

فرض کنید تعیین اعتبار یکی از فیلدهای فرم نیاز به انجام محاسباتی در سمت سرور دارد و این کار را می‌خواهیم با استفاده از jQuery Ajax انجام دهیم. مشکلی که در اینجا وجود دارد، این است که Ajax به معنای asynchronous است. یعنی زمانیکه کاربر دکمه submit را فشرده، دیگر برنامه منتظر این نخواهد شد که پاسخ کامل دریافت شود، سایر پردازش‌ها صورت گیرد و سپس فرم را به سرور ارسال نماید (شبیه به ایجاد یک ترد جدید در برنامه‌های ویندوزی). مثال زیر را در نظر بگیرید:

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

<!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() {
            var number1 = $("#<%=txtNumber1.ClientID %>").val();
            var number2 = $("#<%=txtNumber2.ClientID %>").val();
            var result = false;
            $.ajax({
                type: "POST",
                url: 'AjaxSrv.aspx/ValidateIt',
                data: '{"number1":' + number1 + ', "number2":' + number2 + '}',
                contentType: "application/json; charset=utf-8",
                dataType: "json",
                success:
                    function(msg) {
                        if (msg.d) {
                            result = true;
                            alert('بسیار خوب');
                        }
                        else {
                            result = false;
                            alert('دوباره سعی کنید');
                        }
                    },
                error:
                    function(XMLHttpRequest, textStatus, errorThrown) {
                        result = false;
                        alert("خطایی رخ داده است");
                    }
            });
            //debugger;
            return result;
        }
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            number 1 :
            <asp:TextBox runat="server" ID="txtNumber1" />
            <br />
            number 2 :
            <asp:TextBox runat="server" ID="txtNumber2" />
            <br />
            <asp:Button ID="btnSubmit" Text="Submit" UseSubmitBehavior="false" runat="server"
                OnClientClick="if(!validate()){ return false;}" OnClick="btnSubmitClick" />
        </div>
    </form>
</body>
</html>
```

این مثال یک نوع اعتبار سنجی سفارشی را در حین submit با استفاده از وب سرویس زیر انجام می‌دهد (حاصلضرب دو عدد دریافتی را بررسی می‌کند که باید مساوی 10 باشند. البته هدف از این مثال ساده، آشنایی با نحوه‌ی انجام این نوع عملیات است که می‌تواند شامل کار با دیتابیس و غیره هم باشد. و گرنه بدیهی است این بررسی را با دو سطر کد جاوا اسکریپتی نیز می‌شد انجام داد):

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Web.Script.Services;

namespace TestjQueryAjax
{
    /// <summary>
    /// Summary description for AjaxSrv
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    [ScriptService]
    public class AjaxSrv : System.Web.Services.WebService
    {
        [WebMethod(EnableSession = true)]
        [ScriptMethod(ResponseFormat = ResponseFormat.Json)]
        public bool ValidateIt(int number1, int number2)
        {
            return number1 * number2 == 10;
        }
    }
}
```

در این مثال هنگام submit، عملیات اعتبار سنجی با توجه به وضعیت asynchronous عملیات Ajax، تمام و کمال رخ نداده، یا اعتبار سنجی انجام نمی‌شود و یا پیام خطایی را دریافت خواهیم کرد.

راه حل چیست؟

راه حل‌های فضایی بسیاری را در وب در این مورد می‌توان پیدا کرد؛ اما راه حل استاندارد آن در این حالت ویژه، استفاده از Ajax در حالت غیرهمزمان است. یعنی این فریم ورک Ajax را وادار کنیم که تا پایان عملیات مورد نظر، منتظر بماند و سپس فرم را ارسال کند. برای این منظور تنها کافی است یک سطر زیر را پیش از فراخوانی تابع Ajax، اضافه و فراخوانی نمائیم:

```
$.ajaxSetup({async: false});
```

نکته:

UseSubmitBehavior دکمه ما را به شکل زیر رندر می‌کند (دکمه به یک button معمولی (بجای حالت submit) تبدیل شده و سپس یک doPostBack را اضافه خواهد کرد):

```
<input id="btnSubmit" type="button" onclick="if(!validate()){ return false;};__doPostBack('btnSubmit','')" value="Submit" name="btnSubmit"/>
```

## نظرات خوانندگان

نویسنده: Anonymous  
تاریخ: ۱۳۸۸/۰۵/۱۲ ۰۹:۳۲:۴۸

سلام و خسته نباشید.

جناب نصیری من کلا هدف شما از این کار رو متوجه نشدم.

یعنی شما قصد اینو دارید که تا اعتبار سنجی انجان نشده و جواب فرستاده نشده فرم منتظر بمونه ؟

نویسنده: وحید نصیری  
تاریخ: ۱۳۸۸/۰۵/۱۲ ۱۰:۰۴:۵۴

سلام،

دو هدف در این مثال مد نظر بود:

الف) تا پایان عملیات اعتبار سنجی توسط عملیات Ajax ایی صبر شود و فرم به یکباره به سرور ارسال نگردد.

ب) اگر فرم اعتبار سنجی نشد، به سرور submit نشود.