به نام خداوند بخشنده مهربان

آموزش طراحی صفحات وب

تکنولوژی Ajax

تالیف و گردآوری

مهرداد فتاحی – ندا زرودی

Developer Studio Network

برای دریافت اطلاعات کاملتر و آموزش سایر مباحث طراحی وب به وب سایت ما سر بزنید

www.Developer1.ir

۱۳۹۲

مشخصات کتاب

نام کتاب : آموزش تکنولوژی Ajax همراه با مثال های عملی

نویسنده : مهرداد فتاحی – ندا زرودی

ناشر : www.Developer1.ir

سال : ۱۳۹۲

تمام مطالب این کتاب همراه با توضیحات و مطالب و مثال های آموزشی بیشتر ، بر روی پرتال آموزش Ajax در سایت ما قرار دارد .

درباره سایت ما

سایت www.Developer1.ir **یک سایت آموزشی رایگان در زمینه آموزش کامپیوتر ، طراحی وب و برنامه نویسی موبایل است . برای دریافت آموزش های بیشتر در مورد سایر مباحث به وب سایت ما سر بزنید .**

در این سایت آموزش MySQL , SQL , jQuery , XML , ASP.Net , PHP , Java Script , CSS , HTML **و مجموعه آفیس ۲۰۰۷ به صورت تصویری قرار داده شده است .**

همچنین برنامه نویسی اندروید و ویندوز فون ۸ نیز جز برنامه های آینده می باشد .

فهرست مطالب آموزشی

مقدمه آموزش Ajax - ایجکس چیست و چه کاربردی دارد ؟

AJAXچیست؟ واژه AJAX با تلفظ ایجکس یا ایژاکس سرنام عبارت Asynchronous Java and XML و به معني تركيب نامتقارن جاوا اسكريپت و XML است. ماهيت صفحات وب و پروتكل HTTP به گونه‌اي است كه به طور معمول وقتي درحال وب‌گردي هستيم، به ازاي هر كنش و واكنش ميان ما و سايتي كه در حال كار با آن هستيم، كل يک صفحه وب از نو بارگذاري و تازه‌سازي (refresh) مي‌شود.

ايجكس فناوري جديدي است كه تغيير محسوسي را در اين سناريو به وجود مي‌آورد؛ به اين ترتيب كه به جاي بارگذاري مجدد كل صفحه، فقط قسمتي تغيير مي‌كند كه قرار است اطلاعات جديد را به نمايش درآورد و كليه عمليات ارسال اطلاعات و دريافت نتايج در پشت صحنه انجام مي‌شود. در نتيجه هيچ‌گاه صفحه سفيد و خالي وب در فواصل كنش و واكنش‌هاي هنگام كار با مرورگر ديده نمي‌شود و احساسي مشابه تجربه كار با يک نرم‌افزار دسكتاپ به كاربر دست مي‌دهد.

جادوي ايجاكس چيزي نيست جز يک فكر بكر و آن هم تركيب كردن جاوا اسكريپت و XML در قالب يک موجود نرم‌افزاري جديد.
اصطلاح AJAX را اولين بار كارشناسي از شركت Adaptive Path به نام جسي جيمزگرت در مقاله‌اي با عنوان ايجكس؛ رهيافت جديدي در برنامه‌هاي تحت وب مطرح كرد و خيلي سريع مورد استقبال گسترده برنامه نويسان وب در سراسر جهان قرارگرفت. اعتقاد عمومي اين است كه تاريخچه به كارگيري تكنيک مذكور به پيدايش نرم‌افزار Outlook WebAccess و استفاده از فرمان XMLHttpRequest كه مايكروسافت ابداع كرده و در نرم‌افزار اينترنت اكسپلور به‌كار رفته است، برمي‌گردد. اما امروزه اغلب مرورگرهاي مهم و شناخته‌شده (از جمله فايرفاكس) از آن پشتيباني مي‌كنند و ديگر يک فناوري محدود به اينترنت اكسپلورر نيست.

چرا Ajax این قدر اهمیت دارد؟

فناوري ايجكس جديد است و شگفت‌انگيز به‌نظر مي‌رسد. ولي در حقيقت كل اين شعبده بر اساس فناوري‌هايي بنا شده است كه هم‌اكنون موجودند: جاوا اسكريپت و XML. هر دوي اين فناوري‌ها تا حد زيادي باز هستند و منحصر به شركت خاصي نيستند. به همين دليل اين روزها تمام محافل دنياي برنامه نويسي مملو از مقالات و تحليل‌هايي درباره AJAX است. به عنوان نمونه مي‌توانيد صفحه نخست سايت موسوم به كانال شبكه برنامه نويسان سان را بازكنيد. بيشتر اين صفحه (در زمان نگارش اين يادداشت) به مقالات و مطالب متنوعي درباره AJAX اختصاص يافته است. مجلات برنامه‌نويسي نيز مقالات متعددي در اين زمينه منتشر كرده‌اند و سايت‌هاي معروفي مانند O" Reilly و xml.com نيز در اين باره مطلب دارند.
بنابراين ظرفيت بالايي براي تبديل ايجكس به يک استاندارد جهاني وجود دارد. از اين رو قرار است در تاريخ سيزدهم مارس سمينار مهمي در زمينه Ajax برگزار شود. اگر همين الان به سايت ajaxseminar.com مراجعه كنيد، متوجه مي‌شويد كه علاوه بر طراح اين تكنيک، چندين برنامه‌نويس شاخص از شركت‌هاي بزرگي همچون ياهو در آن شركت خواهند داشت.

اين فناوري از يک جنبه ديگر نيز اهميت دارد. به‌دليل عدم نياز به بارگذاري مجدد کل يک صفحه وب، مقدار داده‌هايي که لازم است براي تکميل يک Interaction ميان کاربر و سايت مبادله شود، به شدت کاهش مي‌يابد و اين به معني افزايش محسوس سرعت نرم‌افزارهاي تحت وب، سهولت به‌کارگيري اينترفيس‌هاي مبتني بر اي‌جکس و کاربرپسندترشدن آن‌ها مي‌شود. به همين دليل اين روزها اکثر پورتال‌هاي بزرگ (مانند ياهو) مشغول طراحي اينترفيس‌هاي مبتني بر AJAX هستند.

با اين همه مزاياي اي‌جکس به همين موارد ختم نمي‌شود. تصور کنيد وقتي در طول شبانه روز ميلياردها بار فرآيند بارگذاري مجدد صفحات وب تعاملي در شبکه اينترنت تکرار مي‌شود، استفاده از اي‌جکس چگونه موجب آزاد شدن پهناي‌باند اينترنت و در نتيجه ميلياردها دلار صرفه‌جويي اقتصادي در اين زمينه مي‌شود و اين به نوبه خود انقلابي در عرصه وب و فضاي سايبر به شمار مي‌رود. اين تحول در شرايطي اتفاق مي‌افتد که همزمان فناوري‌هاي اينترنت پرسرعت همچون ADSL به شدت در حال رشد هستند.

حال تصور کنيد اگر فناوري اي‌جکس در مقياس گسترده‌اي مورد استفاده نرم‌افزارهاي تحت وب قرارگيرد. آنگاه افزايش سرعت کار با اينترنت چند برابر خواهد شد. چنين تحولي مي‌تواند اينترنت را به کامپيوتر دوم کاربران تبديل کند. به گونه‌اي که براي آن‌ها اجراي يک نرم‌افزار از روي وب تفاوت محسوسي با اجراي آن از روي کامپيوتر دسکتاپ نداشته باشد. در اين صورت ممکن است واقعا بخش مهمي از توان پردازشي نرم‌افزارها، چه از نوع دسکتاپ و چه از نوع تحت وب به سيستم‌هاي موسوم به Web Service واگذارشود. از اين رو کساني که مبتکر و مشوق ايده وب سرويس بوده‌اند، اين روزها انگيزه تازه‌اي براي سرعت بخشيدن به اين فناوري پيدا کرده‌اند و تلاش مي‌کنند دو فناوري اي‌جکس و وب سرويس را به‌يکديگر پيوند دهند و راهکارهاي تازه‌اي بيافرينند.

کدام سايت‌ها از اي‌جکس استفاده کرده‌اند؟

دنياي برنامه‌نويسي وب هنوز درگير هيجانات مربوط به آشنايي با اين پديده است و در حال بررسي مشکلات تکنيکي و هضم و جذب آن در بافت برنامه‌هاي تحت وب مي‌باشد. به همين دليل ميزان استفاده عملي از آن چندان گسترده نيست. اما چون کارايي اي‌جکس ديگر براي همه ثابت شده است، حرکت‌هاي بزرگي در سراسر اينترنت به سمت استفاده از اين فناوري در جريان است. از جمله اينترفيس نسخه دوم Yahoo Mail که به نسخه بتا معروف است و فعلا در دسترس کاربران سرويس غير رايگان ياهو قراردارد، از همين فناوري استفاده مي‌کند که به زودي اينترفيس جديد در دسترس عموم قرار مي‌گيرد. سرويس Gmail و Google Map نيز از اين فناوري استفاده مي‌کنند.

Ajax چگونه کار می کند ؟

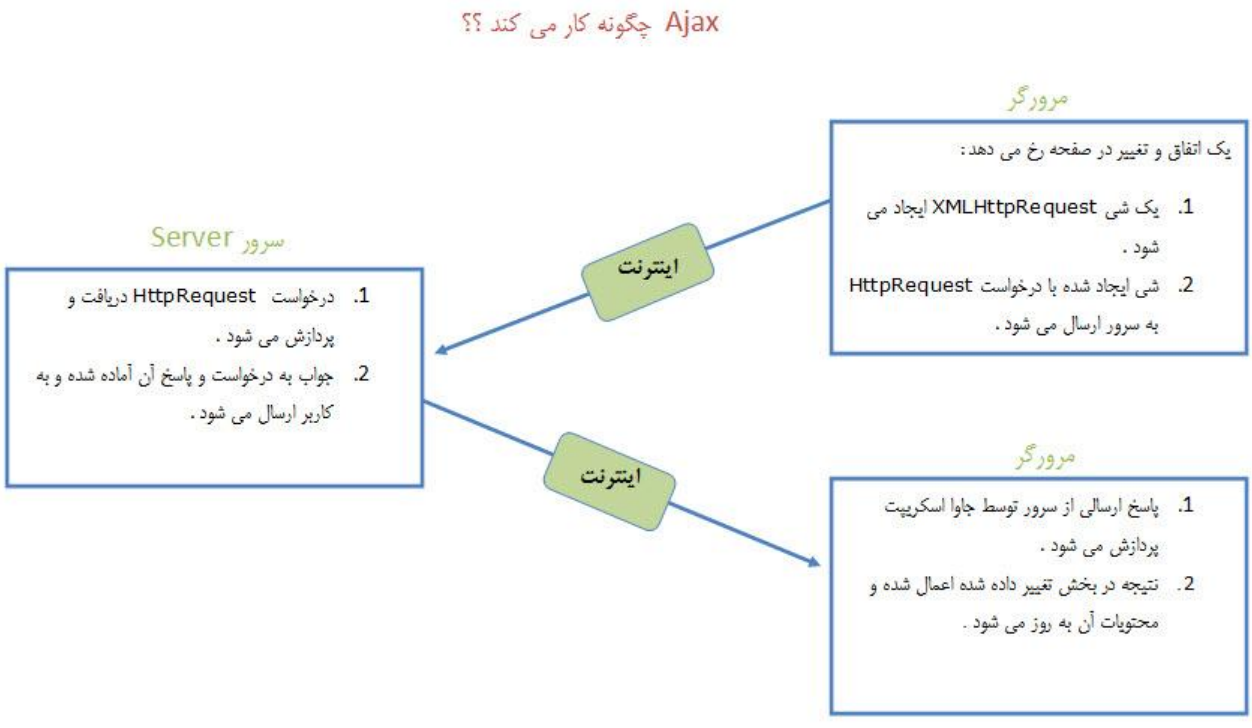
کاربرد و نحوه کارکرد Ajax :

همانطور که در بخش قبل به معرفی Ajax پرداختیم ، با مفهوم کلی این تکنیک آشنا شدید . در این بخش می خواهیم به تشریح کاربرد و نحوه کارکرد تکنیک ایجکس بپردازیم .

