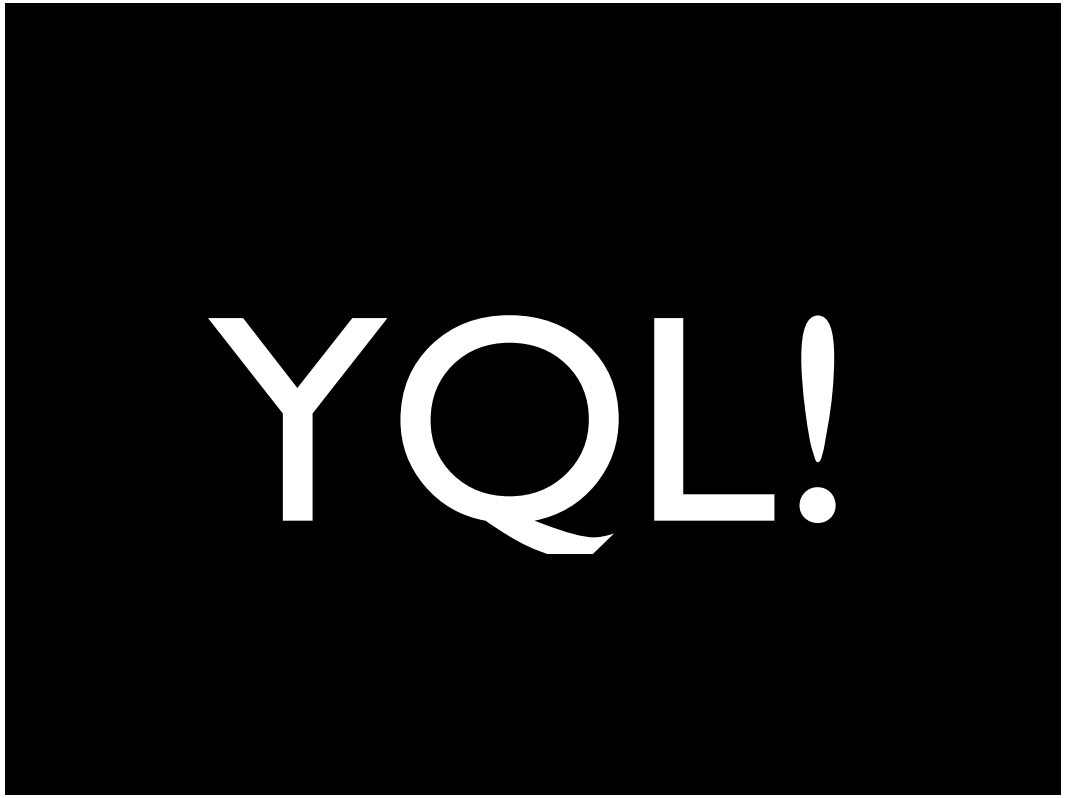
Most web companies
offer APIs now days.

As neat as they are,
they are imperfect.
Why?
You have to read
documentation to use
them.
BOOOOOOO!!!!

We're developers,
we're lazy,
and we want to build stuff,

NOW!

Yahoo invented a solution to this problem...


YQL!

"Yahoo! Query Language is an expressive SQL-like language that lets you query, filter, and join data across Web services.

With YQL, apps run faster with fewer lines of code and a smaller network footprint."

http://developer.yahoo.com/yql/

You now have a common way of accessing any data on the web, with or without APIs, and all in one common syntax that you are likely familiar with, SQL.

YQL is...

- RESTful
- Scaleable
- Customizable
- ... and lots of other “ables”

How do you use it?

```
$format = "json"; // or xml;
$base = "http://query.yahooapis.com/v1/public/yql";
$url = "{$base}?q={$yql_query}&format={$format}";
$json_string = goGetIt($url);  // likely a curl() call
$data = json_decode($json);
```

Or use any of the libraries written for your favorite language/framework

YQL Queries

SELECT {fields}
FROM {table}
WHERE {conditions}

```
SELECT * FROM
weather.forecast
WHERE
location=90210
```

```
SELECT * FROM
data.html.cssselect
WHERE
url="http://yahoo.com"
AND css="body a";
```

```
SELECT height,width,url
FROM search.images
WHERE
query="kitteh" AND
mimetype LIKE "%jpeg%"
```

SELECT * FROM
google.search
WHERE
q="pizza"

```
SELECT status.text FROM
twitter.user.timeline
WHERE
screen_name="derek"
```

```
SELECT * FROM
foursquare.history
WHERE
username="foo" AND
password="bar"
```

```
SELECT * FROM rss WHERE
url IN (SELECT title FROM
atom WHERE url="http://
spreadsheets.google.com/feeds/list/
pg_T0Mv3iBwlJoc82J1G8aQ/od6/public/
basic") LIMIT 10 | unique
(field="title")
```

Where's the magic?

