

# Fixing Internet Explorer & AJAX



Lori MacVittie, 2008-26-06

A few weeks ago, as developers are wont to do, I rewrote our online gameroom. Version 1 was getting crusty, and I'd written all the AJAX handlers manually and wanted to clean up the code by using [Prototype](#) and [Script.aculo.us](#). You may recall we discussed using these tools [to build a Web 2.0 interface](#) to [iControl](#).

So I rewrote it and was pretty pleased with myself. Until one of our players asked why it wasn't working in Internet Explorer (IE). Now Version 1 hadn't worked in IE either, but because I have a captive set of users I ignored the problem and forced them all to use [FireFox](#) instead. But this player's wife will be joining us soon and she's legally blind. She uses a reader to get around the Internet and as luck would have it, the reader only works with IE.

So I started digging into the problem. I had thought it was *my* code (silly me), and thus moving to prototype would solve the problem. No such luck. Everything but the periodically updated pieces of the application worked fine. The real-time updating components? Broken in IE.

I looked around and found [this very interesting article](#) on [Wikipedia](#) regarding known problems with IE and [XMLHttpRequest](#), the core of AJAX.

From the Wikipedia article

Most of the implementations also realize HTTP caching. Internet Explorer and Firefox do, but there is a difference in how and when the cached data is revalidated. Firefox revalidates the cached response every time the page is refreshed, issuing an "If-Modified-Since" header with value set to the value of the "Last-Modified" header of the cached response.

Internet Explorer does so only if the cached response is expired (i.e., after the date of received "Expires" header).

Basically, the problem lies with IE's caching mechanisms. So if you were trying to build an AJAX application with a real-time updating component and it didn't seem to work in IE, now you may know why that is.

There are workarounds:

1. Modify the AJAX call (within the client-side script) to check the response and, if necessary, make a second call with a Date value in the past to force the call to the server.
2. Append a unique query string to the call, for example appending a timestamp. This makes the URI unique, ensuring it won't be in the cache and forcing IE to call out to the server to get it.
3. Change all requests to use POST instead of GET.
4. Force the "Expires" header to be set in the past (much in the way we expire cookies programmatically). Setting cache control headers may also help force IE to act according to expectations.

I used option #3, because it was a simple, quick fix for me to search the single script using [Ajax.PeriodicalUpdater](#) and automatically change all the GETs to POSTs. That may not be feasible for everyone, hence the other available options.

Option #4 could easily be achieved using [iRules](#), and could be coded such that only requests sent via IE were modified. In fact, [Joe](#) has a great [post on how to prevent caching on specific file types](#) that can be easily modified to solve the problem with IE.

First we want to know if the browser is IE, and if so, we want to modify the caching behavior on the response. Don't forget that [IE7 is using a slightly different User-Agent header](#) than previous versions of IE. Don't look for specific versions, just try to determine if the browser is a version of IE.

```
when HTTP_REQUEST {
  if {[string tolower [HTTP::header "User-Agent"]] contains "msie"} {
    set foundmatch 1
  }
}
```

```
}
```

```
when HTTP_RESPONSE {  
  if {$foundmatch == 1} {  
    HTTP::header replace Cache-Control no-cache  
    HTTP::header replace Pragma no-cache  
    HTTP::header replace Expires -1  
  }  
}
```

You could also use an iRule to accomplish #3 dynamically, changing the code *only* for IE browsers instead of all browsers. This requires a bit more work as you'll have to search through the payload for 'GET' and replace it with 'POST'. It's a good idea to make the search string as specific as possible to ensure that only the HTTP methods are replaced in the [Ajax.PeriodicalUpdater](#) calls and not everywhere the letters may appear in the document, hence the inclusion of the quotes around the methods.

Happy Coding!



*Imbibing: Coffee*

---

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](#)

F5 Networks, Inc.  
Corporate Headquarters  
[info@f5.com](mailto:info@f5.com)

F5 Networks  
Asia-Pacific  
[apacinfo@f5.com](mailto:apacinfo@f5.com)

F5 Networks Ltd.  
Europe/Middle-East/Africa  
[emeainfo@f5.com](mailto:emeainfo@f5.com)

F5 Networks  
Japan K.K.  
[f5j-info@f5.com](mailto:f5j-info@f5.com)

---

©2016 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at [f5.com](#). Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113