ALLAN THRAEN | 15 years ago | PDF |

OUTPUT PAGEDATA AS JSON

JSON has gained a lot of popularity recently – and with good reason, in my opinion. It's fairly easy to work with, a lot more compact than XML and quite flexible. Especially javascript frameworks like JQuery has helped JSON win ground – and obviously the possibilities in working with JSON & JQuery has caught my eye as well as everybody else's.

Basically the cool thing about JSON is that it can express data as objects and lists in a compressed, yet understandable manner – and at the same time, it can be directly parsed by javascript and create js objects. This makes the line a bit blurry between code, configuration and data – but that's not always a bad thing. I'd guess that in the future we'll see more and more configuration becoming code (and not as microsoft is trying – to make coding into configuration). But that's an entirely different discussion.

I made a small prototype of a module that can export any pagedata object to JSON. Simply, copy the assembly to your bin folder, and append the following to the querystring of any EPiServer page: "json=[x]" where x is a string, that if it contains "current" will output the current page data as JSON, if it contains "children" will output the children to the current page.

Like this: *http://localhost/News/Technology/?json=current,children.*

Using this, it was fairly easy to make a dynamic content that using jquery did a JSON call-back, got the children of the active page and displayed them in a timed box:

```
<div id="ChildBox" style="width:200px;height:40px; background-color:Green; color:Yellow;"></div>
<script type="text/javascript">

    $.getJSON(window.location + "?json=children", function(data) {
        var children = data.children;
        var childidx = 0;

        setInterval(function() {
            $("#ChildBox").fadeOut("Slow", function() {
            childidx = childidx + 1;
            if (childidx >= children.length) childidx = 0;
            $(this).text($(children).get(childidx).PageName);
```

Simple, eh?

This usercontrol was then turned into Dynamic Content, using the DCPlugin – and of course the code-behind adds references to the JQuery library.

RECENT POSTS

CodeArt ApS
Teknikerbyen 5, 2830 Virum, Denmark
Email: info@codeart.dk
Phone: +45 26 13 66 96
CVR: 39680688