A simple AJAX website with jQuery

Martin Angelov September 7th, 2009

jQuery Trickshots is our new epic jQuery tips and tricks book. Check it out! [1]

Introduction

This time we are going to create a simple AJAX website with jQuery and the right amount of PHP & CSS. It is going to have a few pages loaded by AJAX from the PHP back-end, and a complete support of the browser history – a real pain for any AJAX or Flash site .

So **get the demo files** and lets start rollin'.

The XHTML

First, we create the XHTML backbone of the site.

demo.html

```
<div id="rounded">

<img src="img/top_bg.gif" /><!-- image with rounded left and right top corners -->
<div id="main" class="container"><!-- our main container element -->

<h1>A simple AJAX driven jQuery website</h1> <!-- titles -->
<h2>Because simpler is better</h2>

<ul id="navigation"> <!-- the navigation menu -->
<li><a href="#page1">Page 1</a></li> <!-- a few navigation buttons -->
<li><a href="#page2">Page 2</a></li>
<li><a href="#page3">Page 3</a></li>
<li><a href="#page4">Page 4</a></li>
<li><img id="loading" src="img/ajax_load.gif" alt="loading" /></li> <!-- rotating gif - hidden by default -->
</ul>

<div class="clear"></div> <!-- the above links are floated - we have to use the clearfix hack -->

<div id="pageContent"> <!-- this is where our AJAX-ed content goes -->
Hello, this is the default content
</div>

</div>
```

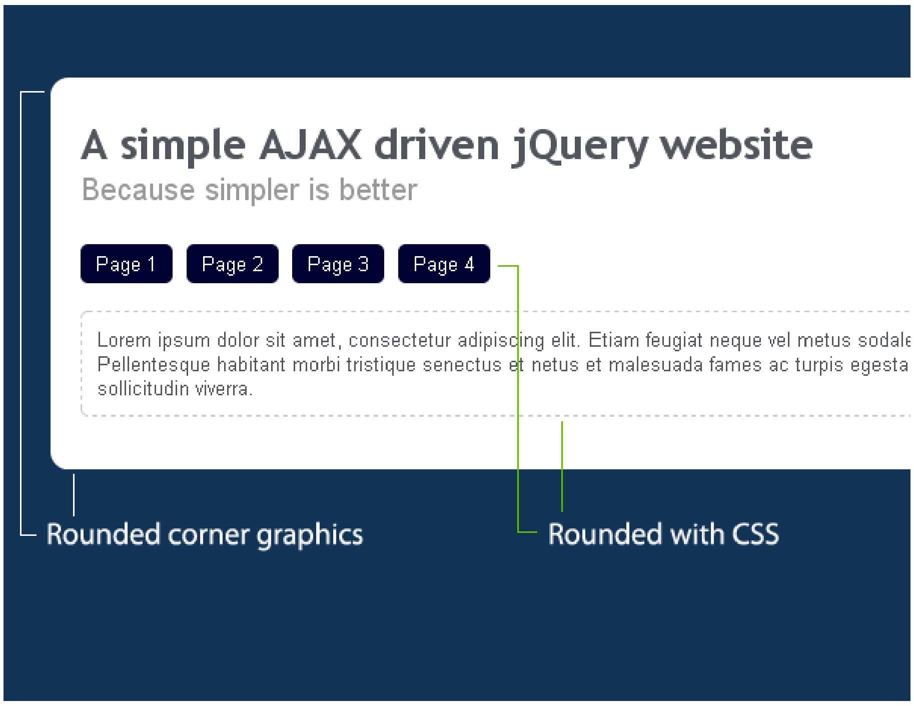
```
<div class="clear"></div> <!-- clearing just in case -->

<img src="img/bottom_bg.gif" /> <!-- the bottom two rounded corners of the page -->

</div>
```

This code is positioned in the **body** part of our **demo.html** file. Its main purpose is to serve as a front-end to the php back-end, with jQuery handling all the communication in between.

Note the addresses of the navigation links – **#page** and a page number. This part, called a **hash**, is included in the current URL without a page refresh, creating an entry in the browser's history. By monitoring this hash with javascript, we can change the loaded page by AJAX and provide a seamless browsing experience.