Data Tables

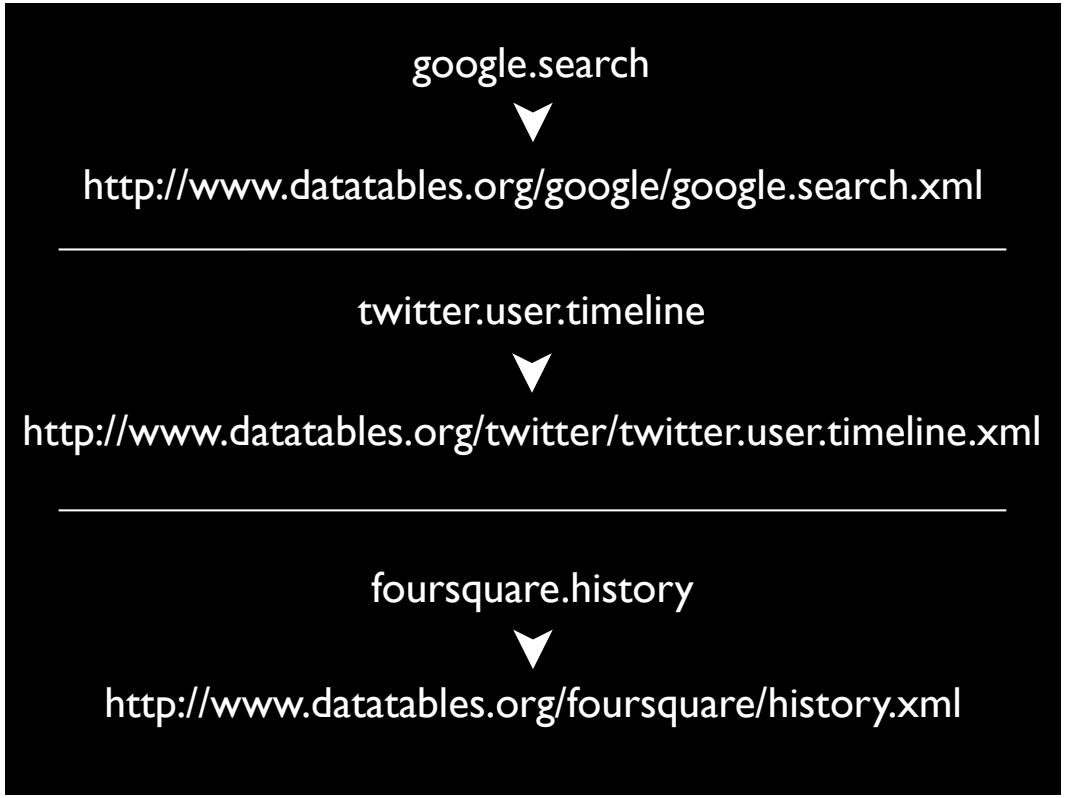
13 categories & counting, including...

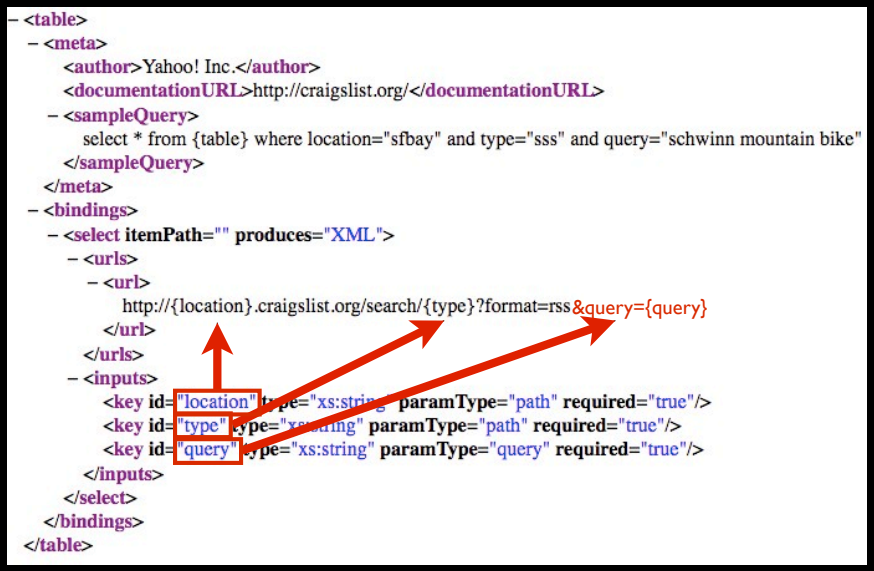
- Geo
- Flickr
- Local
- Maps
- Meme
- Music
- Social
- Upcoming
- Weather
- Yahoo (Search)
- YMail
- YQL (Storage)

Open Data Tables

900+ community contributed tables in hundreds of categories, including...

