Extracted from:

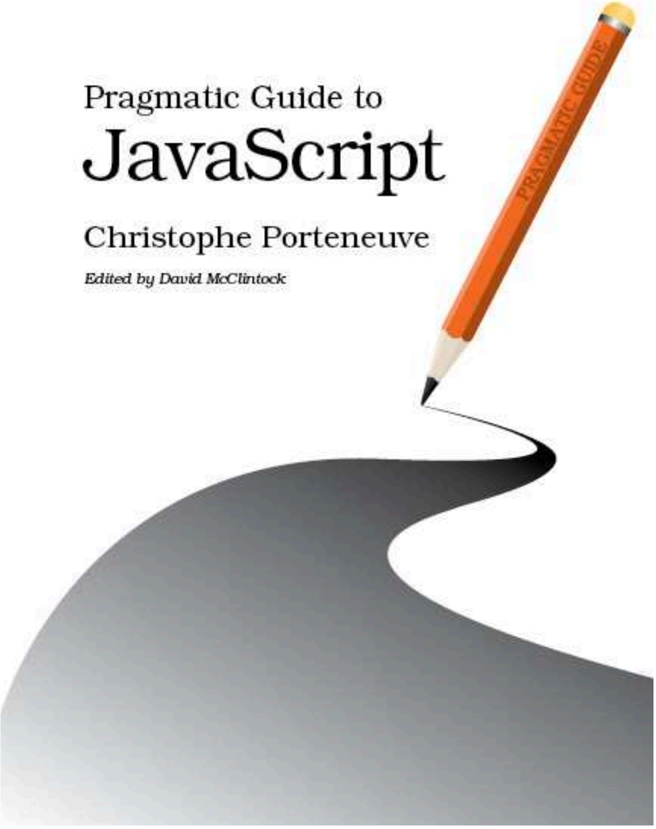
Pragmatic Guide to JavaScript

The Pragmatic Programmers

Pragmatic Guide to JavaScript

Christophe Porteneuve

Edited by David McClintock



Part VI

Making mashups

The final part of this book covers concrete examples of making your own web page, located on your own domain, talk with third-party services. There won't be any custom server proxy or XML in here, just JSON-P and YQL, so this code is self-contained and works anywhere!

However, you may find only the juicy bits in the code pages of this book. Sometimes a few finishing touches are required to wrap the demo. So, be sure to check these in the source code archive[28] or the live demo site[29] to see how it all fits together.

- Syndicating your Twitter feed on a web page is a common feature nowadays, and doing it on the client side can help when you otherwise heavily cache the hosting page. Task 33, *Syndicating Your Twitter Updates*, has you covered.
- Slapping your recent Flickr photo uploads in a page block somewhere is equally common and is explained in Task 34, *Syndicating Your Flickr Updates*.
- The third task, Task 35, *Geocoding a Location and Getting Photos For It*, explores an increasingly important aspect of the Web: geocoding (not to be confused with geolocation, which is also an important emerging trend but will only work on the latest, bleeding-edge browsers). Essentially, it's about turning place names and addresses into actual geographic coordinates so you can pinpoint them on a map, put them in relation with other data (say, photos from Flickr or tweets!), and create all sorts of useful mashups tying multiple datasets together. The Web and its data are more hackable today than they ever were, and there is enormous potential in what we can do with this!

[28] http://pragprog.com/titles/pg_js
[29] http://demos.pocketjavascript.com/

33 Syndicating Your Twitter Updates

Fetching your recent tweets is a piece of cake, as is using most of Twitter's API. It's just a simple JSON-P call, really!

We won't fetch retweets, mentions, or whatnot here. This is because most Twitter syndication happens in a business context, where your Twitter account is used as an extra marketing channel and you're not interested much in showing retweets, replies, or mentions alongside your own messages. Moreover, the Twitter API does not provide a straightforward way to grab, say, retweets alongside tweets—you'd need two separate calls, one of them authenticated, which means that authentication *to your own account* would be available on the client side *in your visitors' browsers*. And we don't want that. (The other option is syndication on the server side, which falls outside the scope of this book.)

In the code on the facing page, the actual fetching takes only a couple of lines, in the loadTwitterStream( ) method. Twitter gives you read access to any username's direct tweets through a URL,[30] and we're interested in a JSON format here.

What you get back is an array of tweet objects with a wealth of properties, such as created_at, geo, in_reply_to_status_id, source, and text.[31]

The twitterCallback( ) function on the facing page illustrates simple tweet formatting: linking up a reply mention and URLs in general. In the online example code for this task, you'll get an augmented version that also handles hash tags and mentions and shows author information (avatar, name, tweet count, and so on).

Before you do too much with Twitter's API, you should go through Twitter's nice API documentation[32] and also be careful about rate limits enforced on parts of the API to avoid overtaxing the system as your usage scales up.

[30] http://twitter.com/statuses/user_timeline/username.format

[31] You'll get full details at http://apiwiki.twitter.com/Return-Values.

[32] http://apiwiki.twitter.com/

- Fetch your recent tweets.

The following code relies on Prototype for a few things ($( ), each( ), escapeHTML( ), insert( )...) but is readily convertible to other frameworks.

Download mashups/twitter/twitter.js

```
var REGEXP_URL = new RegExp('(https?://.*?)(\\W?(?:\\s|$))', 'gi');

function twitterCallback(data) {
  var stream = $('twitterStream'), replyTo, contents;
  data.each(function(tweet) {
    contents = tweet.text.escapeHTML().replace(REGEXP_URL,
      '<a href="$1">$1</a>$2');
    if (replyTo = tweet.in_reply_to_screen_name) { // Intentional assign
      contents = contents.replace('@' + replyTo,
        '<a href="http://twitter.com/' + replyTo + '/statuses/' +
        tweet.in_reply_to_status_id + '">$&</a>');
    }
    contents = '<li><p>' + contents + '</p>' +
      '<p class="stamp">' + tweet.created_at + '</p></li>';
    stream.insert(contents);
  });
}

function loadTwitterStream(userName) {
  var uri = 'http://twitter.com/statuses/user_timeline/'+userName+'.json';
  document.documentElement.firstChild.appendChild(
    new Element('script', { type: 'text/javascript',
      src: uri + '?callback=twitterCallback&r=' + Math.random() }));
}
```

- Check out part of a JSON-encoded tweet.

(Actual returned datasets are far more detailed, and URLs are obviously not truncated—this is just to give you an idea.)

```
{
  "in_reply_to_screen_name": null,
  "user": {
    "friends_count": 27, "statuses_count": 622,
    "name": "ChristophePorteneuve",
    "followers_count": 215,
    "profile_image_url": "http://a3.twimg.com/.../headshot_tdd_normal.jpg",
  },
  "id": 9537162839, "created_at": "Tue Feb 23 18:35:22 +0000 2010",
  "in_reply_to_status_id": null,
  "text": "15' pour 850m. Sympa av Saint-Ouen + av Clichy aux heures de..."
}
```

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Pragmatic Guide to JavaScript

http://pragprog.com/titles/pg_js
Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

http://pragprog.com/updates
Be notified when updates and new books become available.

Join the Community

http://pragprog.com/community
Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

http://pragprog.com/news
Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/pg_js.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	support@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com
Contact us:	1-800-699-PROG (+1 919 847 3884)