ASP.NET JavaScript Callbacks Without Full PostBacks?

Asked 16 years, 3 months ago Modified 2 years, 5 months ago

Viewed 5k times

6

I'm about to start a fairly Ajax heavy feature in my company's application. What I need to do is make an Ajax callback every few minutes a user has been on the page.

- I don't need to do any DOM updates before, after, or during the callbacks.
- I don't need any information from the page, just from a site cookie which should always be sent with requests anyway, and an ID value.

What I'm curious to find out, is if there is any clean and simple way to make a JavaScript Ajax callback to an ASP.NET page without posting back the rest of the information on the page. I'd like to not have to do this if it is possible.

I really just want to be able to call a single method on the page, nothing else.

Also, I'm restricted to ASP.NET 2.0 so I can't use any of the new 3.5 framework ASP AJAX features, although I can use the ASP AJAX extensions for the 2.0 framework.

UPDATE
I've decided to accept DanP's answer as it seems to be exactly what I'm looking for. Our site already uses jQuery for some things so I'll probably use jQuery for making requests since in my experience it seems to perform much better than ASP's AJAX framework does.

What do you think would be the best method of transferring data to the IHttpHandler? Should I add variables to the query string or POST the data I need to send?

The only thing I think I have to send is a single ID, but I can't decide what the best method is to send the ID and have the IHttpHandler handle it. I'd like to come up with a solution that would prevent a person with basic computer skills from accidentally or intentionally accessing the page directly or repeating requests. Is this possible?

asp.net javascript ajax

Share

Improve this question

Follow

edited May 23, 2017 at 12:13

Community Bot
1 ● 1

asked Aug 26, 2008 at 2:07

Dan Herbert
103k ● 51 ● 192 ● 221

9 Answers

Sorted by: Highest score (default)

6

If you don't want to create a blank page, you could call a IHttpHandler (ashx) file:

```
public class RSSHandler : IHttpHandler
    {
        public void ProcessRequest (HttpContext contex
        {
            context.Response.ContentType = "text/xml";

            string sXml = BuildXMLString(); //not show
            //but it creates the XML string
            context.Response.Write( sXml );
        }

        public bool IsReusable
        {
            get { return true; }
        }

    }
```

Share Improve this answer Follow

answered Aug 26, 2008 at 2:14

Daniel Pollard

1,230 ● 1 ● 12 ● 11

3

You should use ASP.Net Callbacks which were introduced in Asp.Net 2.0. Here is an article that should get you set to go:

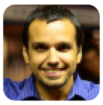
[Implementing Client Callbacks Programmatically Without Postbacks in ASP.NET Web Pages]()

Edit: Also look at this: [ICallback & JSON Based JavaScript Serialization]

Share Improve this answer Follow

edited Aug 26, 2008 at 3:08

answered Aug 26, 2008 at 2:36



Ricky
5,377 ● 2 ● 33 ● 30

▲
3
▼

> What do you think would be the best method of transferring data to the IHttpHandler? Should I added variables to the query string or POST the data I need to send? The only thing I think I have to send is a single ID, but I can't decide what the best method is to send the ID and have the IHttpHandler handle it. I'd like to come up with a solution that would prevent a person with basic computer skills from accidentally or intentionally accessing the page directly

Considering the callback is buried in the client code, it would take someone with equal determination to get either the querystring or the POST request. IE, if they have firebug, your equally screwed.

So, in that case, do whatever is easiest to you (Hint: I'd just use the querystring).

To handle repeating requests/direct access, I'd generate a key that is sent with each request. Perhaps a hash of the current time (Fuzzy, I'd go down to minutes, but not seconds due to network latency) + the client IP.

Then in the HTTPHandler, perform the same hash, and only run if they match.

Share Improve this answer Follow

answered Aug 26, 2008 at 3:21

FlySwat

175k ● 75 ● 248 ● 314

2

You are not just restricted to ASP.NET AJAX but can use any 3rd party library like jQuery, YUI etc to do the same thing. You can then just make a request to a blank page on your site which should return the headers that contain the cookies.

Share Improve this answer Follow

answered Aug 26, 2008 at 2:11

abigblackman

1,491 ● 1 ● 9 ● 4

2

My vote is with the HTTPHandler suggestion as well. I utilize this often. Because it does not invoke an instance of the page class, it is very lightweight.

All of the ASP.NET AJAX framework tricks actually instantiate and create the entire page again on the backend per call, so they are huge resource hogs.

Hence, my typical style of XmlHttpRequest back to a HttpHandler.

Share Improve this answer Follow

edited Aug 26, 2008 at 3:04

answered Aug 26, 2008 at 2:45

FlySwat

175k ● 75 ● 248 ● 314

1

Since you are using only ASP.NET 2.0 I would recommend AjaxPro will which create the .ashx file. All you have to do is to pull the AjaxPro.dll into your web site. I developed an entire application with AjaxPro and found it worked very well. It uses serialization to pass objects back and forth.

This is just a sample on how to simply use it.

```
namespace MyDemo
{
  public class Default
  {
    protected void Page_Load(object sender, EventArgs
    {
      AjaxPro.Utility.RegisterTypeForAjax(typeof(Defau
    }

    [AjaxPro.AjaxMethod]
    public DateTime GetServerTime()
    {
      return DateTime.Now;
    }
  }
}
```

To call it via JavaScript it is as simple as

```
function getServerTime()
{
  MyDemo._Default.GetServerTime(getServerTime_callback
}

// This method will be called after the method has bee
// and the result has been sent to the client.

function getServerTime_callback(res)
{
  alert(res.value);
}
```

EDIT

You also have to add

To the config. AjaxPro also works well side by side with APS.NET Ajax and you can pass C# objects from Client to Sever if the class is marked as [Serializable]

Share Improve this answer Follow

answered Aug 26, 2008 at 13:59

David Basarab

73.2k ● 43 ● 130 ● 157

▲

1

Just to offer a different perspective, you could also use a PageMethod on your main page. Dave Ward has a nice post that illustrates this. Essentially you use jQuery ajax post to call the method, as illustrated in Dave's post:

```
$.ajax({
    type: "POST",
    url: "Default.aspx/GetFeedburnerItems",
    // Pass the "Count" parameter, via JSON object.
    data: "{'Count':'7'}",
    contentType: "application/json; charset=utf-8",
    dataType: "json",
    success: function(msg) {
      BuildTable(msg.d);
    }
  });
```

No need for Asp.Net Ajax extensions at all.

Share Improve this answer Follow

answered Dec 24, 2009 at 13:40

David Robbins
10k ● 7 ● 54 ● 83

-1

You should use a web service (.asmx). With Microsoft's ASP.NET AJAX you can even auto-generate the stubs.

Share Improve this answer Follow

answered Aug 26, 2008 at 2:19

Jim
3,170 ● 4 ● 32 ● 38

▲

-1

▼

You can also use WebMethods which are built into the asp.net ajax library. You simply create a static method on the page's codebehind and call that from your Ajax.

There's a pretty basic example of how to do it [here]

Share Improve this answer

Follow

edited Jul 21, 2022 at 14:05

Glorfindel

22.6k ● 13 ● 89 ● 116

answered Aug 26, 2008 at 2:18

lomaxx

116k ● 58 ● 147 ● 180