همانطور که در هنگام کار با صفحاتی که با زبان های سمت سرور ، مثل ASP.Net یا PHP طراحی شده اند متوجه شده اید ، زمانی که در این صفحات تغییری ایجاد نموده و یا مثلا کنترلی را کلیک نمایید ، صفحه به طور کامل (حتی بخش هایی که تغییر نکرده اند) به سرور ارسال شده و در واقع صفحه مجدد لود می شود .**سپس در** اجرای جدید صفحه نتیجه تغییر ایجاد شده نمایان می شود .
تصور کنید شما به طور مداوم می خواهید محتویات یک صفحه را تغییر دهید ، با هر بار ایجاد تغییر و ارسال کل صفحه به سرور ، چند مشکل عمده به وجود می آید :

۱. با ارسال کل حجم صفحه (به جای بخشی که فقط تغییر کرده **است**) و افزایش حجم اطلاعات مبادله شده ، مدت زمان فرایند پردازش افزایش یافته و باعث اتلاف وقت کاربر می شود.
۲. باعث افزایش ترافیک سرور می شود.
۳. باعث مصرف پهنای باند و مدت زمان بیشتر اینترنت می شود.
۴. در فرایند گسترده ایجاد تغییرات و ارتباط با سرور ، صفحه مرتبا لود و فراخوانی می شود.

اما برای این مشکلات باید چه کار کرد ؟
تکنیک Ajax برای حل این مشکلات ابداع شد . تکنیک ایجکس مانع ارسال کل صفحه به سرور و فراخوانی آن در هنگام تغییر در یک بخش از صفحه می شود .
دیاگرام زیر نشان دهنده نحوه کار Ajax **است** . به فرایند آن دقت نمایید :



به تشریح دیاگرام بالا می پردازیم .
هنگامی که تغییری در یک بخش از صفحه ایجاد می شود ، Ajax اطلاعات بخشی که تغییر کرده را بدون لود شدن صفحه و ارسال کامل آن به سرور ، توسط یک درخواست HttpRequest و توسط یک شی XMLHttpRequest در یک عملیات پشت پرده به سرور ارسال می کند .
سرور اطلاعات دریافتی را پردازش کرده و نتیجه را به صفحه ارسال می کند .
مرورگر اطلاعات ارسالی از سرور را دریافت و با Java Script پردازش می کند . سپس تغییرات لازم را در بخش تغییر کرده اعمال نموده و محتویات آن را به روز می کند .

یک مثال آشنا از کاربرد Ajax :

یکی از موارد کاربرد Ajax که حتما در گذشته با آن برخورد کرده اید و شاید نمی دانستید چگونه کار می کند ، قابلیت Suggest در موتور جستجوی گوگل است . هنگامی که شما یک حرف را در کادر موتور گوگل وارد می کنید ، یک عملیات پردازش انجام شده و به شما کلماتی که با حرف یا حروف وارد کرده شما ، شروع می شوند را در یک کادر نمایش می دهد . این عملیات توسط تکنیک ایجکس انجام می شود .

ساختار Ajax - تشریح Syntax ایجکس با یک مثال

تشریح ساختار Ajax :

در این بخش قصد داریم تا با ارائه یک مثال ساده از دستورات Ajax و نمایش خروجی واقعی آن ، شما را با ساختار کلی این تکنیک و نحوه عملکرد آن آشنا نماییم .
در مثال زیر ، یک صفحه ساده HTML داریم که در آن یک تگ < div > و یک دکمه فرمان button قرار دارد .
تگ < div > در ابتدا یک متن را نمایش می دهد و قصد داریم کاری کنیم تا پس از کلیک کاربر بر روی دکمه فرمان ، متن جدید آن از سرور دریافت شده و به روز رسانی شود .
کد موجود در صفحه HTML به صورت زیر است :

کد صفحه

```
< html >

< head >
  < script type = " text/javascript " >
    function loadXMLDoc ( )
      {
        .... در اینجا قرار می گیرد Ajax کد ها و دستورات ...
      }
  < /script >
< /head >

< body >
  < div id="myDiv" >
    تغییر دهید Ajax متن این قسمت را با
  < /div >
  < button type="button" onclick=" loadXMLDoc() " > Change Content <
/button >
< /body >

< /html >
```

در کد بالا ، هنگامی که کاربر بر روی دکمه فرمان کلیک می نماید ، تابع loadXMLDoc ( ) که کد آن در بخش < head > صفحه قرار دارد ، فراخوانی و اجرا شده و یک درخواست را به سرور ارسال می کند . **سرور** درخواست را دریافت کرده و آن را پردازش می کند . سپس خروجی تولید شده را به صفحه بر می گرداند . **مرورگر** این خروجی را دریافت کرده و محتویات تگ < div > را تغییر داده و به روز رسانی می کند .

کد تابع loadXMLDoc ( ) در جدول زیر قرار داده شده است . در قسمت بعدی ، به تشریح و آموزش نوشتن این دستورات خواهیم پرداخت .

کد تابع loadXMLDoc ( )	function loadXMLDoc() { var xmlhttp; if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari کد برای مرورگرهای جدید xmlhttp=new XMLHttpRequest(); } else { // code for IE6, IE5 کد برای مرورگرهای قدیمی تر xmlhttp=new ActiveXObject("Microsoft.XMLHTTP"); } xmlhttp.onreadystatechange=function() { if (xmlhttp.readyState==4 && xmlhttp.status==200) { document.getElementById("myDiv").innerHTML=xmlhttp.responseText; } } xmlhttp.open("GET","ajax_info.txt",true); xmlhttp.send(); }

آموزش Ajax - ساخت شی XmlHttpRequest

اساسی ترین شی در Ajax که تقریبا انجام تمام عملیات بر روی دوش آن است ، شی XMLHttpRequest **است** .
در این بخش قصد داریم تا شما را با شی XMLHttpRequest و نحوه تعریف و استفاده از آن آشنا نماییم .

تعریف و ساخت یک شی XMLHttpRequest :

شی XMLHttpRequest وظیفه ارسال و دریافت اطلاعات بین مرورگر کاربر و سرور را داشته و مانع لود شدن مجدد صفحه در هنگام بروز یک تغییر در صفحه می شود . این شی امکان انجام تغییرات در بخش های مختلف یک صفحه و تبادل اطلاعات با سرور را بدون اینکه صفحه Refresh شود را فراهم نموده است .

تمام مرورگرهای مطرح از شی XMLHttpRequest پشتیبانی کرده و به صورت درون ساخته آن را دارند .
در ورژن های قدیمی IE 6 و IE 5 به جای شی XMLHttpRequest ، از شی ActiveXObject پشتیبانی می **شود** . بنابراین در هنگام کد نویسی Ajax می توان کد را طوری نوشت که با این مرورگر ها نیز ، سازگاری داشته **باشد** . در ادامه به تشریح این مسئله خواهیم پرداخت .

شکل کلی تعریف یک شی XMLHttpRequest به صورت زیر است :

Syntax	variable = new XMLHttpRequest ( ) ; * variable : یک متغیر با نام دلخواه است

شکل کلی تعریف این شی در مرورگرهای قدیمی IE 6 , IE 5 و برای سازگاری با آنها به صورت زیر است :

Syntax	variable = new ActiveXObject (" Microsoft.XMLHTTP ") ;

مثال ساخت یک XMLHttpRequest :

پس از اینکه با نحوه تعریف یک شی XMLHttpRequest در Ajax آشنا شدید ، در کد مثال زیر یک نمونه از این شی را ساخته ایم . در این مثال برای سازگاری با تمام مرورگرهای جدید و قدیمی ، یک دستور شرطی را به کار برده ایم . در این دستور ابتدا وضعیت تعریف شی XMLHttpRequest در مرورگر بررسی شده و سپس با توجه به قابلیت مرورگر کد لازم استفاده می **شود** :
همانطور که مشاهده می نمایید ، شی XMLHttpRequest ساخته شده در متغیر xmlhttp ذخیره می شود . **پس** از تعریف این متغیر ، از آن برای کار با Ajax در سطح برنامه استفاده می شود . در بخش های بعدی توضیح این مسئله داده شده است .

Syntax	var xmlhttp; if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari xmlhttp = new XMLHttpRequest () ; } else { // code for IE6, IE5 xmlhttp = new ActiveXObject (" Microsoft.XMLHTTP ") ; }

آموزش Ajax - ارسال یک درخواست XmlHttpRequest به سرور

ارسال یک درخواست XMLHttpRequest به سرور :

در بخش قبل به آموزش نحوه ساخت یک شی XMLHttpRequest پرداختیم . این شی وظیفه ارسال و دریافت ، یک درخواست و اطلاعات لازم برای انجام عملیات تغییر در صفحه را دارد . پس از اینکه یک شی XMLHttpRequest را ساختید ، باید به وسیله آن درخواست خود را به سرور ارسال نمایید .
این درخواست اطلاعات لازم را به سرور ارسال کرده و یک فایل که حاوی اسکریپت ، اطلاعات و یا دستور لازم برای ایجاد تغییر در صفحه است را بر روی سرور باز می کند .
سرور پس از باز کردن فایل درخواستی ، آن را پردازش کرده و پاسخ لازم را به کامپیوتر کاربر ارسال می کند .

برای انجام این عملیات از ۲ متد ( ) send و ( ) open شی XMLHttpRequest استفاده می شود . با متد ( ) open ابتدا درخواست ساخته شده و سپس با متد ( ) send ارسال می شود .

شکل کلی ارسال یک درخواست به سرور به صورت زیر است :

Syntax	xmlhttp.open(metohd , url , async) ; xmlhttp.send (" text ") ;

مثل	xmlhttp.open("GET","ajax_info.txt",true); xmlhttp.send( );

در ادامه به توضیح موارد در Syntax دستور می پردازیم .

توضیح هر یک از موارد syntax	
متد	توضیح
open ( )	این متد درخواست را ایجاد کرده ودارای سه پارامتر اصلی زیر است . به وسیله این پارامترها متد نحوه ارسال اطلاعات ، آدرس فایلی که بر روی سرور باید باز شود و اینکه آیا درخواست به صورت asynchronous یا synchronous باشد ، تعیین می شود . در ادامه به توضیح این پارامترها به صورت کامل می پردازیم . • method • url • async
send ( )	این متد درخواست ایجاد شده را به سرور ارسال می کند . به وسیله پارامتر text ، زمانی که متد POST را برای ارسال درخواست انتخاب کرده اید ، می توانید اطلاعات مورد نظرتان را به همراه درخواست خود به

	سرور ارسال نمایید : • text

پارامتر اول : متد GET یا POST

method تعیین کننده نحوه ارسال اطلاعات به سرور توسط شی XMLHttpRequest است . دو روش برای این منظور وجود دارد : GET یا POST
متد GET سریعتر و ساده تر از متد POST است . در متد GET اطلاعات ارسالی به سرور به ادامه نام فایل درخواستی (URL) اضافه می شوند ، اما در متد POST اطلاعات به صورت مخفی و پشت پرده به سرور منتقل می شوند .

