

مزیت استفاده از یوزر کنترل‌ها، ماژولار کردن برنامه است. برای مثال اگر صفحه جاری شما قرار است از چهار قسمت اخبار، منوی پویا، سخن روز و آمار کاربران تشکیل شود، می‌توان هر کدام را توسط یک یوزر کنترل پیاده سازی کرده و سپس صفحه اصلی را از کنار هم قرار دادن این یوزر کنترل‌ها تهیه نمود.

با این توضیحات اکنون می‌خواهیم یک یوزر کنترل ASP.Net را توسط jQuery Ajax بارگذاری کرده و نمایش دهیم. حداقل دو مورد کاربرد را می‌توان برای آن متصور شد:

الف) در اولین باری که یک صفحه در حال بارگذاری است، قسمت‌های مختلف آن را بتوان از یوزر کنترل‌های مختلف خواند و تا زمان بارگذاری کامل هر کدام، یک عبارت لطفا منتظر بمانید را نمایش داد. نمونه‌ی آن را شاید در بعضی از CMS های جدید دیده باشید. صفحه به سرعت بارگذاری می‌شود. در حالیکه مشغول مرور صفحه جاری هستید، قسمت‌های مختلف صفحه پدیدار می‌شوند.

ب) بارگذاری یک قسمت دلخواه صفحه بر اساس درخواست کاربر. مثلا کلیک بر روی یک دکمه و امثال آن.

روش کلی کار:

- 1) تهیه یک متد وب سرویس که یوزر کنترل را بر روی سرور اجرا کرده و حاصل را تبدیل به یک رشته کند.
 - 2) استفاده از متد Ajax جی کوئری برای فراخوانی این متد وب سرویس و افزودن رشته دریافت شده به صفحه.
- بدیهی است زمانیکه متد Ajax فراخوانی می‌شود می‌توان عبارت یا تصویر منتظر بمانید را نمایش داد و پس از پایان کار این متد، عبارت (یا تصویر) را مخفی نمود.

پیاده سازی:

قسمت تبدیل یک یوزر کنترل به رشته را قبلا در مقاله "[تهیه قالب برای ایمیل‌های ارسالی یک برنامه ASP.Net](#)" مشاهده کرده‌اید. در اینجا برای استفاده از این متد در یک وب سرویس نیاز به کمی تغییر وجود داشت (KeyValuePair ها درست سریالایز نمی‌شوند) که نتیجه نهایی به صورت زیر است. یک فایل Ajax.aspx را به برنامه اضافه کرده و سپس در صفحه Ajax.aspx.cs کد آن به صورت زیر می‌تواند باشد:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Reflection;
using System.Text.RegularExpressions;
using System.Web;
using System.Web.Script.Services;
using System.Web.Services;
using System.Web.UI;
using System.Web.UI.HtmlControls;

namespace AjaxTest
{
    public class KeyVal
    {
        public string Key { set; get; }
        public object Value { set; get; }
    }

    /// <summary>
    /// Summary description for Ajax
    /// </summary>
    [ScriptService]
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    public class Ajax : WebService
    {
        /// <summary>
        /// Removes Form tags using Regular Expression
    }
```

```

/// </summary>
private static string cleanHtml(string html)
{
    return Regex.Replace(html, @"<[/]?(form)[^>]*?>", string.Empty, RegexOptions.IgnoreCase);
}

/// <summary>
/// تبدیل یک یوزر کنترل به معادل اچ تی ام ال آن
/// </summary>
/// <param name="path">مسیر یوزر کنترل</param>
/// <param name="properties">لیست خواص به همراه مقادیر مورد نظر</param>
/// <returns></returns>
/// <exception cref="NotImplementedException"><c>NotImplementedException</c>.</exception>
[WebMethod(EnableSession = true)]
[ScriptMethod(ResponseFormat = ResponseFormat.Json)]
public string RenderUserControl(string path,
    List<KeyVal> properties)
{
    Page pageHolder = new Page();

    UserControl viewControl =
        (UserControl)pageHolder.LoadControl(path);

    viewControl.EnableViewState = false;

