

# Using Javascript inside a CMS

## What I set out to achieve

I was trying to add JSON, RSS and ATOM feeds from a few different online services to a page on our web site. I notice the Twitter API is quite thorough and has a great JSON function that returns a list of tweets. It would be great if the other services we are using had that, however there is always Atom (preferable) or RSS.

## Feedburner

Up until now we have been using feedburner to create and monitor our feeds. It gives great statistics. It is as simple as creating a feedburner account, logging in and pasting your subscription url into the feedburner site (can create many feeds using your account).

## Content Management System

We are using a system that allows you create textual content using either Word or Open Office. For this page containing feeds we broke the page up using a table and then used the embedding feature that this CMS has. This now allows us to create separate .htm files and imbed them into the Word or Open Office document. Why would you want to do this ? Lets say I need a form or a JSON feed that uses Javascript. I simply create a place holder on the page and link to the code like this “thecode.htm?embed”

## The Code

I am simply using the code offered by Google as part of their [feeds playground](#).

This is what we ended up with. A separate .htm file that references a .jsp file. The reason that we have a seperate htm and jsp file is that because the htm file is imbedded in the Word/Open Office file, the CMS has difficulty interpreting the symbols used in the Javascript eg <. This creates a problem when the htm file contains for loops etc using these symbols.

## HTML

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-
```

```

8"/>

<title>

</title>

<link href="/skin/style.css" rel="stylesheet" type="text/css" />

<link href="/skin/default.css" rel="stylesheet" type="text/css" />

</head>

<body>

<script type="text/javascript"
src="http://www.google.com/jsapi"></script>

    <script type="text/javascript"
src="javascript_file.js"></script>

<div id="blog_update_list" class="update_list" ></div>

</body>

</html>

```

## Javascript

After using primarily Javascript, I asked for some quick advice from the programmers sitting near me and we ended up with two solutions to the following jsp. One of the solutions use Javascript only and the other uses jQuery (the write less do more library). I had a bit of an issue with the fact that the jQuery solution looks to me like building strings as apposed to using what I would describe as a more object orientated or modular approach. However it is true there is a lot less code to the jQuery solution.

## Plain Javascript Solution

```

google.load("feeds", "1");

function initialize() {

    var feed = new
google.feeds.Feed("http://feeds.feedburner.com/myfeed");

    feed.load(function(result) {

        if (!result.error) {

```

```

        var container =
document.getElementById("blog_update_list");

        for (var i = 0; i < result.feed.entries.length; i++) {

            var entry = result.feed.entries[i];

            var attributes = ["title", "link", "publishedDate",
"contentSnippet"];

            var strTitle=entry[attributes[0]];

            var strLinkage=entry[attributes[1]];

            var strDate=entry[attributes[2]];

            var strContent=entry[attributes[3]];

            var elementList=document.createElement("li");

            var aLinkElement=document.createElement("a");

            var aLinkElementMore=document.createElement("a");

            var brElement=document.createElement("br");

            elementList.appendChild(document.createTextNode(strDate));

            elementList.appendChild(brElement);

            aLinkElement.innerHTML=strTitle;

            aLinkElement.href=strLinkage;

            aLinkElement.target="_blank";

            elementList.appendChild(aLinkElement);

            elementList.appendChild(brElement);

            elementList.appendChild(document.createTextNode(strContent));

            elementList.appendChild(document.createTextNode("..."));

            aLinkElementMore.innerHTML=" more";

            aLinkElementMore.href=strLinkage;

            aLinkElementMore.target="_blank";

            elementList.appendChild(aLinkElementMore);

            container.appendChild(elementList);

```

```

    }

    }

    });

}

google.setOnLoadCallback(initialize);

```

## jQuery Solution

```

google.load("feeds", "1");

function initialize() {

    var feed = new google.feeds.Feed("http://feeds.feedburner.com/
myfeed");

    feed.load(function(result) {

        if (!result.error) {

            for (var i = 0; i < result.feed.entries.length; i++) {

                var entry = result.feed.entries[i];

                var attributes = ["title", "link", "publishedDate",
"contentSnippet"];

                var strTitle=entry[attributes[0]];

                var strLinkage=entry[attributes[1]];

                var strDate=entry[attributes[2]];

                var strContent=entry[attributes[3]];

                var container = jq("#blog_update_list");

                var li = jq("<li>"+strDate+"<br/><a href='"+strLinkage+"'
target='_blank'>"+strTitle+"</a><br/>"+

                    strContent+"<a href='"+strLinkage+"'
target='_blank'> ...more</a></li>");

```

```
        container.append(li);  
  
    }  
}  
});  
}  
google.setOnLoadCallback(initialize);
```