

برای استفاده از قابلیت‌های Ajax کتابخانه jQuery، شش متد زیر در اختیار برنامه نویس‌ها است:

`$.ajax()`, `load()`, `$.get()`, `$.getJSON()`, `$.getScript()`, and `$.post()`

که در حقیقت 5 مورد آخر ذکر شده صرفاً بیان اولین متد ajax فوق به نحوی دیگر می‌باشند و محصور کننده توانایی‌های آن هستند.

برای مثال کد زیر زمان جاری را از سرور دریافت کرده و نتیجه را در سه تکست باکس قرار داده شده در صفحه نمایش می‌دهد. ابتدا وب سرویس ساده زیر را در نظر بگیرید که زمان شمسی جاری را بر می‌گرداند:

```
using System;
using System.Globalization;
using System.Web.Script.Services;
using System.Web.Services;

namespace TestjQueryAjax
{
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    [ScriptService]
    public class AjaxSrv : WebService
    {
        public class TimeInfo
        {
            public string Date { set; get; }
            public string Hr { set; get; }
            public string Min { set; get; }
        }

        [WebMethod(EnableSession = true)]
        [ScriptMethod(ResponseFormat = ResponseFormat.Json)]
        public TimeInfo GetTime()
        {
            DateTime now = new DateTime(DateTime.Now.Year, DateTime.Now.Month, DateTime.Now.Day);
            PersianCalendar pc = new PersianCalendar();
            string today = string.Format("{0}/{1}/{2}",
                                         pc.GetYear(now),
                                         pc.GetMonth(now).ToString("00"),
                                         pc.GetDayOfMonth(now).ToString("00"));

            TimeInfo ti = new TimeInfo
            {
                Date = today,
                Hr = DateTime.Now.Hour.ToString("00"),
                Min = DateTime.Now.Minute.ToString("00")
            };
            return ti;
        }
    }
}
```

سپس اگر از تابع Ajax کتابخانه jQuery جهت دریافت زمان جاری از وب سرویس استفاده نمائیم، کد صفحه ما به صورت زیر خواهد بود:

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

<!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 type="text/javascript">
function validate() {
$.ajax({
type: "POST",
url: 'AjaxSrv.aspx/GetTime',
data: '{}',
contentType: "application/json; charset=utf-8",
dataType: "json",
success:
function(msg) {
$("#<%=txtDate.ClientID %>").val(msg.d.Date);
$("#<%=txtHr.ClientID %>").val(msg.d.Hr);
$("#<%=txtMin.ClientID %>").val(msg.d.Min);
},
error:
function(XMLHttpRequest, textStatus, errorThrown) {
alert("خطایی رخ داده است");
}
});
//debugger;
}
</script>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:TextBox ID="txtDate" runat="server" />
<br />
<asp:TextBox ID="txtHr" runat="server" />
<br />
<asp:TextBox ID="txtMin" runat="server" />
<br />
<asp:Button ID="btnGetTime" runat="server" Text="Click here!" UseSubmitBehavior="false"
OnClientClick="validate();return false;" />
</div>
</form>
</body>
</html>

```

تنها نکته‌ی جدید این اسکریپت، نحوه‌ی استفاده از خروجی JSON وب متد ما است که از نوع TimeInfo تعریف شده است. خروجی نمونه این وب متد به صورت زیر می‌تواند باشد:

```

{"d":{"__type":"TestjQueryAjax.AjaxSrv+TimeInfo","Date":"1388/07/14","Hr":"12","Min":"59"}}

```

که نحوه‌ی دسترسی به اجزای آن را در متد validate ملاحظه می‌نمائید.

باید به خاطر داشت که برای هر 6 متد Ajax ایی jQuery ، عملیات کش شدن اطلاعات در مرورگر کاربر به صورت پیش فرض فعال است. اما این نکته تنها زمانی که dataType مورد استفاده از نوع script یا jsonp باشد، صادق نبوده و کش شدن به صورت خودکار غیرفعال می‌گردد.

روش سنتی غیرفعال کردن کش در حین عملیات اجکسی، استفاده از یک کوئری استرینگ متغیر در پایان url درخواستی است. به این صورت مرورگر درخواست صادره را جدید فرض کرده و از کش خود استفاده نمی‌نماید (همین مورد در حالت کش شدن تصاویر هم صادق است).

jQuery نیز همین عملیات را در پشت صحنه انجام داده اما تنظیم آن را به نحوی مطلوب‌تری ارائه می‌دهد. یا پارامتر cache را در تعریف متد ajax خود اضافه نموده و مقدار آن را مساوی false قرار دهید و یا جهت تاثیر گذاری بر روی کلیه متدهای مورد استفاده، پیش از استفاده از آن‌ها این تنظیم را مشخص سازید:

```

$.ajaxSetup({cache: false});

```

نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۱۳۸۹/۱۱/۱۱ ۲۱:۵۶:۴۰

سلام.
علت Return False که آخر این دستور گذاشتید چیست؟
;OnClientClick="validate();return false
ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۱۱/۱۱ ۲۳:۲۰:۰۰

سلام، از UseSubmitBehavior هم استفاده شده. علت غیرفعال کردن postback است تا بشود نمایی از عملیات Ajax را مشاهده کرد.