    Type viewControlType = viewControl.GetType();

    if (properties != null)
        foreach (var pair in properties)
        {
            if (pair.Key != null)
            {
                PropertyInfo property =
                    viewControlType.GetProperty(pair.Key);

                if (property != null)
                {
                    if (pair.Value != null) property.SetValue(viewControl, pair.Value, null);
                }
                else
                {
                    throw new NotImplementedException(string.Format(
                        "UserControl: {0} does not have a",
                        path, pair.Key));
                }
            }
        }

    //Form control is mandatory on page control to process User Controls
    HtmlForm form = new HtmlForm();

    //Add user control to the form
    form.Controls.Add(viewControl);

    //Add form to the page
    pageHolder.Controls.Add(form);

    //Write the control Html to text writer
    StringWriter textWriter = new StringWriter();

    //execute page on server
    HttpContext.Current.Server.Execute(pageHolder, textWriter, false);

    // Clean up code and return html
    return cleanHtml(textWriter.ToString());
}
}
}

```

تا این جا متد وب سرویسی را داریم که می تواند مسیر یک یوزر کنترل را به همراه خواص عمومی آن را دریافت کرده و سپس یوزر کنترل را رندر نموده و حاصل را به صورت HTML به شما تحویل دهد. با استفاده از reflection خواص عمومی یوزر کنترل یافت شده و مقادیر لازم به آنها پاس می شوند.

چند نکته:

الف) وب کانفیگ برنامه ASP.Net شما اگر با VS 2008 ایجاد شده باشد مداخل لازم را برای استفاده از این وب سرویس توسط jQuery Ajax دارد در غیر اینصورت موفق به استفاده از آن نخواهید شد.

ب) هنگام بازگرداندن این اطلاعات با فرمت json = ResponseFormat.Json جهت استفاده در jQuery Ajax ، گاهی از اوقات بسته به حجم بازگردانده شده ممکن است خطایی حاصل شده و عملیات متوقف شد. این طول پیش فرض را (maxJsonLength) در وب کانفیگ به صورت زیر تنظیم کنید تا مشکل حل شود:

```
<system.web.extensions>
  <scripting>
    <webServices>
      <jsonSerialization maxJsonLength="10000000"></jsonSerialization>
    </webServices>
  </scripting>
</system.web.extensions>
```

برای پیاده سازی قسمت Ajax آن برای اینکه کار کمی تمیزتر و با قابلیت استفاده مجدد شود یک پلاگین تهیه شده (فایلی با نام jquery.advloaduc.js) که سورس آن به صورت زیر است:

```
$.fn.advloaduc = function(options) {
  var defaults = {
    webServiceName: 'Ajax.asmx', // نام فایل وب سرویس ما
    renderUCMethod: 'RenderUserControl', // متد وب سرویس
    ucMethodJsonParams: '{path: \'\'}', // قرار است یاس شوند
    completeHandler: null // متد جاوا اسکریپتی فراخوانی می شود
  };
  var options = $.extend(defaults, options);

  return this.each(function() {
    var obj = $(this);
    obj.prepend("<div align='center'>لطفا اندکی تامل بفرمائید <img src='\"images/loading.gif\"'/></div>");

    $.ajax({
      type: "POST",
      url: options.webServiceName + "/" + options.renderUCMethod,
      data: options.ucMethodJsonParams,
      contentType: "application/json; charset=utf-8",
      dataType: "json",
      success: function(msg) {
        obj.html(msg.d);