معمولا از متد GET برای ارسال اطلاعات استفاده می شود . اما در موارد زیر باید از متد POST استفاده کنید :

۱. هنگامی که نمی خواهید اطلاعات به صورت Cashed باشند . یعنی اطلاعات درون حافظه سرور باقی مانده و ممکن است در درخواست های بعدی از اطلاعات تکراری استفاده شود . برای مثال در عملیات updateیک فایل و یا ارسال و دریافت اطلاعات از پایگاه داده از متد Post استفاده کنید.
۲. حجم اطلاعات ارسالی زیاد باشد . متد GET در حجم اطلاعات ارسالی بسیار محدود بوده و در حجم های اطلاعات بالا باید از متد POST استفاده نمایید.
۳. هنگامی که می خواهید اطلاعات حساسی مثل رمز عبور را منتقل کنید ، باید از متد POST استفاده شود . زیرا در متد GET اطلاعات ارسالی به انتهای نام فایل درخواست شده اضافه شده و به همین دلیل قابل رویت هستند . اما در متد POST اطلاعات به صورت مخفی و پشت پرده منتقل می شوند.

۱. مثل ارسال درخولست با متد GET :

مثل ۱ ارسال با متد GET	xmlhttp.open(**"GET"** , "ajax_info.txt" , true) ; xmlhttp.send( );

۲. مثال ارسال درخواست به سرور با متد GET و فرستادن اطلاعات مورد نظر با اضافه کردن چند مقدار به ادامه نام فایل درخواستی :

مثل ۲ ارسال اطلاعات با متد GET	xmlhttp.open("GET" , **"demo_get2.asp?Name=Omid&Family=Rezaee"** , true) ; xmlhttp.send( );

۳. مثل ارسال درخواست به سرور با متد POST :

مثال ۳ ارسال درخولست با متد POST	xmlhttp.open(**"POST"** , "demo_post.asp" , true) ; xmlhttp.send( );

۴. مثال ارسال درخواست به سرور با متد POST و انتقل اطلاعات مورد ظر با آن:

اگر هنگام استفاده از متد POST بخواهید ، اطلاعات مورد نظرتان را نیز ارسال کنید ، باید یک HTTP header را به وسیله متد( ) setRequestHeader به متد خود اضافه کنید .سپس اطلاعات مورد ظر را با متد ( ) send به سرور ارسال کنید . کد زیر روش این کار را نمایش می دهد:

مثل ٤ ارسل درخولست و اطلاعات با متد POST	xmlhttp.open("POST" , "ajax_test.asp" , true) ; xmlhttp.setRequestHeader("Content-type" , "application/x-www-form-urlencoded") ; xmlhttp.send("Name=Omid&Family=Rezaee") ;

پارامتر دوم : url آدرس یک فایل بر روی سرور.

پارامتر url در متد ( ) open ، در هنگام ارسل درخولست Ajax به سرور ، آدرس یک فایل بر روی سرور را تعیین می کند .
سرور در هنگام دریافت درخواست XMLHttpRequest ، فایلی که آدرس آن را با پارامتر url تعیین کرده اید ، باز کرده و بسته به نوع آن ، آن را پرادزش می کند . این فایل می تواند هر نوع فایلی باشد مثلا یک فایل ساده متنی باشد که حاوی مقداری اطلاعات است . همچنین می تواند یک فایل اسکریپتی مثل java script، ASP.Net و یا PHP باشد ، که کدهای خاصی را اجرا می کند .

سرور پس از اجرای دستورات موجود در فایل مقصد ، پاسخ لازم را تولید کرده و به کامپیوتر کاربر ارسال می کند . در قسمت بعدی به آموزش نحوه دریافت این پاسخ خواهیم پرداخت .

۵. در مثال زیر به تعیین آدرس یک فایل برای پارامتر url پرداخته ایم .

مثل ٥ تعیین فایل مقصد با استفاده از خاصیت url	xmlhttp.open("POST" , **"demo_post.asp"** , true) ; xmlhttp.send( );

آموزش Ajax - دریافت پاسخ XmlHttpRequest ازسرور

مقدمه :

پس از اینکه یک شی XMLHttpRequest را ساخته و به وسیله آن درخواست خود را به سرور ارسال نمودید ، سرور دستورات برنامه Ajax تعیین شده را پردازش کرده و یک خروجی را به عنوان پاسخ درخواست ، به کاربر ارسال می کند .
مرورگر در کامپیوتر کاربر این پاسخ را دریافت کرده و تغییرات لازم را در صفحه و بخش مربوطه انجام خواهد داد .

به وسیله قابلیت Respone در ایجکس ، می توانید کدی طراحی نمایید تا مرورگر پاسخ سرور را دریافت کند .
برای این منظور از ۲ خاصیت شی XMLHttpRequest به شرح زیر استفاده می شود :

- خاصیت : responseText از این خاصیت در زمانی استفاده می شود که پاسخ سرور به صورت یک متن خالی از نوع داده ای string **است**.
برای مثال فرض کنید که محتوای یک عنصر پاراگراف باید با متن جدید جایگزین شود . در این حالت چون **جواب** سرور از نوع متن است ، از خاصیت responseText استفاده می شود.
- خاصیت : responseXML این خاصیت در زمانی استفاده می شود که پاسخ سرور به صورت یک فایل XMLبوده و یا قالب بندی داده ای شده باشد.

در ادامه به تشریح کاربرد این خاصیت ها و مثال استفاده از آنها پرداخته ایم .

استفاده از خاصیت responseText :

گفتیم زمانی که پاسخ سرور به اسکریپت و یا فایل درخواستی ایجکس ، به صورت متن ساده باشد ، از خاصیت responseText شی XMLHttpRequest ، برای دریافت پاسخ استفاده می شود . سپس شما می توانید این خاصیت را به صورت مستقیم در کد خود به کار ببرید .

مثال : در مثال زیر ابتدا یک شی XMLHttpRequest به نام xmlhttp ساخته شده است . سپس درخواستی به سرور برای باز کردن فایل ajax_ex1.txt و **استخراج متن درون آن فرستاده‌شده است** . سپس جواب خروجی از سرور به وسیله خاصیت responseText دریافت شده و با **متن موجود در تگ** < "div id ="ex_1 > جایگزین **شده است** .

Example	
کد	< html > < head > < script type="text/javascript"> function loadXMLDoc() { var xmlhttp; if (window.XMLHttpRequest)

```
        {
            // code for IE7+, Firefox, Chrome, Opera, Safari
            xmlhttp = new XMLHttpRequest( ) }
        else
        {
            // code for IE6, IE5
            xmlhttp = new ActiveXObject("Microsoft.XMLHTTP") ;
        }
   xmlhttp.onreadystatechange = function ( )
        {
          if ( xmlhttp.readyState == 4  &&  xmlhttp.status == 200 )
              {
                 document.getElementById(" ex_1 ").innerHTML =
xmlhttp.responseText;
              }
        }
       xmlhttp.open("GET", "ajax_ex1.txt", true);
       xmlhttp.send();
     }
   </script>
  < /head >
< body >
   < div id = " ex_1 " > بر روی دکمه زیر کلیک نمایید ajax برای تغییر متن به وسیله  < /div >
   < button type="button" onclick="loadXMLDoc( )"> تغییر متن < /button >
  < /body >
< /html >
```

خروجی

برای تغییر متن به وسیله *ajax* بر روی دکمه زیر کلیک نمایید

تغییر متن

استفاده از خاصیت responseXML :

پس از ایکنه با دریافت یک پاسخ ساده متنی از سرور آشنا شدید ، حال بایستی یک پاسخ قالب بندی شده به زبان XML را از سرور دریافت کنیم . برای این منظور از خاصیت responsXML شی XMLHttpRequest ، استفاده می کنیم .

در این پروسه ، ابتدا فایل XML خروجی از سرور دریافت شده و خاصیت responseXML آن را پردازش می کند . سپس اطلاعات آن را در اختیار مرورگر قرار می دهد تا در بخش هایی از صفحه که باید تغییر کنند ، استفاده شود .

برای درک بهتر این مسئله یک مثال ساده را برای شما تشریح می کنیم . در این مثال یک فایل XML به نام Book_List.xml داریم که درون آن لیست یک سری کتاب با نام نویسنده و سال انتشار **قرار دارد** . با اجرای **دستور** زیر این فایل از سرور توسط شی responseXML دریافت شده و عنوان کتاب ها از بین اطلاعات دریافتی استخراج می شود .**سپس تگ** < "div id = " ex-2 " > ، این عنوان ها را در خروجی نمایش می دهد . برای مشاهده عملیات و خروجی مثال بر رور دکمه فرمان دریافت عنوان ها ، کلیک نمایید .

Example

کد

```
< html >
  < head >
    < script type="text/javascript">
      function Book()
       {
         var xmlhttp;
         var txt,x,i;
         if (window.XMLHttpRequest)
          {
             // code for IE7+, Firefox, Chrome, Opera, Safari
             xmlhttp = new XMLHttpRequest( ) }
         else
          {
             // code for IE6, IE5
             xmlhttp = new ActiveXObject("Microsoft.XMLHTTP") ;
          }
    xmlhttp.onreadystatechange = function ( )
         {
           if ( xmlhttp.readyState == 4  &&  xmlhttp.status == 200 )
               {
                  xmlDoc = xmlhttp.responseXM L;
                  txt = "  ";
                  x = xmlDoc.getElementsByTagName ( "TITLE" ) ;
                 for ( i=0 ; i<x.length ; i++ )
                  {  txt=txt + x[i].childNodes[0].nodeValue + " " ; }
            document.getElementById("ex_2").innerHTML=txt;
          }
         xmlhttp.open(" GET " , "Book_List.xml " , true ) ;
         xmlhttp.send( );
       }
    </script>
  < /head >
< body >
    < div id = " ex_2 " > < /div >
```

< button type="button" onclick="Book( )"> مشاهده عنوان ها < /button > < /body > < /html >	
مشاهده عنوان ها	خروجی

خاصیت readyState در شی XMLHttpRequest :

هنگامی که درخواست خود را به سرور ارسال می کنید بنا بر جواب آن می خواهید تصمیمات خاصی را لحاظ نمایید .

رویداد( ) onreadystatechange هر زمان که وضعیت Ready State تغییر کند ، رخ می دهد . خاصیت Ready State وضعیت شی XMLHttpRequest را در هر لحظه نگهداری می کند . بطور کلی ۳ خاصیت یا رویداد در هنگام کار با XMLHttpRequest وجود دارد که بوسیله آن می توان جواب را از سرور دریافت کرده و بر اساس آن عمل نمود :

۱. onreadystatechange : این خاصیت یک تابع را نگهداری می کند . هر زمان که وضعیت خاصیت Ready State شی XMLHttpRequest تغییر کند این رویداد رخ می دهد و تابع را اجرا می کند.
۲. readyState : این خاصیت وضعیت شی XMLHttpRequest را در هر لحظه مشخص می کند . این شی می توان ۴ حالت را داشته باشد که بنابر آن یکی از اعداد ۰ تا ۴ را برمی گرداند:
: 0درخواست دریافت نشده یا جوابی برای آن جدا نشده است.
: 1اعتبار باسرور برقرارشد.
: 2درخواست به سرور رسیده و دریافت شده است.
: 3سرور در حال پردازش درخولست شما است.
: 4درخواست بطور کامل انجام شده و پاسخ آماده است.
۳. status : این پارامتر وضعیت کلی درخواست و شی را مشخص می کند . که می تواند ۲ حالت داشته باشد :
۱. 200 : OK
یعنی وضعیت درست است.
۲. 404 : Page not found
یعنی صفحه یا فایل مورد نظر پیدا نشده است.

نکته : در رویداد onreadystatechange تابعی را تعیین می کنیم تا در صورتی که جواب سرور به درخواست ما آماده و ارسال شد عمل مورد نظرمان را انجام دهد . این عمل مورد نظر معمولا عملیات به روز رسانی بخش تغییر کرده در صفحه را انجام می دهد .

