

با استفاده از AutoComplete TextBoxes می‌توان گوشه‌ای از زندگی روزمره‌ی کاربران یک برنامه را ساده‌تر کرد. مشکل مهم dropDownList ها در یک برنامه‌ی وب، عدم امکان تایپ قسمتی از متن مورد نظر و سپس نمایان شدن آیتم‌های متناظر با آن در اسرع وقت می‌باشد. همچنین با تعداد بالای آیتم‌ها هم حجم صفحه و زمان بارگذاری را افزایش می‌دهند. راه‌های بسیار زیادی برای حل این مشکل وجود دارند و یکی از آن‌ها ایجاد AutoComplete TextBoxes است. پلاگین‌های متعددی هم جهت پیاده‌سازی این قابلیت نوشته شده‌اند منجمله [jQuery Autocomplete](#). این پلاگین دیگر توسط نویسنده‌ی اصلی آن نگهداری نمی‌شود اما توسط برنامه نویسی دیگر در [github](#) ادامه یافته است. در ادامه نحوه‌ی استفاده از این افزونه را در ASP.NET Webforms بررسی خواهیم کرد.

الف) دریافت افزونه

لطفاً به آدرس [GitHub](#) [ذکر شده](#) مراجعه نمایید.

سپس برای مثال پوشه‌ی js را به پروژه افزوده و فایل‌های jquery-1.5.min.js ، jquery.autocomplete.js ، jquery.autocomplete.css و indicator.gif را در آن کپی کنید. فایل indicator.gif به همراه مجموعه‌ی دریافتی ارائه نمی‌شود و یک آیکن [loading](#) معروف می‌تواند باشد. علاوه بر آن یک فایل جدید custom.js را نیز جهت تعاریف سفارشی خودمان اضافه خواهیم کرد.

ب) افزودن تعاریف افزونه به صفحه

در ذیل نحوه‌ی افزودن فایل‌های فوق به یک master page نمایش داده شده است. در اینجا از قابلیت‌های جدید ScriptManager (موجود در سرویس پک یک دات نت سه و نیم و یا دات نت چهار) جهت یکی کردن اسکریپت‌ها کمک گرفته شده است. به این صورت تعداد رفت و برگشت‌ها به سرور به‌جای سه مورد (تعداد فایل‌های اسکریپت مورد استفاده)، یک مورد (نهایی یکی شده) خواهد بود و همچنین حاصل نهایی به صورت خودکار به شکلی فشرده شده به مرورگر تحویل داده شده، سرآیندهای کش شدن اطلاعات به آن اضافه می‌گردد (که در سایر حالات متداول اینگونه نیست)؛ به علاوه Url نهایی آن هم بر اساس hash فایل‌ها تولید می‌شود. یعنی اگر محتوای یکی از این فایل‌ها تغییر کرد، چون Url نهایی تغییر می‌کند، دیگر لازم نیست نگران کش شدن و به روز نشدن اسکریپت‌ها در سمت کاربر باشیم.

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="AspNetjQueryAutocompleteTest.Site" %>

<!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>
    <asp:Placeholder Runat="server">
        <link href="%= ResolveClientUrl("~/js/jquery.autocomplete.css")%" rel="stylesheet"
type="text/css" />
    </asp:Placeholder>
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
    <asp:ScriptManager ID="ScriptManager1" runat="server">
        <CompositeScript>
            <Scripts>
                <asp:ScriptReference Path="~/js/jquery-1.5.min.js" />
                <asp:ScriptReference Path="~/js/jquery.autocomplete.js" />
```

```

        <asp:ScriptReference Path="~/js/custom.js" />
    </Scripts>
</CompositeScript>
</asp:ScriptManager>
<div>
    <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
    </asp:ContentPlaceHolder>
</div>
</form>
</body>
</html>

```

علت استفاده از ResolveClientUrl در حین تعریف فایل css در اینجا به عدم مجاز بودن استفاده از ~ جهت مسیر دهی فایل‌های css در header صفحه بر می‌گردد.

(ج) افزودن یک صفحه‌ی ساده به برنامه

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
    CodeBehind="default.aspx.cs" Inherits="AspNetjQueryAutocompleteTest._default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <asp:TextBox ID="txtShenas" runat="server" />
</asp:Content>