        // if specified make callback and pass element
        if (options.completeHandler)
          options.completeHandler(this);
      },
      error: function(XMLHttpRequest, textStatus, errorThrown) {
        obj.html("امکان اتصال به سرور در این لحظه مقدور نیست. لطفا مجددا سعی کنید");
      }
    });
  });
};
```

برای اینکه با کلیات این روش آشنا شوید می توان به مقاله "[بررسی وجود نام کاربر با استفاده از jQuery Ajax در ASP.Net](#)" مراجعه نمود که از ذکر مجدد آن ها خودداری می شود. همچنین در مورد نوشتن یک پلاگین جی کوئری در مقاله "[افزونه جملات jQuery](#)" توضیحاتی داده شده است.

عمده کاری که در این پلاگین صورت می گیرد فراخوانی متد Ajax جی کوئری است. سپس به متد وب سرویس ما (که در اینجا نام آن به صورت پارامتر نیز قابل دریافت است)، پارامترهای لازم پاس شده و سپس نتیجه حاصل به یک شیء در صفحه اضافه می شود.

completeHandler آن اختیاری است و پس از پایان کار متد اجکس فراخوانی می شود. در صورتیکه به آن نیازی نداشتید یا مقدار آن را null قرار دهید یا اصلا آن را ذکر نکنید.

مثالی در مورد استفاده از این وب سرویس و همچنین پلاگین جی کوئری نوشته شده:

الف) یوزر کنترل ساده زیر را به پروژه اضافه کنید:

```
<%@ Control Language="C#" AutoEventWireup="true" CodeBehind="part1.ascx.cs"
Inherits="TestjQueryAjax.part1" %>
<asp:Label runat="server" ID="lblData" ></asp:Label>
```

بدیهی است یک یوزر کنترل می‌تواند به اندازه یک صفحه کامل پیچیده باشد به همراه انواع و اقسام ارتباطات با دیتابیس و غیره.

سپس کد آن را به صورت زیر تغییر دهید:

```
using System;
using System.Threading;

namespace TestjQueryAjax
{
    public partial class part1 : System.Web.UI.UserControl
    {
        public string Text1 { set; get; }
        public string Text2 { set; get; }

        protected void Page_Load(object sender, EventArgs e)
        {
            Thread.Sleep(3000);
            if (!string.IsNullOrEmpty(Text1) && !string.IsNullOrEmpty(Text2))
                lblData.Text = Text1 + "<br/>" + Text2;
        }
    }
}
```

این یوزر کنترل دو خاصیت عمومی دارد که توسط وب سرویس مقدار دهی خواهد شد و نهایتاً حاصل نهایی را در یک لیبل در دو سطر نمایش می‌دهد.

عمداً یک sleep سه ثانیه‌ای در اینجا در نظر گرفته شده تا اثر آن را بهتر بتوان مشاهده کرد.

ب) اکنون کد مربوط به صفحه‌ای که قرار است این یوزر کنترل را به صورت غیرهمزمان بارگذاری کند به صورت زیر خواهد بود (مهم‌ترین قسمت آن نحوه تشکیل پارامترها و مقدار دهی خواص یوزر کنترل است):

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="TestjQueryAjax._Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>

    <script src="js/jquery.js" type="text/javascript"></script>
    <script src="js/jquery.advloaduc.js" type="text/javascript"></script>
    <script src="js/json2.js" type="text/javascript"></script>

    <script type="text/javascript">
        function showAlert() {
            alert('finished!');
        }

        //تشکیل پارامترهای متد وب سرویس جهت ارسال به آن
        var fileName = 'part1.ascx';
        var props = [{ 'Key': 'Text1', 'Value': 'سطر یک' }, { 'Key': 'Text2', 'Value': '2 سطر' }];
        var jsonText = JSON.stringify({ path: fileName, properties: props });

        $(document).ready(function() {
            $("#loadMyUc").advloaduc({
                webServiceName: 'Ajax.asmx',
                renderUCMethod: 'RenderUserControl',
                ucMethodJsonParams: jsonText,
                completeHandler: showAlert
            });
        });
    </script>
```

```
</script>
</head>
<body>
  <form id="form1" runat="server">
    <div id="loadMyUc">
    </div>
  </form>
</body>
</html>
```

نکته:

برای ارسال صحیح و امن اطلاعات json به سرور، از اسکریپت استاندارد [json2.js](#) استفاده شد.

