

Web storage: an alternative to PHP for managing Twitter data

Twitter, just like many other consuming web services, is unreliable. The point is that the Twitter's API may be unavailable, thus making our remote calls fail. When it comes to display my tweets or search results taken from Twitter, I generally prefer to use jQuery and JavaScript for this particular task. I've used PHP and the WordPress built-in functions for a long time to do the same thing, but eventually I came to the conclusion that what I want to achieve is only adding some additional information. Since these info are not relevant to the full comprehension of my web sites, why bother?

This is not an holy war between JavaScript and PHP, but a simple consideration about the primary and secondary goals of our sites. We maintain our web sites to speak freely about the things that are really important to us. Sure, we also tweet our thoughts, but we primarily write posts and pages.

Using PHP and WordPress just to display my latest tweet is like to kill a fly with a bazooka. Further, using this approach involves caching, timeouts, transient APIs, error handling and logging, all on the server-side.

JavaScript is less expensive in terms of time spent to write an error-free code that always works. Error handling is trivial:

```
try {  
    $.getJSON(url, function(data) {  
  
        // fetch and display  
  
    });  
} catch(e) {
```

```
        // Error handling  
  
    }
```

What about caching tweets? I didn't test this feature yet, but I think web storage could be a feasible option:

```
var localTweet = localStorage.getItem('tweet'); //  
string  
  
if(localTweet == '') {  
  
    // fetch, display and store the tweet  
  
} else {  
  
    // display the tweet  
  
}
```

With local storage it's simply a matter of parsing and displaying a string as normal HTML content. If users surf through our pages, they will get their copy of the original tweet.

Everything will be handled by the browser. We won't need to get mad about Twitter downtime periods or, more frankly, we'll feel a little bit lighter by knowing that our logs and pages will be without those frightening PHP errors.