- Amazon
- Craigslist
- Facebook
- Foursquare
- Google
- HackerNews
- LastFM
- Netflix
- NY Times
- SimpleGeo
- SPARQL
- Twitter
- Wordpress
- YouTube


google.search
http://www.datatables.org/google/google.search.xml
twitter.user.timeline
http://www.datatables.org/twitter/twitter.user.timeline.xml
foursquare.history
http://www.datatables.org/foursquare/history.xml



http://www.datatables.org/craigslist/craigslist.search.xml

YQL != Voodoo Magic

It is just rewriting a YQL query into one (or many) HTTP calls for you.

```
USE "http://www.datatables.org/nyt/nyt.bestsellers.xml"
AS nyt.bestsellers;

USE "https://github.com/gcb/yql.opentable/raw/master/
text.concat.xml"
AS text.concat;

SELECT text FROM text.concat
    WHERE text.key1 = "http://www.amazon.com/dp/" AND
    (text.key2) IN (
        SELECT isbns.isbn.isbn10
        FROM nyt.bestsellers
        WHERE apikey='yourAPIKey'
    );

// Generates strings like "http://www.amazon.com/dp/031603617X"
```

```
USE "https://github.com/gcb/yql.opentable/raw/master/
text.concat.xml"
AS text.concat;
```

https://github.com/gcb/yql.opentable/raw/master/text.concat.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<table xmlns="http://query.yahooapis.com/v1/schema/table.xsd">
    <bindings>
        <select itemPath="" produces="XML">
            <urls>
                <url/>
            </urls>
            <inputs>
                <map id="text" type="xs:string" paramType="variable" required="true"/>
            </inputs>
            <execute><![CDATA[
               var results = "";
               var keys = [];

               for (k in text) {
                  keys.push(k);
               }

               keys.sort();

               for (var i=0;i<keys.length;i++) {
                  key = keys[i];
                  value = text[key];
                  results += value;
               }

               results = <entry><text>{results}</text></entry>;