مثال عملی :

در مثال زیر ، در یک دستور شرطی بررسی کرده ایم که اگر جواب سرور یعنی پارامتر readyState دارای مقدار ۴ که به معنای اتمام پردازش درخواست و آماده بودن جواب است ، بود . سپس وضعیت شی نیز بررسی شود تا چنانچه مقدارش ۲۰۰ به معنی ok عملیات است ، سپس دستور به روز رسانی در صفحه اجرا شود :

مثال	xmlhttp.onreadystatechange=function( ) { if (xmlhttp.readyState==4 && xmlhttp.status==200)

```
        {
            document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
        }
    }
```

آموزش Ajax > مثال های عملی کار با Ajax > کاربرد Ajax در PHP

کاربرد Ajax در PHP :

در این بخش قصد داریم ، تا با ارائه یک مثال عملی ، نحوه استفاده از دستورات Ajax در یک صفحه PHP را به صورت گام به گام ، شرح دهیم .
در این مثال یک کادر متن داریم که به تایپ کردن کاربر درون آن ، حساس است . هر زمان که کاربر کاراکتری را درون آن تایپ نماید ، این کنترل لیستی از آیتم های پیشنهادی که با آن کاراکتر شروع می شوند ، را به کاربر نمایش می دهد .
در این مثال صفحه PHP به وسیله یک دستور Ajax ، با یک فایل آماده از عناوین ارتباط برقرار کرده و اطلاعات لازم را از آن دریافت می کند .

۱) کد تابع ( ) showHint :

هنگامی که کاربر در کادر متن موجود در صفحه PHP تایپ می کند ، به وسیله رویداد onkeyup ( ) آن ، تابع ( ) showHint اجرا می شود .
کد این تابع در جدول زیر نمایش داده شده است . کد را یکبار مرور نمایید . سپس به توضیح گام به گام هر بخش از مثال پرداخته شده است :

کد تابع ShowHint ( )	function showHint(str) { var xmlhttp; // ایجاد یک متغیر برای ذخیره کردن شی Ajax if (str.length==0) { document.getElementById("txtHint").innerHTML=""; return; } if (window.XMLHttpRequest) { // IE7+, Firefox, Chrome, Opera, Safari کد برای مرورگرهای xmlhttp=new XMLHttpRequest(); } else { // IE6, IE5 کد برای مرورگرهای xmlhttp=new ActiveXObject("Microsoft.XMLHTTP"); } xmlhttp.onreadystatechange=function() { if (xmlhttp.readyState==4 && xmlhttp.status==200) { document.getElementById("txtHint").innerHTML=xmlhttp.responseText; } }

```
    xmlhttp.open("GET","gethint.php?q="+str,true);
    xmlhttp.send( );
}
```

۲) توضیح گام به گام کد مثال :

هر یک از بخش های کد مثال که با یک رنگ خاص مشخص شده اند ، را توضیح می دهیم :

- این بخش کد مثال ، بررسی می کند اگر کادر متن خالی باشد (str.length==0) ، تابع مقدار عنصر txtHintموجود بر روی صفحه را پاک کرده و از تابع خارج می شود.
- این بخش از کد یک شی XMLHttpRequest را ساخته و عملیات Ajax را آغاز می کند . کد لازم برای پشتیبانی انواع مرورگرها را قرار داده ایم.
- این بخش از کد ، بررسی می کند که آیا پاسخ درخواست Ajax از سرور برگشته و آماده است یا خیر . سپس در صورت آماده بودن جواب ، مقدار آن را در عنصر txtHint موجود بر روی صفحه قرار می دهد .
- این بخش از کد ، شی xmlhttp ، را آماده کرده و به وسیله متد GET ، آن را به فایل مورد نظرمان یعنی gethint.phpارسال می کند .
همچنین توجه داشته باشید ، که متغیر str به وسیله پارامتر q به صفحه مقصد PHP ارسال شده است ، که بعدا در آن دریافت و استفاده خواهد شد.

۳) کد صفحه PHP مثل :

همانطور که در کد ایجکس در بخش اول مشاهده کردید ، کد ایجکس ، فایل gethint.php را در دستور خود فراخوانی و اجرا می کند .
این فایل ، به جستجوی نام های معادل با کاراکتر ورودی توسط کاربر پرداخته و نتیجه ای معادل را بر می گرداند .
در صورتی که هم ، معادلی برای مقدار ورودی پیدا نشود ، ، عبارت " no suggestion " به معنای بدون نتیجه صادر می شود .
کد فایل php در جدول زیر تشریح شده است :

کد فایل PHP

```
< ?php
    // لیست نام های مثال
    $a[ ]="Anna";
    $a[ ]="Brittany";
    $a[ ]="Cinderella";
    $a[ ]="Diana";
    $a[ ]="Eva";
    $a[ ]="Fiona";
    $a[ ]="Gunda";
    $a[ ]="Hege";
    $a[ ]="Inga";
    $a[ ]="Johanna";
    $a[ ]="Kitty";
```

```
$a[ ]="Linda";
$a[ ]="Nina";
$a[ ]="Ophelia";
$a[ ]="Petunia";
$a[ ]="Amanda";
$a[ ]="Raquel";
$a[ ]="Cindy";
$a[ ]="Doris";
$a[ ]="Eve";
$a[ ]="Evita";
$a[ ]="Sunniva";
$a[ ]="Tove";
$a[ ]="Unni";
$a[ ]="Violet";
$a[ ]="Liza";
$a[ ]="Elizabeth";
$a[ ]="Ellen";
$a[ ]="Wenche";
$a[ ]="Vicky";

// دریافت پارامتر q از صفحه مبدا
$q = $_GET["q"];

// جستجو برای نام مورد نظر در صورت وارد کردن مقداری از سوی کاربر
if (strlen($q) > 0)
    {
        $hint="";
        for( $i=0 ; $i < count($a) ; $ i++ )
            {
                if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q))))
                    {
                        if ($hint==" ")
                            {
                                $hint=$a[$i];
                            }
                        else
                            {
                                $hint=$hint." , ".$a[$i];
                            }
                    }
            }
    }
```

	`` // اگر نتیجه ای پیدا نشد ، عبارت بدون نتیجه تعیین شود // یا اینکه جواب خروجی ارسال شود if ($hint == "") { $response="no suggestion"; } else { $response=$hint; } // ارسال نتیجه خروجی به صفحه مبدا - دستور ایجکس echo $response; ?> ```
خروجی	مشاهده خروجی مثال

آموزش Ajax > مثال های عملی کار با Ajax > کاربرد Ajax در پایگاه داده

کاربرد Ajax در پایگاه داده :

در این بخش قصد داریم ، تا با ارائه یک مثال عملی و تشریح گام به گام آن ، نحوه استفاده و کار با دستورات Ajax ، برای فراخوانی و دریافت اطلاعات از یک پایگاه داده و سپس نمایش آنها ، بر روی یک صفحه PHP را نمایش دهیم .
در این مثال یک منوی کرکره ای قرار دارد که نام چندین مشتری در آن تعیین شده است . کاربر با انتخاب نام هر کدام از مشتریان ، به اطلاعات آن دسترسی پیدا خواهد کرد .
هنگامی که کاربر گزینه ای را در منوی کرکره ای انتخاب می کند ، مجموعه ای از دستورات Ajax و PHP اجرا شده و در پس زمینه صفحه ، به پایگاه داده متصل می شوند . سپس اطلاعات درخواستی را دریافت کرده و پس از انتقال به صفحه در بخش مورد نظر ، به روز رسانی می نماید .

۱) کد فایل html و تابع ( ) showUser :

هنگامی که کاربر گزینه ای را از منوی کرکره ای بر روی صفحه انتخاب می کند ، رویداد ( ) onchnage کنترل اجراشده و تابع ( ) showUser را فراخوانی می کند . این رویداد مقدار انتخاب شده در کنترل را به وسیله دستور this.value به پارامتر str در تابع ارسال می کند ، تا به تابع اعلام کند ، کاربر کدام گزینه را انتخاب کرده است .
جدول زیر ، کد فایل HTML و کد تابع ( ) showUser را نمایش می دهد . کد آن را مرور نمایید . سپس به توضیح گام به گام آن پرداخته ایم .

این پروژه دارای یک فایل PHP نیز می باشد که کد آن ، عملیات اتصال به پایگاه داده را انجام می دهد . برای مشاهده کد آن ، به بخش سوم مثال بروید .

کد صفحه HTML و کد تابع ShowUser ( )	<html> <head> <script> function showUser(str) { if (str == "") { document.getElementById("txtHint").innerHTML = ""; return; } if (window.XMLHttpRequest) { // IE7+, Firefox, Chrome, Opera, Safari کد لازم برای مرورگر های جدید xmlhttp = new XMLHttpRequest(); } else { // IE6, IE5 کد برای مرورگرهای قدیمی xmlhttp = new ActiveXObject("Microsoft.XMLHTTP"); } xmlhttp.onreadystatechange = function ( ) { if (xmlhttp.readyState == 4 && xmlhttp.status == 200) { document.getElementById("txtHint").innerHTML =

	xmlhttp.responseText; } } xmlhttp.open("GET", "getuser.php?q=" + str, true); xmlhttp.send ( ); } </script> </head> <body> <form> <select name="users" onchange="showUser(this.value)"> <option value="">Select a person:</option> <option value="1">Ali</option> <option value="2">Reza</option> <option value="3">Mohsen</option> </select> </form> اطلاعات مشتری: </body> </html>
خروجی	Select User اطلاعات مشتری :

۲) توضیح گام به گام کد مثال :

هر یک از بخش های کد مثال که با یک رنگ خاص مشخص شده اند ، را توضیح می دهیم :

- این بخش کد مثال ، بررسی می کند اگر کادر متن خالی نبوده و کاربر گزینه ای را انتخاب کرده باشد ، " (str == "، تابع مقدار عنصر txtHint موجود بر روی صفحه را پاک کرده و از تابع خارج می شود.
- این بخش از کد یک شی XMLHttpRequest را ساخته و عملیات Ajax را آغاز می کند . کد لازم برای پشتیبانی انواع مرورگرها را قرار داده ایم.
- این بخش از کد ، بررسی می کند که آیا پاسخ درخواست Ajax از سرور برگشته و آماده است یا خیر . سپس در صورت آماده بودن جواب ، مقدار آن را در عنصر txtHint موجود بر روی صفحه قرار می دهد .
- این بخش از کد ، شی xmlhttp ، را آماده کرده و به وسیله متد GET ، آن را به فایل مورد نظرمان یعنی getuser.php ارسال می کند .

همچنین توجه داشته باشید ، که متغیر str به وسیله پارامتر q به صفحه مقصد PHP ارسال شده است ، که بعدا در آن دریافت و استفاده خواهد شد.

- فرم HTML موجود بر روی صفحه نیز ، کنترل کرکره ای را نمایش می دهد ، که کاربر می تواند از آن انتخاب نماید.
- همچنین تگ div با id برابر txtHint ، پاسخ Ajax ارسالی را از سرور دریافت کرده و با یک عملیات به روز رسانی نمایش می دهد.

۳) نمایش و توضیح کد فایل PHP :

این پروژه دارای یک فایل PHP هم می باشد ، که عملیات اتصال و دریافت از پایگاه داده را انجام می دهد .
نام این فایل getuser.php است که دستور Ajax صفحه HTML ، فایل PHP را با پاس دادن متغیر str به آن ، فراخوانی و اجرا می کند .
این صفحه مقدار پارامتر str ، که همان گزینه انتخاب شده در منوی کرکره ای است را دریافت کرده و به پایگاه داده متصل می شود . سپس اطلاعات فرد انتخاب شده را از پایگاه داده استخراج کرده و به صفحه HTML بر می گرداند .

