AJAX Before Time

Adrian Kosmaczewski

2021-02-05

In 2002 I moved back to Switzerland, and found a first job as a developer in a company making a super expensive product nobody needed. It tanked one year later for well known reasons, but in the meantime I got to see the most complex codebase I had ever met, in this case in JScript.

It was humongous; we are talking here about an ASP application in the hundreds of thousand lines of JScript code, in impeccable object-oriented style, with all the Gang of Four design patterns showcased in an extreme example of good architecture. Do not get me wrong; for I am not being ironic here. It was a wonderful team with kind and brilliant people, and I learnt a lot during my year with them. The web application, however, took a sensible time to startup and to run; it had serious performance issues. Its complexity had a huge cost.

But the most fascinating thing about this codebase what that it had a hidden secret deep inside. See, the app *would update itself without reloading the page*. This was 2002. I had never seen anything like this before. The first time I noticed it I thought my eyes had skipped a frame or two, but then I realized that every so often the UI was changing; things were appearing and disappearing, text and images would change dynamically. I first thought they were JavaScript animations – like the ones you could create with Macromedia Dreamweaver back then, with long chunks of JavaScript separated with rather horrendous `if (IE) {` blocks.

But no, this something else. This was actual data being reloaded and reshuffled and the whole thing was alive. Then I thought that they might be using a hidden `<IFRAME>` to get the trick; so I grabbed my beloved EditPlus and I started digging in the codebase.

What I found was astonishing. There was a bit of code at the far end of the call stack, with the following line:

```
var request = new ActiveXObject("Microsoft.XMLHTTP");
```

That little object had event handlers, and those handlers were updating the user interface accordingly, every time that the backend had changed its state for whatever reason.

I saw AJAX in action for the first time in 2002, three years before Jesse James Garrett coined the name in an article in his blog.

By the time Mr. Garrett wrote his article, Safari, Firefox and Opera had also added a component with pretty much the same interface, which allowed developers to write code that looked like this:

```
if (window.XMLHttpRequest) {
    //Firefox, Opera, IE7, and other browsers will use the native object
    var request = new XMLHttpRequest();
} else {
    //IE 5 and 6 will use the ActiveX control
    var request = new ActiveXObject("Microsoft.XMLHTTP");
}
```

A few years later, Prototype.js and jQuery were born to hide this `if/else` statement inside of a library, among other functionalities. The rest is history, and it was called “Web 2.0”, whatever that meant.