               response.object = results;
            ]]></execute>
        </select>
    </bindings>
</table>
```

<execute>

- Execute arbitrary JavaScript in Rhino (a JS engine)
- E4X Support (XML literals in JS)
- Speak protocols and handle authentication;
  Basic auth, OAuth, XAuth, XMLRPC, ...
- Best feature? View-source!

Summary

- YQL is very useful for...
  - Scraping
  - Creating an API where one doesn't exist
  - Converting XML -> JSON, & vice-versa
  - JSONP for JS-only apps
  - Many HTTP requests -> single HTTP request
  - Server-side JS processing

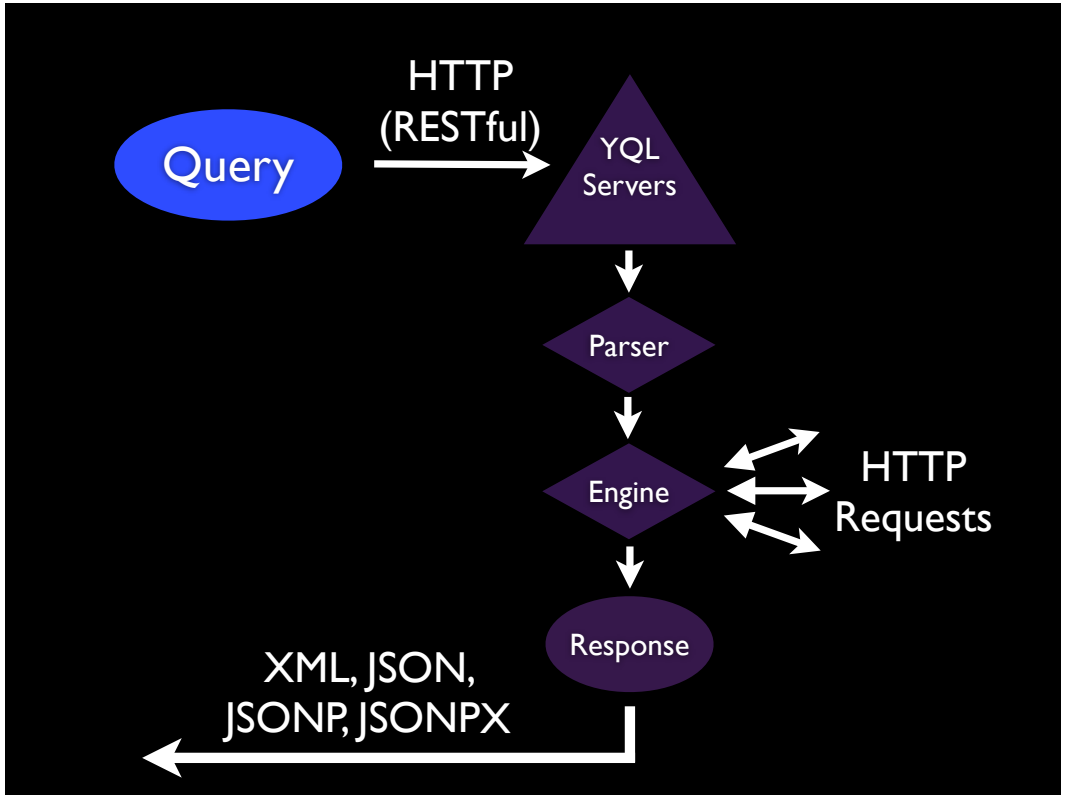
NY Times Data Tables

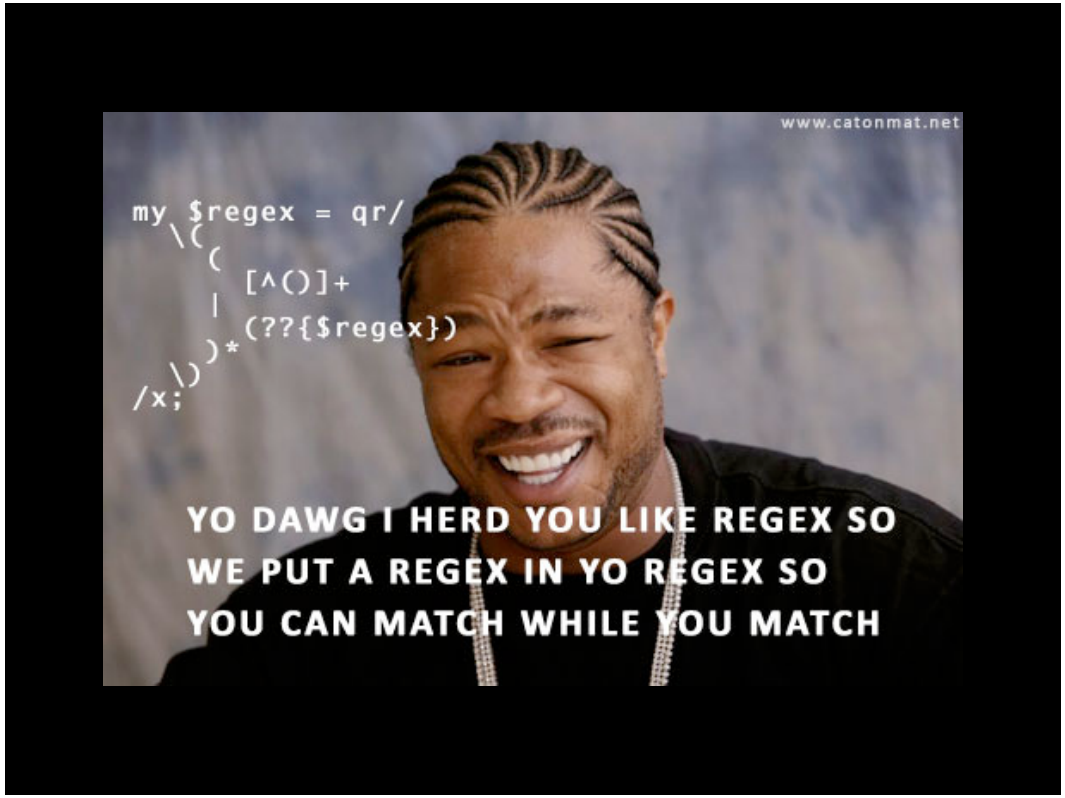
- nyt.article.search
- nyt.bestsellers
- nyt.bestsellers.history
- nyt.bestsellers.search
- nyt.movies.critics
- nyt.movies.picks
- nyt.movies.reviews
- nyt.newswire
- nyt.people.activities
- nyt.people.followers
- nyt.people.following
- nyt.people.newsfeed
- nyt.people.profiles
- nyt.people.users

Get started @
http://developer.yahoo.com/yql

Thanks!

Questions? Find, Tweet, or Email me.
@derek or drg@yahoo-inc.com


Query
HTTP
(RESTful)
YQL
Servers
Parser
Engine
HTTP
Requests
Response
XML, JSON,
JSONP, JSONPX


www.catonmat.net
my $regex = qr/
\(
(
[^()]+
|
(??{$regex})
)*
\)
/x;
YO DAWG I HERD YOU LIKE REGEX SO
WE PUT A REGEX IN YO REGEX SO
YOU CAN MATCH WHILE YOU MATCH

<blink>Warning!</blink>

The following presentation contains gratuitous animations, lolcats, <blink> tags, unicorns, and comic sans.

If you find any of the above obnoxious, please attend another talk.