جدول زیر کد فایل PHP مثال را نمایش می دهد . آن را مرور نمایید . به توضیح بخش های مختلف آن پرداخته ایم :
(توضیحات کوتاه با //)

کد فایل PHP	< ?php $q=$_GET["q"]; // این دستور مقدار پارامتر ارسالی از دستور ای جکس را دریافت می کند $con = mysql_connect('localhost', 'Developer ' , 'abc123'); // ایج اد ارتباط با پایگاه داده مورد ظر mysql_select_db("ajax_demo", $con); // انتخاب جدول مورد ظر $sql="SELECT * FROM user WHERE id = '".$q."'"; // دریافت اطلاعات از q جدول مورد ظر بر مبنای پارامتر $result = mysql_query($sql); // ریختن نتیجه عملیات انتخاب اطلاعات از پایگاه result داده در متغیر echo "<table border='1'> <tr> <th>Firstname</th> <th>Lastname</th> <th>Age</th> </tr>"; while($row = mysql_fetch_array($result)) // ساخت جدول در خروجی با نتیجه دریافتی از پایگاه داده { echo "<tr>"; echo "<td>" . $row['FirstName'] . "</td>"; echo "<td>" . $row['LastName'] . "</td>";

```
                echo "<td>" . $row['Age'] . "</td>";
            echo "</tr>";
        }
    echo "</table>";

    mysql_close($con);   //      قطع ارتباط با پایگاه داده
?>
```

کاربرد Ajax در XML :

در این بخش قصد داریم ، تا با ارائه یک مثال عملی ، نحوه کار با فایل های XML در Ajax و دریافت اطلاعات از آنها را نمایش دهیم .
در این مثال یک دکمه فرمان قرار داده شده است ، که در هنگام کلیک کاربر بر روی آن ، اطلاعات چند CD را از یک فایل XML خوانده و سپس در جدول نمایش می دهد .

۱) کد تابع ( ) loadXMLDoc :

کد تابع ( ) loadXMLDoc که عملیات ساخت شی Ajax و دریافت اطلاعات از فایل XML را انجام می دهد ، به صورت زیر است . این تابع پارامتر url را به عنوان آدرس فایل XML در ورودی دریافت می کند . کد آن را مرور نمایید . سپس به تشریح خط به خط کد آن می پردازیم :

کد فایل تابع loadXMLDoc ( )

```
<!DOCTYPE html>

<html>
<head>
  <script>
    function loadXMLDoc(url) {                          //   شروع کد تابع
      var xmlhttp;                                      // Ajax  ایجاد متغیر لازم برای شی
      var txt, x, xx, i;                                // ایجاد سایر متغیر های لازم برای تابع
      if (window.XMLHttpRequest) {                      //  با این دستور ، تابع پشتیبانی
مرورگر از ایجکس را می سنجد
        xmlhttp = new XMLHttpRequest();                 // سپس یک شی جدید ایجکس ایجاد می کند
      }
      xmlhttp.onreadystatechange = function ( ) {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
        txt = "<table border='1'><tr><th>Title</th><th>Artist</th></tr>";
        x = xmlhttp.responseXML.documentElement.getElementsByTagName( "
CD " ) ;
        for (i = 0; i < x.length; i++) {
          txt = txt + "<tr>";
          xx = x[i].getElementsByTagName("TITLE");
            {
              try {
                  txt = txt + "<td>" + xx[0].firstChild.nodeValue + "</td>";
                }
              catch (er) {
                  txt = txt + "<td> </td>";
                }
```

```
                    }
                xx = x[i].getElementsByTagName("ARTIST");
                        {
                            try {
                                txt = txt + "<td>" + xx[0].firstChild.nodeValue +
"</td>";
                                }
                            catch (er) {
                                txt = txt + "<td> </td>";
                                }
                        }
                    txt = txt + "</tr>";
                }
                txt = txt + "</table>";
                document.getElementById('txtCDInfo').innerHTML = txt;
                }
            }
        xmlhttp.open("GET", url, true);        // باز کردن فایل مقصد و خواندن اطلاعات
        xmlhttp.send( );
    }
</script>
</head>

<body>
<div id="txtCDInfo">
   <button onclick="loadXMLDoc('cd_catalog.xml')"> مشاهده اطلاعات CD ها
</button>        //     دکمه فرمان فراخوانی تابع
</div>
</body>
</html>
```

۲) توضیح گام به گام کد مثال :

هر یک از بخش های کد مثال که با یک رنگ خاص مشخص شده اند ، را توضیح می دهیم :

- این بخش کد تابعی است ، که عملیات ساخت شی Ajax وخواندن اطلاعات از فایل XML را انجام می دهد .
- این بخش از کد بررسی می کند که آیا وضعیت شی ساخته شده Ajax ، آماده بوده و جواب لازم نیز از سرور برگشت داده شده است یا خیر.
- این بخش از کد ، بر پایه اطلاعات دریافتی از فایل XML ، جدول لازم را برای نمایش اطلاعات ساخته و آن را در تگ قرار می دهد.

- این بخش از کد ، دکمه فرمانی است که تابع loadXMLDoc ( ) را فراخوانی کرده و آدرس فایل XML را به آن ارسال می کند.
همچنین شامل تگ ای است که ، محتویات اطلاعات خروجی از فایل XML را در خود جای می دهد.

۳) مشاهده خروجی مثال :

برای مشاهده خروجی مثال ، بر روی دکمه فرمان زیر کلیک نمایید . همچنین کد فایل XML نیز در آخر صفحه به شما نمایش داده شده است.

خروجی مثال	مشاهده اطلاعات CD ها

۴) کد فایل XML :

کد فایل XML مورد استفاده در مثال :

کد فایل XML

```
<?xml version="1.0" encoding="utf-8" ?>
<catalog>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <PRICE>9.90</PRICE>
    <YEAR>1982</YEAR>
  </CD>
</catalog>
```

مرور کلی بر تکنولوژی مایکروسافت Ajax :

بوسیله استفاده از تکنولوژی Ajax در صفحات ASP.Net شما می توانید برنامه ها و صفحاتی به مراتب سریعتر و با قدرت پاسخگویی بالاتر تولید نمایید . ASP.Net دارای چندین کنترل پیش ساخته Ajax می باشد ، که قابلیت های بسیار زیادی را در اختیار کاربر قرار می دهد . فواید و امکانات استفاده از تکنولوژی Ajax در صفحات ASP.Net عبارتند از :

- نمایش رابط های کاربری گرافیکی متداول مثل نمایش دهنده های میزان پیشرفت پروسه ، پیام های راهنما یا منوهای POP-UP.
- آپدیت و به روز رسانی فقط بخشی از صفحه که محتویات آن تغییر کرده ، به جای رفرش کردن کل صفحه.
- ارتباط کلاینت سایدی با سرویس های Asp.Nt برای هویت شناسی در فرم های ورد و خروج کاربران در سایت.
- اضافه کردن قابلیت کارکرد اسکریپتی برای کنترل های سرور ساید Asp.Net.
- پردازش و اعتبارسنجی کلاینت سایدی (به جای سرور ساید) محتویات وارد شده در فرم های وب.

معرفی کنترل های Ajax مایکروسافت ASP.Net :

ASP.Net دارای چندین کنترل پیش ساخته Ajax است که قابلیت ها و امکانات این تکنولوژی جدید را بدون نیاز به کدنویسی خاصی در اختیار کاربر قرار می دهد . لیست زیر کنترل های Ajax در ASP.Net را بصورت مختصر معرفی می کند . برای دریافت اطلاعات بیشتر بر روی نام آنها کلیک نمایید :

- **ScriptManager** : این کنترل وظیفه پردازش و مرتب سازی اسکریپت های کلاینتی صفحه را به عهده دارد . عملیات های Ajax بر اساس اسکریپت ها در صفحه اجرا می شوند . این کنترل اسکریپت ها را مدیریت و اجرا می کند.
- **UpdatePanel** : این کنترل بخش محتویات کدی که درون آن قرار داشته را در صفحه آپدیت و به روز رسانی می کند و مانع رفرش شدن کد صفحه می شود.
- **UpdateProgress** : این کنترل می تواند عملیات به روز رسانی و آپدیت اطلاعات در صفحه را در قالب یک نمایه گرافیکی یا پیام نمایش دهد.
- **Timer** : بوسیله این کنترل می توانید در فواصل زمانی تعیین شده صفحه را PostBack کرده و یا با اتصال آن به یک کنترل **UpdatePanel** کد آن را اجرا نمایید.

ساخت یک سایت ASP.Net با قابلیت ایجکس :

در این راهکار یک سایت ساده ASP.Net را خواهیم ساخت ، که شامل صفحه ای است که برخی از قابلیت های مایکروسافت Ajax را نمایش خواهد داد .
ابن برنامه اطلاعات دانشجویان را از یک پایگاه داده فرضی دریافت کرده و بر روی صفحه نمایش می دهد .
برنامه از یک کنترل UpdatePanel ، برای آپدیت و به روز رسانی بخشی از صفحه که تغییر می کند استفاده کرده و مانع رفرش و Postback شدن کامل صفحه در هنگام دریافت یا ارسال اطلاعات می شود .
همچنین این برنامه از یک کنترل UpdateProgress برای نمایش میزان پیشرفت عملیات انتقال اطلاعات ، استفاده می کند .

۱) اضافه کردن یک کنترل UpdatePanel به صفحه ASP.Net :

پس از اینکه یک سایت ASP.Net را ساختید (برای دریافت اطلاعات بیشتر به راهکار ساخت یک سایت ساده در ASP.Net بروید) ، صفحه ای را به آن اضافه نمایید که قرار است کنترل UpdatePanel را بر روی خود نگه دارد . قبل از اینکه یک کنترل UpdatePanel را به صفحه اضافه نمایید ، بایستی یک کنترل ScriptManger را نیز بر روی صفحه قرار دهید . کنترل UpdatePanel برای انجام امور آپدیت و دریافت و ارسال اطلاعات از قابلیت های کنترلScriptManger استفاده می کند .

نحوه ساخت یک صفحه جدید ASP.Net :

برای شروع پروژه باید یک صفحه جدید ASP.Net را به سایت خود اصافه نماییم . برای این منظور مراحل زیر را انجام دهید:

۱. از منوی Solution Explorer ، بر روی نام سایت کلیک سمت راست کرده و گزینه Add New Item را انتخاب نمایید.
۲. از منوی باز شده ، نوع فایل را Web Form انتخاب کرده و سپس نام آن را به Students.aspx تغییر دهید . همچنین تیک گزینه Place code in separate file را بردارید (تا فایل کد جداگانه ای برای صفحه ایجاد نشود.)
۳. با زدن دکمه Add ، صفحه جدید را ایجاد کرده و به حالت Design بروید.
٤. از منوی AJAX Extensions یک کنترل ScriptManger و یک کنترل UpdatePanel را بر روی صفحه قرار دهید.