```

فرض کنید می‌خواهیم افزونه‌ی ذکر شده را به TextBox استاندارد فوق اعمال کنیم. ID این TextBox در نهایت به شکل ContentPlaceHolder1\_txtShenas رندر خواهد شد. البته در ASP.NET 4.0 با تنظیم ClientIDMode=Static می‌توان ID انتخابی خود را به جای این ID خودکار در نظر گرفت و اعمال کرد. اهمیت این مساله در قسمت (ه) مشخص می‌گردد.

(د) فراهم آوردن اطلاعات مورد استفاده توسط افزونه‌ی AutoComplete به صورت پویا

مهم‌ترین قسمت استفاده از این افزونه، تهیه‌ی اطلاعاتی است که باید نمایش دهد. این اطلاعات باید به صورت فایلی که هر سطر آن حاوی یکی از آیتم‌های مورد نظر است، تهیه گردد. برای این منظور می‌توان از فایل‌های ASHX یا همان Generic handlers استفاده کرد:

```

using System;
using System.Data.SqlClient;
using System.Text;
using System.Web;

namespace AspNetjQueryAutocompleteTest
{
    public class AutoComplete : IHttpHandler
    {
        public void ProcessRequest(HttpContext context)
        {
            string prefixText = context.Request.QueryString["q"];
            var sb = new StringBuilder();

            using (var conn = new SqlConnection())
            {
                //todo: این مورد باید از فایل کانفیگ خوانده شود
                conn.ConnectionString = "Data Source=(local);Initial Catalog=MyDB;Integrated Security =
true";
                using (var cmd = new SqlCommand())
                {
                    cmd.CommandText = @" select Field1 ,Field2 from tblData where Field1 like
@SearchText + '%' ";
                    cmd.Parameters.AddWithValue("@SearchText", prefixText);
                    cmd.Connection = conn;
                    conn.Open();
                    using (var sdr = cmd.ExecuteReader())
                    {
                        if (sdr != null)

```

```

        while (sdr.Read())
        {
            string field1 = sdr.GetValue(0) == DBNull.Value ? string.Empty :
sdr.GetValue(0).ToString().Trim();
            string field2 = sdr.GetValue(1) == DBNull.Value ? string.Empty :
sdr.GetValue(1).ToString().Trim();
            sb.AppendLine(field1 + "|" + field2);
        }
    }
}

context.Response.Write(sb.ToString());
}

public bool IsReusable
{
    get
    {
        return false;
    }
}
}
}

```

در این مثال از ADO.NET کلاسیک استفاده شده است تا به عمد نحوه‌ی تعریف پارامترها یکبار دیگر مرور گردند. اگر از LINQ to SQL یا Entity framework یا NHibernate و موارد مشابه استفاده می‌کنید، جای نگرانی نیست؛ زیرا کوئری‌های SQL تولیدی توسط این ORMs به صورت پیش فرض از نوع پارامتری هستند ( [+](#) ).

در این مثال اطلاعات دو فیلد یک و دوی فرضی از جدولی با توجه به استفاده از like تعریف شده دریافت می‌گردد. به عبارتی همان متد StartsWith معروف LINQ بکارگرفته شده است. به صورت خلاصه افزونه، کوئری استرینگ q را به این فایل ashx ارسال می‌کند. سپس کلیه آیتم‌های شروع شده با مقدار دریافتی، از بانک اطلاعاتی دریافت شده و هر کدام قرارگرفته در یک سطر جدید بازگشت داده می‌شوند. اگر دقت کرده باشید در قسمت sb.AppendLine، با استفاده از "|" دو مقدار دریافتی از هم جدا شده‌اند. عموماً یک مقدار کفایت می‌کند (در 98 درصد موارد) ولی اگر نیاز بود تا توضیحاتی نیز نمایش داده شود از این روش نیز می‌توان استفاده کرد. برای مثال یک مقدار خاص به همراه توضیحات آن به عنوان یک آیتم نمایش داده شده مد نظر است.