ابزار دیباگ:

بهترین ابزار برای دیباگ این نوع اسکریپت‌ها استفاده از افزونه فایرباگ فایرفاکس است. برای مثال مطابق تصویر زیر، یوزر کنترلی فراخوانی شده است که در سرور وجود ندارد:

The screenshot shows the Firefox Developer Tools interface with the 'Net' panel selected. The 'Response' tab is active for the failed POST request. The error message is: "The file \u0027monthcalendar.ascx\u0027 does not exist." The stack trace indicates the error occurred in the System.Web.UI.Util.CheckVirtualFileExists method, which was called by the TestJQueryAjax.Ajax.RenderUserController.

Method	Status	Local Address	Size
GET localhost:25413	200 OK	localhost:25413	1 KB
GET jquery.js	200 OK	localhost:25413	54 KB
GET jquery.advloaduc.js	200 OK	localhost:25413	1 KB
GET json2.js	200 OK	localhost:25413	16 KB
GET loading.gif	200 OK	localhost:25413	2 KB
POST RenderUserController	500 Internal Server Error	localhost:25413	1 KB

Response:

```
{
  "Message": "The file \u0027monthcalendar.ascx\u0027 does not exist.",
  "StackTrace": "
    at System.Web.UI.Util.CheckVirtualFileExists(VirtualPath virtualPath)\r\n
    at System.Web.Compilation.BuildManager.GetVPathBuildResultInternal(VirtualPath virtualPath, Boolean noBuild, Boolean allowCrossApp, Boolean allowBuildInPrecompile)\r\n
    at System.Web.Compilation.BuildManager.GetVPathBuildResultWithNoAssert(HttpContext context, VirtualPath virtualPath, Boolean noBuild, Boolean allowCrossApp, Boolean allowBuildInPrecompile)\r\n
    at System.Web.Compilation.BuildManager.GetVPathBuildResult(HttpContext context, VirtualPath virtualPath, Boolean noBuild, Boolean allowCrossApp, Boolean allowBuildInPrecompile)\r\n
    at System.Web.UI.TemplateControl.LoadControl(VirtualPath virtualPath)\r\n
    at System.Web.UI.TemplateControl.LoadControl(String virtualPath)\r\n
    at TestJQueryAjax.Ajax.RenderUserController(String path, List`1 properties) in I:\asp_net_works\wwwroot\1388\AjaxTemplate\TestJQueryAjax\TestJQueryAjax\Ajax.aspx.cs:line 49",
  "ExceptionType": "System.Web.HttpException"
}
```

6 requests 77 KB

[دریافت مثال فوق](#)

نظرات خوانندگان

نویسنده: نیما

تاریخ: ۱۳۸۸/۰۴/۱۵ ۲۲:۱۳:۴۲

سلام استاد نصیری

بسیار مقاله مفید و Trick جالبی بود. به تازگی افزونه ای برای فایر باگ عرضه شده به اسم Fire Query که مواردی را برای کار و دیباگ jQuery فراهم میکند.

یه سوال از حضورتون داشتم چرا خروجی Execute در وب سرویس شامل تگ های html و body نیست؟ آیا این همیشه ثابت است؟ و اینکه مواردی مثل \n و \r رو هم از خروجی حذف کنیم بهتر نیست؟
باسپاس از شما استاد عزیز

نویسنده: Anonymous

تاریخ: ۱۳۸۸/۰۴/۱۵ ۲۲:۳۷:۳۶

سلام آقای نصیری

امکانش هست در مورد متد stringify و عملی که در اینجا انجام میدی کمی توضیح بدی؟ من اون لینک صفحه JSON رو خوندم ولی متوجه نشدم چرا شما اینجا از این متد استفاده کردید.
ممنون

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۴/۱۵ ۲۳:۱۰:۱۶

@نیما

بله. یک trim را هم می توانید اضافه کنید بعلاوه نکته حذف فاصله خالی بین تگ ها که در کاهش حجم نهایی مؤثر است.

@ناشناس

می شد این آرگومان های مورد نظر رو به شکل یک رشته یک سطر هم در آورد (فرقی نمی کرد). اما با استفاده از این تابع شما میتونید یک شیء جاوا اسکریپتی را تبدیل به json کنید که از این لحاظ خوب یک قدم پیشرفت است و در خطایابی کدها کمک بزرگی است (رشته قابل خطایابی نیست).