۲) اضافه کردن محتویات به یک کنترل UpdatePanel :

کنترل UpdatePanel فقط بخشی از صفحه که قرار است ، تغییر کند را به روز رسانی کرده و کاری به بقیه صفحه ندارد . در این بخش از راهکار ، کنترل داده ای را به صفحه اضافه خواهیم کرد ، که قرار است اطلاعات را از پایگاه داده Students دریافت کرده و بر روی صفحه نمایش دهد . برای اضافه کردن محتویات جهت بروز رسانی بر روی صفحه به کنترل UpdatePanel ، مراحل زیر را انجام دهید :

۱. از بخش Data منوی Toolbox برنامه ، یک کنترل GridView را انتخاب کرده و در بخش قابل ویرایش کنترل UpdatePanel قرار دهید.
۲. از منوی وظایف Tasks کنترل GridView که به صورت یک فلش بر روی آن قرار دارد ، کادر کشویی Choose Data Sourceرا کیلک کرده و سپس گزینه <New data source>را انتخاب نمایید . پنجره Data Source Configuration باز می شود.
۳. در قسمت Where will the application get data from ، گزینه Database را انتخاب کرده و سپس دکمه Ok را بزنید.
٤. در بخش Configure Data Source ، به پایگاه داده Students متصل شده و سپس گزینه Next را بزنید.
٥. از مرحله Configure the Select Statement ، گزینه Specify a custom SQL statement or stored procedureرا انتخاب کرده و مجددا Next را بزنید.
٦. در بخش Select قسمت Define Custom Statement or Stored Procedures ، دستور Selectزیر را وارد نمایید :

SELECT FirstName, LastName From Students	دستور Select

۷. سپس کلید Next و بعد از آن Ok را بزنید.
۸. همچنین در بخش GridView Tasks ، گزینه Enable Paging را علامت بزنید تا این کنترل اطلاعات خود را صفحه بندی نماید.
۹. تغییرات خود را ذخیره نموده و سپس کنترل های Ctrl + F5 را برای اجرا صفحه فشار دهید.

همانطور که در خروجی صفحه مشاهده می کنید ، در هنگام تغییر شماره صفحه در کنترل GridView و به روز رسانی اطلاعات نمایش داده شده آن ، صفحه ASP.Net رفرش یا Postback نشده و فقط بخش جدول کنترل GridView به روز رسانی می شود .
در صورتی که در حالت استفاده نکردن از کنترل UpdatePanel و تکنولوژی ایجکس ، با هر بار تغییر اطلاعات کنترل GridView ، صفحه مجبور به Postback شدن و بار گذاری مجدد می شود .

اضافه کردن یک کنترل UpdateProgress به صفحه :

کنترل UpdateProgress ، یک پیام یا نمایه وضعیت را زمانی که اطلاعات و محتویات در حال لود شدن برای کنترل GridView است ، نمایش می دهد .
برای مثال نمونه آن را که در در اکثر سایت ها دیده اید ، ساعت یا دایره رنگی است که تا زمان آماده شدن اطلاعات در حال چرخش است .

نحوه اضافه کردن یک کنترل UpdateProgress به صفحه :

۱. از قسمت Ajax Extensions نوار ابزار Toolbox برنامه ، یک کنترل UpdateProgress را انتخاب کرده و در زیر کنترل UpdatePanel قرار دهید.
۲. کنترل UpdateProgress را انتخاب کرده و از قسمت Properties آن ، مقدار خاصیت AssociatedUpdatePanelIDآن را به UpdatePanel1 تغییر دهید .
این کار کنترل UpdateProgress را به کنترل UpdatePanel که از قبل بر روی صفحه قرار داده بودید ، متصل می کند.

۳. در قسمت قایل ویرایش کنترل UpdateProgress ، بنویسید **" در حال دریافت اطلاعات. " ...**
٤. تغییرات ایجاد کرده را ذخیره کرده و برای اجرای صفحه Ctrl + F5 را بزنید.

اگر تاخیر یا مدت زمانی برای دریافت اطلاعات توسط کنترل داده GridView و به روز رسانی آنها بر روی صفحه وجود داشته باشد ، در این فرصت کنترل UpdateProgress ، متن خود را به کاربر نمایش می دهد ، تا عملیات تمام شود .

اضافه کردن تاخیر در اجرای کدها در صفحه ایجکس ASP.Net :

اگر در مثال ارائه شده در صفحه ، کنترل داده عملیات دریافت و به روز رسانی اطلاعات را با سرعت بسیار زیادی انجام دهد ، ممکن است کاربر هیچ گاه پیام یا نمایه کنترل UpdateProgress را مشاهده نکند .
عدم مشاهده این پیام ، ممکن است کاربر را به این اشتباه بیاندازد که اطلاعات مورد نظر در صفحه آپدیت و به روز رسانی نشده اند .
کنترل UpdateProgress ، از یک خاصیت به نام DisplayAfter پشتیبانی می کند ، که این قابلیت را به طراح می دهد تا تاخیری را برای اجرا و نمایش پیام کنترل تعیین نمایید .
تعیین این خاصیت مانع از فلش زدن و نمایش بسیار سریع پیام کنترل در زمانی که عملیات دریافت و به روز رسانی اطلاعات از سرور بسیار سریع است ، می شود .
به طور پیش فرض این مقدار برای کنترل UpdateProgress ، نیم ثانیه (۵۰۰ میلی ثانیه) بوده و به این معناست که کنترل UpdateProgress ، چنانچه عملیات ارسال و به روز رسانی اطلاعات کمتر از نیم ثانیه طول بکشد ، نمایش داده نخواهد شد .
در محیط برنامه خودتان می توانید یک زمان تاخیر را به کنترل UpdateProgress اضافه نمایید ، تا مطمئن شوید که این کنترل ، همانطور که انتظار دارید ، عمل کرده و حتما پیام خود را نمایش دهد . این کار یک کار کاملا اختیاری است و استفاده از آن بر طبق سلیقه طراح خواهد بود .

نحوه اضافه کردن تاخیر به برنامه خودتان :

۱. درون کنترل UpdateProgress ، کنترل GridView را انتخاب نمایید.
۲. در قسمت Properties آن ، دکمه Event را کلیک نمایید.
۳. رویداد PageIndexChanged کنترل را دابل کلیک نمایید ، تا صفحه کد پشت صحنه و رویداد مربوطه برای کد نویسی باز شود.
٤. کد زیر را به event handler رویداد PageIndexChanged کنترل GridView ، اضافه نمایید تا تاخیری ۳ ثانیه را به پروسه پردازشی کد ، اضافه نماید :

کد	// کد برای C# System.Threading.Thread.Sleep(3000); // کد برای VB System.Threading.Thread.Sleep(3000)

۵. تغییرات را ذخیره کرده و مجددا صفحه را اجرا نمایید.
٦. با کار با کنترل GridView و مشاهده تفاوت با حالت قبل ، کاربرد تاخیر را در پروسه صفحه مشاهده نمایید.

آموزش Ajax > مایکروسافت Ajax > ساخت یک برنامه داده ای با قابلیت ایجکس

ساخت یک برنامه داده ای با قابلیت Ajax :

در این راهکار شما با نحوه ایجاد یک برنامه وب با قابلیت Ajax ، که برای نمایش لیستی از آیتم ها و کارهای در حال اجرا به کار می رود ، آشنا خواهید شد .
این برنامه به شما کمک می کند تا یک رابط کاربری را برای ایجاد ، مدیریت ، پاک کردن لیست ها و آیتم های موجود در آنها بسازید . تمامی اعمال به روز رسانی، وارد کردن ، پاک کردن و صفحه بندی اطلاعات درون یک کنترلUpadatePanel انجام می شود ، که از تکنولوژی Ajax استفاده می کند .
کارهایی که در این راهکار به شما آموزش داده خواهد شد ، عبارتند از :

- ایجاد یک پایگاه داده SQL و اضافه کردن اطلاعات.
- اضافه کردن یک کنترل LinqDataSource به صفحه.
- اضافه کردن کلاس LINQ به کلاس های. SQL
- استفاده از کنترل ListView برای نمایش ، ویرایش و پاک کردن اطلاعات در پایگاه داده.
- استفاده از کنترل LinqDataSource برای اتصال به پایگاه داده با استفاده از زبان. LINQ
- استفاده از کنترل UpdatePannel برای اضافه کردن قابلیت های Ajax به صفحه.

نرم افزار های پیش نیاز :

برای اجرای این راهکار بایستی نرم افزار های زیر بر روی سیستم شما نصب شده باشند :

- نرم افزار Microsoft Visual Studio or Visual Web Developer 2010 Express یا ورژن های مشابه.
- SQL Server Expressکه همراه با ویژوال استودیو نصب می شود.

ایجاد یک وب سایت جدید ASP.Net :

در مرحله اول ، باید سایت ASP.Net خود را ایجاد نمایید . فرض بر این است که کاربر آشنایی کافی برای ایجاد یک وب سایت جدید ASP.Net را دارد . اگر هم با این کار آشنا نیستید ، برای دریافت اطلاعات بیشتر به راهکار ساخت یک وب سایت ساده در ASP.Net بروید .

ایجاد یک پایگاه داده جدید SQL Server :

اکنون که سایت جدید ASP.Net خود را ساخته اید ، قدم بعدی ایجاد یک پایگاه داده و اضافه کردن یک مسیر دسترسی به آن در Server Explorer است . هنگامی که یک پایگاه داده را به Server Explorer ، اضافه می کنید ، می توانید به راحتی از نرم افزار ویژوال استودیو برای اضافه کردن جداول ، stored procedures، views و ... به آن ، استفاده نمایید .
همچنین می توانید از طریق این ابزار اطلاعات موجود در جداول پایگاه داده را مشاهده کرده و یا اینکه Query مورد نظر خود را به راحتی با استفاده از پنجره Query Builder ایجاد نمایید .

نحوه اضافه کردن یک Database به پروژه :

۱. در منوی Solution Explorer ، بر روی نام وب سایت کلیک راست کرده و گزینه Add New Item را انتخاب نمایید.
۲. از کادر باز شده ، گزینه SQL Database را انتخاب کرده و نام آن را به Tasks.mdf تغییر داده و گزینه Ok را بزنید.
۳. هنگامی که نرم افزار ویژوال استودیو از شما پرسید ، که Database باید در پوشه App_Data نگهداری شود ، گزینه Yes را انتخاب نمایید.

طراحی نمایه کلی Database و اضافه کردن اطلاعات اولیه به آن :

می توانید از ابزارها و امکانات ویژوال استودیو برای طراحی نمایه کلی و وارد نمودن اطلاعات اولیه به پایگاه داده خود استفاده نمایید . برای این منظور مراحل زیر را انجام دهید :

۱. در منوی Solution Explorer ، پوشه App_Data را باز کرده و بر روی پایگاه داده Tasks.mdf ، دابل کلیک نمایید .
برنامه پایگاه داده را به صورت درختی در منوی Server Explorer باز می کند.
۲. بر روی پوشه Tables کلیک راست کرده و گزینه Add New Item را انتخاب نمایید.
۳. در ویرایشگر جدول پایگاه داده ، فیلدهای زیر را با مشخصات داده شده ایجاد نمایید :

نام فیلد یا ستون	نوع داده ای	آیا می تواند خالی باشد ؟
taskId	int	Allow Nulls: No
taskName	nvarchar(50)	Allow Nulls: No
dateCreated	datetime	Allow Nulls: No
isComplete	bit	Allow Nulls: No

٤.
٥. بر روی ردیفی که شامل فیلد taskid می باشد ، کلیک راست کرده و با انتخاب گزینه Set Primary Key، آن فیلد را به عنوان فیلد اصلی انتخاب نمایید.
٦. همانطور که ردیف taskid را در حال انتخاب دارید ، در بخش Columnes Properties ، بخش Identity Specificationرا باز کرده و مقدار گزینه Is Identity را به Yes تغییر دهید.
۷. جدول را ذخیره کرده و نام آن را TaskList قرار دهید.
۸. بر روی نام جدول در Server Explorer کلیک راست کرده و گزینه Show Table Data را انتخاب نمایید .
پنجره ای باز می شود که در آن می توانید اطلاعات جدول را مشاهده کرده و یا اطلاعات مورد نظر خود را به آن بیفزایید.
۹. چهار یا پنج رکورد را با اطلاعات دلخواه برای نمایش در صفحه ، به جدول اضافه کرده و سپس آن را ببندید .
دقت داشته باشید که لزومی ندارد برای فیلد taskid ، مقداری تعیین نمایید ، زیرا آن فیلد به صورت

اتوماتیک مقداردهی می شود .
همچنین برای فیلد isComplete ، باید مقدار True یا False را وارد نمایید.