ه) اعمال نهایی افزونه به TextBox

در ادامه پیاده سازی فایل custom.js برای استفاده از امکانات فراهم شده در قسمت‌های قبل ارائه گردیده است:

```

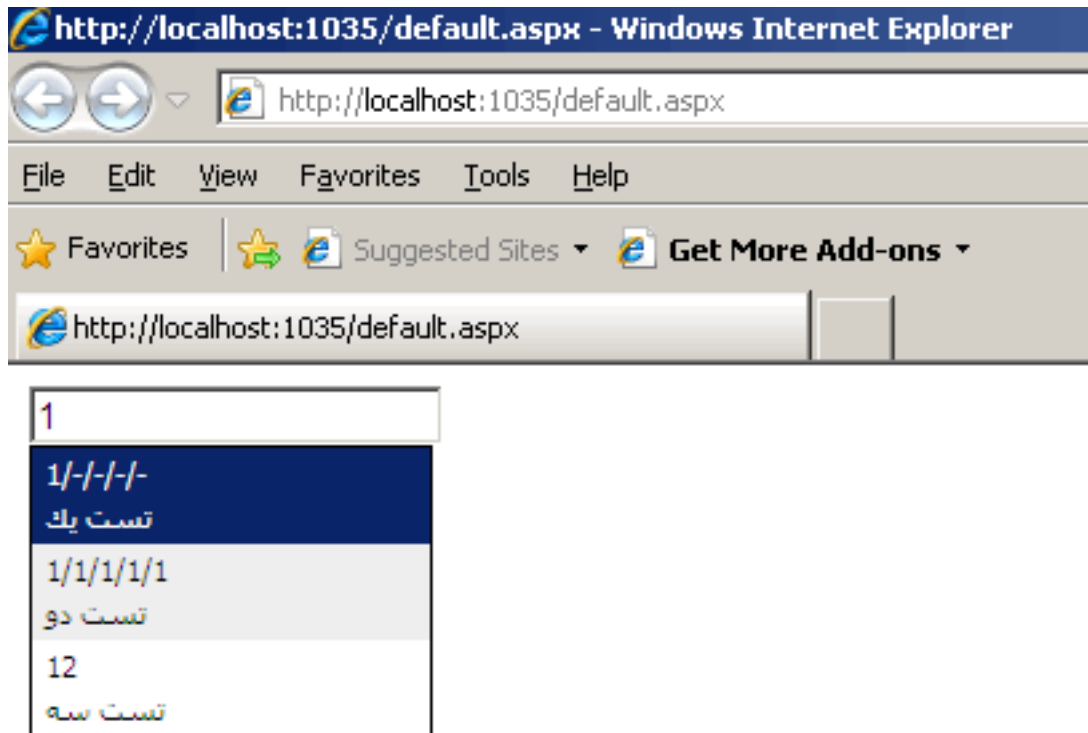
function formatItem(row) {
    return row[0] + "<br/><span style='text-align:justify;' dir='rtl'>" + row[1] + "</span>";
}

$(document).ready(function () {
    $("#ContentPlaceholder1_txtShenas").autocomplete('AutoComplete.ashx', {
        //Minimum number of characters a user has to type before the autocompleter activates
        minChars: 0,
        delay: 5,
        //Only suggested values are valid
        mustMatch: true,
        //The number of items in the select box
        max: 20,
        //Fill the input while still selecting a value
        autoFill: false,
        //The comparison doesn't looks inside
        matchContains: false,
        formatItem: formatItem
    });
});

```

پس از این مقدمات، اعمال افزونه‌ی autocomplete به TextBox ایی با id مساوی ContentPlaceholder1\_txtShenas ساده