نویسنده: مهدی

تاریخ: ۱۳۸۸/۰۵/۰۹ ۱۷:۴۵:۰۹

ایده خیلی جالبی بود. اما چیزی که الان فکر من رو مشغول کرده، اینه که چطور ممکنه که از postback های این کنترل استفاده کرد.

اگر خودتون در این مورد کار کردین هم یک توضیحی بدین. اگر هم کاری نکردین که من خودم اگر وقت کردم یک تستی میکنم و اگر جواب داد به شما هم خبر میدم. نباید کار سختی باشه ولی بازم تا شروع نکنم نمیتونم قطعی نظر بدم.

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۵/۰۹ ۱۹:۰۸:۲۳

اگر خطای viewstate is corrupted را دریافت می کنید به این علت است که الان در صفحه دو فیلد مخفی viewstate داریم. برای حذف این مورد دوم سطر زیر را به تابع cleanHtml اضافه کنید:

```
;(html = Regex.Replace(html, "<input.*name=\"__VIEWSTATE\"[^>]*>", string.Empty, RegexOptions.IgnoreCase
```

+ در این حالت چون اطلاعات کنترل ها به ViewState اضافه نمی شوند، ASP.Net Postback هم کار نمی کنه.

در این حالت برای ارسال دیتا و غیره امکان استفاده از وب سرویس و پاس کردن مقدار به آن هست (با استفاده از جی کوئری اجکس).

کمی بیشتر کار میبهره نسبت به MS Ajax.

نویسنده: sirasad
تاریخ: ۱۳۸۸/۰۵/۱۴ ۱۲:۳۲:۰۴

آقای نصیری عزیز من خواستم این کد رو در net 2.0 تست بکنم , مقادیر اون تکست ها منتقل نمی شود ولی showalert رو فایرآپ میکنه .
یه مسئله دیگه اینکه این ajax شما بدون پارامتر ورودی کار میکنه ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۵/۱۴ ۱۳:۳۳:۳۷

بله. اگر پارامتر ورودی نداشت یک '{}' قرار دهید کفایت میکنه.
برای دات نت 2 کمی کار شما بیشتر میشود چون در VS2008 خیلی از موارد را خودکار اضافه میکنه.
در این حالت یا باید ASP.NET AJAX Extensions روی سرور نصب باشد یا dll های آنرا به صورت دستی به دایرکتوری bin برنامه اضافه کنید و یک ارجاع از آنها را نیز به برنامه اضافه کنید.
همچنین وب کانفیگ را نیز باید ویرایش کنید تا این تغییر در آن لحاظ شود. (در غیر اینصورت کار نمیکند)
یک نکته دیگر هم اینکه بجای msg.d شما فقط msg را در asp.net 2 دارید و این d از 3.5 به بعد اضافه شده (در قسمت success و هنگام دریافت پاسخ از سرور توسط jQuery).

نویسنده: javad.mjt
تاریخ: ۱۳۸۹/۱۰/۲۱ ۱۵:۴۵:۳۸

سلام آقای نصیری
زمانی که یک دکمه یا کنترلی که postback ایجاد می کند را استفاده می کنم به خطای زیر بر می خورم
The state information is invalid for this page and might be corrupted.
Description: An unhandled exception occurred during the execution of the current web request. Please review the
stack trace for more information about the error and where it originated in the code

Exception Details: System.Web.HttpException: The state information is invalid for this page and might be
corrupted.
لطفا راهنمایی کنید با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۰/۲۱ ۱۷:۱۵:۰۲

لطفا کامنتها را مطالعه کنید در بالا توضیح دادم (جولای ۳۱, ۲۰۰۹, ساعت ۲۰:۰۸).
1- باید ViewState مربوط به این یوزر کنترل را خاموش کنید.
2- خاموش کردن ViewState سبب حذف فیلد مخفی آن به صورت کامل نمیشود, علاوه بر آن باید فیلد مخفی ViewState موجود در رشته را با استفاده از Regex ایی که در بالا توضیح دادم کلا حذف کنید تا با ViewState اصلی صفحه تداخل نکند.