ایجاد کنترل های دسترسی و کار با داده در ASP.Net :

در این بخش ، از یک کنترل LinqDataSource استفاده خواهید کرد و به وسیله آن کلاس هایی را طراحی می کنید ، که به جای موجودیت های مختلف پایگاه داده (مثل جدول ، خود پایگاه داده و ...) عمل خواهند کرد .
کنترل LinqDataSource و کلاس های طراحی شده برای آن ، لایه دسترسی به داده ها هستند ، که در این راهکار باید از آنها استفاده نمایید .
کنترل LinqDataSource ، زبان LINQ را به توسعه دهندگان وب از طریق سیستم دیتای ASP.Net ، معرفی کرد . کنترل LinqDataSource کدهای لازم برای انتخاب کردن ، وارد نمودن ، به روز رسانی و یا حذف اشیا و اطلاعات را در پایگاه داده ایجاد می کند . LINQ ویژگی های برنامه نویسی شی گرا را به زبان پایگاه داده اضافه کرده است .
این زبان یک مدل برنامه نویسی منحصر به فرد را برای جستجو و به روز رسانی اطلاعات برای منابع داده ای مختلف فراهم کرده و قابلیت های خود را مستقیما به C# یا VB اضافه می کند .

اتصال پایگاه داده Tasks به یک شی داده ای SQL :

برای شروع ساخت لایه داده ای برنامه ، باید یک شی dataset را به پروژه اضافه نماییم .

نحوه ایجاد یک کلاس برای جدول TaskList :

۱. اگر وب سایتی که ایجاد نموده اید ، از قبل دارای پوشه App_Code نیست ، در منوی Solution Explorer، بر روی نام وب سایت کلیک راست کرده و گزینه Add ASP.Net Folder را انتخاب نموده و سپس پوشه APP_Code را اضافه نمایید.
۲. بر روی پوشه App_Code کلیک راست کرده و گزینه Add New Item را انتخاب نمایید.
۳. از پنجره بازشده ، نوع فایل LINQ to SQL Classes را انتخاب کرده و نام آن را به Tasks.dbml تغییر داده و سپس گزینه Add را بزنید.
٤. از منوی Server Explorer ، جدول TaskList را درگ کرده و آن را بر روی پنجره Object Relational Designerبندازید.
٥. فایل Tasks.dbml را ذخیره نمایید .
هنگامی که فایل فوق را ذخیره می نمایید ، ویژوال استودیو به صورت اتوماتیک دو فایل را در پوشه App_Codeو در زیر فایل Tasks.dbml اضافه می کند . فایل اول Tasks.dbml.layout بوده و فایل دوم Tasks.designer.cs یا Tasks.designer.vb می باشد ، بر حسب زبان برنامه نویسی که برای پروژه خود انتخاب کرده اید.
٦. در منوی Solution Explorer ، فایل Tasks.designer.cs یا Tasks.designer.vb را باز کنید .
اگر به آن دقت نمایید ، مشاهده می کنید که دارای ۲ کلاس به نام های TasksDataContext و TasksListمی باشد . کلاس TasksDataContext به جای خود پایگاه داده و کلاس TasksList به جای جدول آن عمل می کند .
کلاس TasksDataContext که فاقد هر گونه پارامتر است ، اطلاعات اتصال به پایگاه داده (Connection Stringرا از فایل web.config می خواند.

۷. فایل web.config را باز کنید .
دقت نمایید که یک رشته ارتباطی (Connection String) برای پایگاه داده Tasks.mdf به المنت connectionStringsاضافه شده است.
۸. تمامی فایل های که باز کرده را ببندید.

ایجاد و تنظیم کردن کنترل LinqDataSource :

اکنون که شما یک جدول پایگاه داده و کلاس هایی که جانشین اجزای درونی آن هستند ، را در اختیار دارید ، می توانید از یک کنترل ListView برای اتصال به Database بر روی صفحات ASP.Net استفاده کنید . برای این **منظور** مراحل زیر را انجام دهید :

۱. صفحه Default.aspx سایت را باز کرده و به نمای Design بروید.
۲. یک کنترل LinqDataSource را از بخش کنترل های داده منوی Toolbox درگ کرده و بر روی صفحه قرار دهید Id . آن را نیز برابر با LinqDataSource1 قرار دهید.
۳. **از بخش** LinqDataSource Tasks ، که به صورت یک فلش بر روی کنترل LinqDataSource مشاهده می شود ، گزینه Configure Data Source را انتخاب نمایید.
٤. **از بخش** Choose your context object ، گزینه TasksDataContext را انتخاب کرده و گزینه Nextرا بزنید.
٥. از لیست Table ، گزینه TasksLists (Table) را انتخاب کرده و سپس دکمه Finish را بزنید.
٦. **مجددا از قسمت** LinqDataSource Tasks ، گزینه های Enable Delete , Enable Insert **و** Enable Updateرا علامت بزنید .
توجه داشته باشید که شما نیازی به هیچ گونه کدنویسی برای انجام این عملیات ها نداشته و خود کنترل ها آنها را برایتان انجام می دهند.
۷. صفحه را ذخیره نمایید.

استفاده از کنترل های سرور داده :

در این قسمت ، شما کنترل هایی را به صفحه اضافه خواهید کرد که از کنترل LinqDataSource و کلاس های آن ، برای اتصال به پایگاه داده و نمایش اطلاعات مورد نظر استفاده خواهند کرد .
برای این منظور ، یک کنترل ListView را برای نمایش اطلاعات از پایگاه داده SQL اضافه خواهیم کرد .**سپس** یک کنترل DropDownList را برای فیلتر کردن و گزینش اطلاعات خروجی در کنترل ListView به کار خواهید برد .
در نهایت نیز از یک کنترل UpdatePannel برای افزودن قابلیت های Ajax و به روز رسانی فقط بخش مورد نظرتان در صفحه به جای رفرش شدن کل آن ، استفاده خواهید نمود .

نمایش اطلاعات توسط یک کنترل ListView :

کنترل ListView ، برای نمایش اطلاعات ساختار بندی شده، همانند کنترل های Repeater **و** DataList **بسیار مناسب است** . همچنین این کنترل بر خلاف دو کنترل ذکر شده ، از قابلیت های ویرایش ، وارد نمودن ، پاک کردن ، صفحه بندی و مرتب سازی اطلاعات به راحتی پشتیبانی می کند .
اکنون شما یک کنترل ListView را به صفحه اضافه خواهید کرد ، که تمامی اطلاعات جدول Tasks **را** نمایش خواهد داد. در مرحله بعدی یک کنترل DropDownList را نیز به صفحه اضافه خواهیم کرد ، تا فقط اطلاعات مورد نظر خود را نشان داده و بقیه را فیلتر نماید .
کنترل ListView ، اطلاعات جدول بانک اطلاعاتی را در جداول مرتب و با ویرایش کاربر پسند نمایش داده و

همچنین دارای دکمه هایی برای ویرایش ، حذف ، به روز رسانی و یا وارد نمودن اطلاعات جدید به پایگاه داده است .
برای اضافه کردن یک کنترل ListView به صفحه ، مراحل زیر را انجام دهید :

۱. به صفحه ای که می خواهید کنترل ListView را به آن اضافه نمایید ، بروید.
۲. از بخش Data منوی Toolbox ، یک کنترل ListView را برداشته و بر روی صفحه قرار دهید.
۳. از منوی ListView Tasks ، که به صورت یک فلش بر روی کنترل دیده می شود ، کادر Choose Data Source list را انتخاب کرده و گزینه LinqDataSource1 را کلیک نمایید .
این کار ، کنترل ListView را به کنترل LinqDataSource متصل می کند.
٤. مجددا در منوی ListView Tasks ، بر روی گزینه Configure ListView کلیک نمایید .
سپس از پنجره باز شده ، گزینه های Enable Editing, Enable Inserting, Enable Deleting و Enable Paging را علامت بزنید.
٥. دکمه Ok را زده و صفحه را ذخیره نمایید.

اضافه کردن یک کنترل DropDownList برای فیلتر کردن اطلاعات :

شما می توانید ، اطلاعاتی که توسط کنترل ListView نمایش داده می شود را به وسیله قرار دادن یک کادر انتخابی DropDownList ، فیلتر کرده و در هر لحظه ، اطلاعات مورد نظر خود را بر حسب گزینه ای که کاربر در لیست انتخاب کرده ، تغییر دهید .
برای اضافه کردن یک کنترل DropDownList به صفحه مراحل زیر را انجام دهید :

۱. صفحه Default.aspx را باز کرده و به نمای source کد بروید.
۲. درون تگ <form> صفحه و در بالای کد کنترل ListView ، کد زیر را قرار دهید :

کد	 Current List Filter: <asp:DropDownList ID="DropDownList1" AutoPostBack="true" runat="server"> <asp:ListItem Text="Active" Value="False" /> <asp:ListItem Text="Completed" Value="True" /> </asp:DropDownList> <hr id="separator" />

۳. درون کد کنترل LinqDataSource ، مقدار خاصیت AutoGenerateWhereClause آن را بر روی مقدار true تنظیم نمایید.
٤. کد زیر را درون تگ باز و بسته کنترل LinqDataSource اضافه نمایید . این کد تعیین می کند که در هر لحظه چه اطلاعاتی نمایش داده شود . به آن دقت نمایید :

کد	<WhereParameters> <asp:ControlParameter Name="isComplete" ControlID="DropDownList1" Type="Boolean" />

	</WhereParameters>

۵. صفحه را ذخیره نمایید.

اکنون شما می توانید صفحه را تست کرده و مطمئن شوید که اطلاعات مورد نظرتان بر روی صفحه نمایش داده خواهد شد .
برای تست صفحه کلیدهای Ctrl + F5 را همزمان فشار دهید .
سپس از کنترل DropDownList ، گزینه Completed را انتخاب نمایید . مشاهده خواهید کرد فقط اطلاعات ، رکوردهای از جدول Tasks در کنترل ListView نمایش داده می شود که مقدار فیلد isComplete آنها برابر با True است .

اضافه کردن قابلیت Ajax به صفحه :

در این بخش از راهکار ، که بخش پایانی است ، یک کنترل ScriptManager را به صفحه اضافه خواهیم کرد ، که قابلیت های Ajax را بر روی صفحات ASP.Net فعال می کند .
سپس نیز یک کنترل UpdatePannel را بر روی صفحه قرار خواهیم داد ، که این امکان را به ما می دهد تا به روز رسانی و تغییر اطلاعات در کنترل ListView ، بدون Postback شدن و رفرش کل صفحه انجام گیرد .

اضافه کردن یک کنترل ScriptManager به صفحه :

برای استفاده از هر گونه قابلیت Ajax و کنترل های مرتبط با آن بر روی صفحات ASP.Net ، بایستی یک کنترل ScriptManager را به روش زیر ، بر روی صفحه قرار دهید :

۱. صفحه Default.aspx را باز کرده و به نمایه source کد بروید.
۲. از بخش AJAX Extensions منوی Toolbox ، یک کنترل ScriptManager را انتخاب کرده و در میان تگ <form>قرار دهید.