است. اطلاعات از فایل AutoComplete.ashx دریافت می‌گردد و تعدادی از خواص پیش فرض این افزونه در اینجا مقدار دهی شده‌اند. لیست کامل آن‌ها را در فایل jquery.autocomplete.js می‌توان مشاهده کرد. تنها نکته‌ی مهم آن استفاده از پارامتر اختیاری formatItem است. اگر در حین تهیه‌ی AutoComplete.ashx خود تنها یک آیتم را در هر سطر نمایش می‌دهید و از "|" استفاده نکرده‌اید، نیازی به ذکر آن نیست. در این مثال ویژه، فیلد یک در یک سطر و فیلد دو در سطر دوم یک آیتم نمایش داده می‌شوند:



## نظرات خوانندگان

نویسنده: شاهین کیاست  
تاریخ: ۱۳۸۹/۱۱/۲۳ ۲۰:۱۵:۳۴

سلام آقای نصیری.

بسیار ممنونم (:

چند تا سوال دارم.

(1 این روش چه برتری بر پلاگین jQuery UI دارد؟

(2 می شود از Web Service هم استفاده کرد. آیا Generic handlers سرعت و کارایی بهتری می دهد؟

چگونه می توان ID رکوردی که بازگشته را در جایی نگه داشت تا جهت انجام عملیات سمت سرور از آن استفاده کرد. با تشکر.

نویسنده: وحید نصیری  
تاریخ: ۱۳۸۹/۱۱/۲۳ ۲۱:۲۲:۵۱

سلام.

- برتری خاصی ندارد. فقط اگر کسی نمی‌خواهد از jQuery UI و فایل‌های آن استفاده کند، این روش هم هست و سبک‌تر است. در این مورد باز هم بگردید افزونه هست. البته بعدش باید بگردید به چه صورت باید دیتا به آن‌ها پاس کرد و روش‌ها یکی نیست.

- بله. حتی می‌شود از یک صفحه معمولی aspx هم استفاده کرد. در page\_load آن کافی است این Response.Write ذکر شده قرار گیرد. در این مثال فوق از یکی از چندین روش ممکن، استفاده شده با سربرار حداقل ممکن.

- سوال خوبیه. به عبارتی اگر قرار است بجای یک dropdown که قبلا text هر ردیف آن متن خاصی بود و value آن id یا همان primary key رکورد جاری و از دید کاربر مخفی بود، الان باید چکار کرد تا همان id را بتوان به سرور ارسال کرد. چون به ظاهر فقط text دریافت می‌شود.

شما در اینجا از همان روش row\_0 و row\_1 که ذکر شد استفاده کنید. row\_1 را با id در سمت سرور پر کنید id+"|"+field1. سپس این پلاگین متد result هم دارد (مقدار انتخاب شده توسط کاربر را بازگشت می‌دهد). به این صورت فرصت خواهد بود تا یک فیلد مخفی را جهت ارسال به سرور مقدار دهی کرد:

```
(result(function (evt, row, formatted.  
}  
;([hiddenIDbox]).val(row[1#"")$  
{
```

لزومی هم ندارد تا این id یا همان row\_1 را در تابع formatItem نمایش داد. برای نمایش همان row\_0 کافی است.

نویسنده: محسن نجف زاده  
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۰:۲۵

پیشنهادهای امنیتی تون برای استفاده از فایل‌های ASHX یا Generic handlers چیست ؟

ممنون

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۰:۲۸

در مثال جاری با توجه به اینکه ورودی دریافتی از کاربر به کمک SQL پارامتری توسط ADO.NET پردازش می‌شود، مقاومت در برابر تزریق اس‌کیوال آن افزایش یافته است.

نویسنده: محسن نجف زاده  
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۰:۴۶

منظورم در حالت کلی است.

یک موردش : "چنانچه نیاز به استفاده از Session در فایل ASHX است، واسط IReadOnlySessionState به کار گرفته شود. "

نویسنده: وحید نصیری  
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۰:۵۱

در حالت کلی تفاوت خاصی وجود ندارد با سایر قسمت‌ها و [مسایل دیگر امنیتی](#) ASP.NET. بیشتر، ورودی‌های دریافتی توسط آن‌را باید صحیح بکار بگیرید و مراقب آن‌ها باشید.