A simple AJAX driven jQuery website
Because simpler is better
Page 1
Page 2
Page 3
Page 4
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam feugiat neque vel metus sodale
Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egesta
sollicitudin viverra.
Rounded corner graphics
Rounded with CSS

Our simple AJAX enabled jQuery website

The CSS

Lets take a look at our style sheet.

demo.css

```
body,h1,h2,h3,p,td,quote,
small,form,input,ul,li,ol,label{      /* resetting our page elements */
    margin:0px;
    padding:0px;
    font-family:Arial, Helvetica, sans-serif;
}

body{     /* styling the body */
    margin-top:20px;
    color:#51555C;
    font-size:13px;
    background-color:#123456;
}

.clear{     /* the clearfix hack */
    clear:both;
}

a, a:visited {   /* styling the links */
    color:#007bc4;
    text-decoration:none;
    outline:none;
}

a:hover{  /* the hover effect */
    text-decoration:underline;
}

#rounded{      /* the outermost div element */
    width:800px;
    margin:20px auto;     /*center it on the page*/
}

.container{      /* this one contains our navigation, titles, and fetched content */
    background-color:#FFFFFF;
    padding:10px 20px 20px 20px;
}

h1{   /* the heading */
    font-size:28px;
    font-weight:bold;
    font-family:"Trebuchet MS",Arial, Helvetica, sans-serif;
}

h2{   /* the subheading */
    font-weight:normal;
    font-size:20px;

    color:#999999;
}

ul{    /* the unordered list used in the navigation */
    margin:30px 0px;
}

li{     /* we float the li-s, which contain our navigation links - we later apply clearfix */
    list-style:none;
    display:block;
    float:left;
    width:70px;
}

li a,li a:visited{      /* the navigation links */
    padding:5px 10px;
    text-align:center;
    background-color:#000033;
    color:white;

    -moz-border-radius:5px;   /* rounding them */
    -khtml-border-radius: 5px;
    -webkit-border-radius: 5px;
    border-radius:5px;

}

li a:hover{
    background-color:#666666;
    text-decoration:none;
}

#pageContent{      /* the container that holds our AJAX loaded content */
    margin-top:20px;

    border:1px dashed #cccccc;
    padding:10px;

    -moz-border-radius: 5px; /* rounding the element */
    -khtml-border-radius: 5px;
    -webkit-border-radius: 5px;
    border-radius: 5px;
}

#loading{       /* hiding the rotating gif graphic by default */
    visibility:hidden;
}
```

An important reminder would be to note that rounding corners with CSS is only supported in the latest versions of **Firefox**, **Safari** and **Chrome**.

The jQuery source

To complement the front-end, here is the script that drives the site.

script.js

```
$(document).ready(function(){     //executed after the page has loaded

    checkURL();      //check if the URL has a reference to a page and load it
    $('ul li a').click(function (e){      //traverse through all our navigation links..

            checkURL(this.hash);      //.. and assign them a new onclick event, using their own hash as a parameter (#page1 for example)

    });

    setInterval("checkURL()",250);      //check for a change in the URL every 250 ms to detect if the history buttons have been used

});

var lasturl=""; //here we store the current URL hash

function checkURL(hash)
{
    if(!hash) hash=window.location.hash;    //if no parameter is provided, use the hash value from the current address

    if(hash != lasturl)     // if the hash value has changed
    {
        lasturl=hash;   //update the current hash
        loadPage(hash);   // and load the new page
    }
}

function loadPage(url)  //the function that loads pages via AJAX
{
    url=url.replace('#page',''); //strip the #page part of the hash and leave only the page number

    $('#loading').css('visibility','visible');      //show the rotating gif animation

    $.ajax({      //create an ajax request to load_page.php
        type: "POST",
        url: "load_page.php",
        data: 'page='+url,  //with the page number as a parameter
        dataType: "html",  //expect html to be returned
        success: function(msg){

            if(parseInt(msg)!=0)  //if no errors
            {
                $('#pageContent').html(msg);     //load the returned html into pageContet
                $('#loading').css('visibility','hidden');   //and hide the rotating gif
            }
        }

    });

}
```

Note how, on line 3, we call the **checkURL()** function as soon as the page finishes loading – this way we insure that, if a link to an inner page on the site has been shared and a new visitor visits it, the site will fetch the required page and show it when the page is loaded.

As you can see on line 11, we setup an interval for **checkURL()** to check the browser's address 4 times a second in order to detect any possible changes arising from the use of the back/forward buttons.

Now lets take a look at the back-end.

The PHP

The PHP back-end is just a few lines of code and is the place to start, if you want to customize this example.

load_file.php

```
if(!$_POST['page']) die("0");

$page = (int)$_POST['page'];

if(file_exists('pages/page_'.$page.'.html'))
echo file_get_contents('pages/page_'.$page.'.html');

else echo 'There is no such page!';
```

It basically checks whether the variable **$_POST['page']** is set, and if it is, checks whether the respective **page_.html** file exists, and outputs it back to jQuery.

You can improve upon this by fetching data from a database, by using sessions, or displaying a folder of images – anything you might have on your mind.

Conclusion

Today we created a simple, customization-ready, AJAX enabled web site. Feel free to use the code and the techniques that were demonstrated in any of your projects.

by Martin Angelov

Martin is a web developer with an eye for design from Bulgaria. He founded Tutorialzine in 2009 and publishes new tutorials weekly.

Tutorials [2]

1. http://tutorialzine.com/books/jquery-trickshots/

2. http://tutorialzine.com/category/tutorials/