قرار دادن کنترل ListView درون کنترل UpdatePannel :

در این مثال ، کنترل ListView را درون کنترل UpdatePannel قرار می دهیم . انجام این کار باعث می شود تا انجام هر گونه تغییر و به روز رسانی در کنترل ListView ، بدون لود شدن و رفرش کل صفحه انجام شود .
برای این منظور مراحل زیر را انجام دهید :

۱. در صفحه Default.aspx ، کد زیر را مستقیما پس از تگ ابتدایی <form>قرار دهید :

کد	<asp:UpdatePanel ID="UpdatePanel1" runat="server"> <ContentTemplate>

۲. همچنین کد زیر نیز دقیقا قبل از تگ انتهایی </form>قرار دهید :

کد	</ContentTemplate> </asp:UpdatePanel>

این کار باعث می شود تا کنترل های ListView و DropDownList درون کنترل UpdatePannel قرار بگیرند .

مرحله نهایی :

پس از ذخیره صفحه ، برای اجرای آن ، کنترل های Ctrl + F5 را همزمان فشار دهید .
سپس آیتم های موجود در کنترل DropDownList را تغییر دهید . مشاهده می کنید که بدون رفرش شدن صفحه ، اطلاعات کنترل ListView ، به روز رسانی می شود .

آموزش Ajax > کنترل های ASP.Net Ajax > مرور کلی بر کنترل های ASP.Net در Ajax

معرفی کنترل های ASP.Net در Ajax :

در این بخش به معرفی کنترل های ASP.Net در Ajax که امکان اضافه کردن قابلیت های Ajax به صفحات وب را به شما می دهند ، می پردازیم .
همانطور که در بخش های قبل با مفهوم کاربرد Ajax آشنا شدید ، این تکنیک امکان به روز رسانی و آپدیت بخشی از یک صفحه وب را بدون رفرش شدن و بارگذاری مجدد کل صفحه را فراهم می نماید .
در لیست زیر ، کنترل های ASP.Net در Ajax معرفی شده اند . برای دریافت اطلاعات بیشتر راجع به هر کدام بر روی نام آن کلیک نمایید :

- **کنترل : ScriptManager :** کنترل ScriptManager ، وظیفه مدیریت اسکریپت ها در صفحات ASP.Netبا قابلیت Ajax را دارد و به طور پیش فرض ، کنترل ScriptManager ، اسکریپت فراخوانی شده در صفحه را در مجموعه اسکریپت های Ajax صفحه ، ثبت و اجرا می کند .
به عبارت دیگر بار کلی فراخوانی ، اجرا و در نهایت اعمال تغییرات و آپدیت مورد نظر در صفحه ، بر عهده کنترل ScriptManager می باشد.
- **کنترل : Timer :** از کنترل Timer برای اجرای دستورات مورد نظر در بازه های زمانی معین و به صورت متناوب استفاده می شود . اگر این کنترل را با کنترل UpdatePanel به کار ببرید ، می توانید در بازه های زمانی معین ، یک بخش از صفحه را به صورت متناوب ، به روز رسانی نمایید . همچنین از این کنترل می توانید برای postback کردن کامل صفحه استفاده نمایید.
- **کنترل : UpdatePanel :** کنترل UpdatePanel ، این امکان را به شما می دهد تا فقط بخشی از یک صفحه وب را که می خواهید به روز رسانی شده و تغییر کند را آپدیت کرده ، به جای اینکه کل صفحه را رفرش نمایید .
با به کار بردن کنترل UpdatePanel به همراه کنترل ScriptManager در یک صفحه ASP.Net ، برای آپدیت و به روز رسانی اطلاعات در یک صفحه ، دیگر نیاز به نوشتن هیچ کد یا اسکریپت اضافه ای نخواهید داشت.
- **کنترل : UpdateProgress :** به وسیله کنترل UpdateProgress می توانید ، در یک نمایه تصویری یا معمولی ، پروسه و میزان پیشرفت عملیات اجرا و به روز رسانی اطلاعات در صفحه را ، به کاربر نمایش دهید.

آموزش Ajax > کنترل های Ajax ASP.Net > معرفی کنترل ScriptManager

آشنایی با کنترل ScriptManager :

کنترل ScriptManager ، وظیفه مدیریت اسکریپت ها بر روی صفحات ASP.Net ای که قابلیت Ajax آنها فعال است را بر عهده دارد . به طور پیش فرض ، کنترل ScriptManager ، اسکریپت های به وقوع پیوسته متعلق به عملیات های Ajax در صفحه را ، با مجموعه سایر اسکریپت های صفحه مرتبط می کند .
این کار امکان استفاده از قابلیت های مرورگر در سمت کلاینت و آپدیت شدن بخش های مختلف صفحه بدون رفرش شدن کامل آن را می دهد .
شما بایستی از یک کنترل ScriptManager ، برای فعال سازی قابلیت های Ajax زیر در صفحات ASP.Net استفاده نمایید :

- به روز رسانی و تغییر فقط بخشی از صفحه که می خواهید تغییر کند ، به جای Postback شدن کل آن . کنترل های UpdatePannel و UpdateProgress و Timer برای کارکرد صحیح بر روی صفحات ASP.Net، به یک کنترل ScriptManager نیاز دارند.
- امکان دسترسی کلاس های جاوا اسکریپت به سایر اطلاعات صفحات ASP.Net ، مثل اطلاعات اهراز هویت کاربر ، پروفایل ها و. ...
- دسترسی کلاس های جاوا اسکریپت به وب سرویس های فعال بر روی صفحات. ASP.Net

طرز کار کنترل ScriptManager :

هنگامی که صفحه شامل یک یا چندین کنترل UpdatePannel می باشد ، کنترل ScriptManager ، انجام عملیات به روز رسانی و تغییر اطلاعات در آنها را مدیریت می کند . این کنترل با متدهای حیات صفحه (Page life cycles) ، برای به روز رسانی بخش های مختلف کنترل UpdatePannel در ارتباط است .
خاصیت EnablePartialRendering ، تعیین می کند که آیا اطلاعات به روز رسانی و تغییر در صفحه بدون Postback شدن آن فعال باشد یا خیر . به طور پیش فرض ، این مقدار بر روی true تنظیم شده و فعال است .
برای دریافت اطلاعات بیشتر راجع به نحوه استفاده از کنترل های UpdatePannel و UpdateProgress و Timer ، به بخش های آموزشی آنها بروید .

مدیریت خطاهای احتمالی در آپدیت صفحات :

چنانچه در هنگام انجام عمل به روز رسانی و آپدیت بخشی از صفحه ، خطایی رخ دهد ، به وسیله کارهای زیر در کنترل ScriptManager ، می توانید آنها را مدیریت نمایید :

- تنظیم خاصیت AsyncPostBackErrorMessage کنترل ، که تعیین کننده پیام خطایی است که به مرورگر فرستاده می شود.
- مدیریت رویداد AsyncPostBackError کنترل ScriptManager ، که در زمان بروز خطا در عمل آپدیت رخ داده و می توان کدهای مورد نظر خود را در آن قرار داد.
- تنظیم خاصیت AllowCustomErrorsRedirect کنترل ، که تعیین می کند بخش خطاهای احتمالی (Custom Errors) در فایل web.config سایت ، چگونه باید استفاده شوند.

ثبت کردن اسکریپت های مورد نظر در صفحه با کنترل ScriptManager :

می توانید از کنترل ScriptManager ، برای مدیریت منابع کنترل هایی که در عملیات آپدیت و به روز رسانی صفحه ، نقش دارند ، استفاده کنید . این منابع شامل اسکریپت ها ، استایل ها ، فیلدهای مخفی و ... می شوند .
مجموعه اسکریپت های (Scripts Collection) کنترل ScriptManager ، شامل یک شی ScriptReference برای هر اسکریپتی که در صفحه قرار دارد ، می باشد . شما می توانید به طور صریح یا برنامه نویسی شده ، اسکریپت ها را تعیین کنید .
همچنین کنترل ScriptManager ، متدهای ثبت و ارتباط دهی را اجرا می کند ، که به وسیله آنها می توانید ، اسکریپت های کلاینت و فیلدهای مخفی روی صفحه را به صورت برنامه ریزی شده ، مدیریت نمایید .

ثبت کردن وب سرویس ها :

برای ثبت یک وب سرویس که شما می خواهید از یک صفحه ASP.Net با قابلیت Ajax فعال ، آن را فراخوانی نمایید ، باید وب سرویس را با اضافه کردن آن به مجموعه Services کنترل ScriptManager ، ثبت نمایید .
چهارچوب کاری مایکروسافت ایجکس ، یک شی کلاینت را برای هر شی ServiceReference در مجموعه Services کنترل ایجاد می کنند . کلاس پروکسی و اعضای آن از کنترل ScriptManager ، به راحتی با وب سرویس ها از طریق کلاینت اسکریپت ، ارتباط برقرار می کند .

کلاس ScriptManagerProxy :

تنها یک نمونه از کنترل ScriptManager می تواند به صفحه اضافه شود . صفحه می تواند به صورت مستقیم و یا از طریق یک کامپوننت دیگر مثل یک مسترپیج این کنترل را در خود داشته باشد .
اگر یک نسخه از کنترل ScriptManager بر روی صفحه وجود داشته باشد ، ولی یک کامپوننت ترکیبی یا فرزند ، به نمونه دیگری از کنترل نیاز داشته باشد ، آنگاه کامپوننت دوم باید از کلاس ScriptManagerProxy **استفاده** نماید .

آموزش Ajax > کنترل های Ajax ASP.Net > معرفی کنترل Timer

آشنایی با کنترل Timer :

کنترل Timer به صورت متناوب ، در بازه زمانی تعیین شده ، صفحه را Postback می کند . همچنین اگر این کنترل را همراه با یک کنترل UpdatePannel به کار ببرید ، می توانید قفقط بخشی از صفحه که کنترل UpdatePannel شامل آن می شود را به صورت متناوب ، به روز رسانی کنید .
این کنترل همچنین می تواند ، پس از گذشت مدت زمانی معین ، کل صفحه را به آدرس دیگری Post نماید .

کاربردهای کلی کنترل Timer :

شما می توانید از کنترل Timer ، برای انجام امور زیر استفاده نمایید :

- محتویات یک یا چند کنترل UpdatePannel را بدون رفرش کردن و Postback شدن کل صفحه ، در فواصل زمانی معین ، به روز نمایید.
- کد یا کدهای مورد نظر خود را هر بار که کنترل Timer صفحه را Postback می کند ، اجرا نمایید.
- به صورت متناوب و در فواصل زمانی دلخواه ، کل صفحه را به سرور Post نمایید.

نحوه کار کنترل Timer :

کنترل Timer یک کنترل سرور ASP.Net است ، که برای اجرای بر روی صفحات ، یک Component یا جزء جاوا اسکریپت را به صفحه اضافه می کند . جزء جاوا اسکریپت درون صفحه ، پس از گذشت مدت زمان تعیین شده در خاصیت Interval کنترل Timer ، به صورت متناوب ، صفحه را Postback می کند .
در طراحی صفحه ASP.Net ، هر خاصیتی را که برای کنترل Timer تعیین کنید ، در هنگام اجرای صفحه توسط سرور به جزء جاوا اسکریپت مرتبط با آن در صفحه ، ارسال می شود .
برای کارکرد درست کنترل Timer ، حتما بایستی یک کنترل یا یک نمونه از کلاس ScriptManager ، بر روی صفحه وجود داشته باشد .
هر بار که Postback توسط کنترل Timer اجرا شود ، این کنترل رویداد Tick خود را اجرا می کند . شما می توانید ، کدهای مورد نظر خود را برای اجرای متناوب در این رویداد کنترل ، قرار دهید .
از خاصیت Interval کنترل Timer برای تعیین دوره زمانی Postback شدن کنترل و از خاصیت Enabled آن برای روشن یا خاموش کردن آن استفاده می شود .
خاصیت Intervals بر حسب میلی ثانیه تعیین شده و مقدار پیش فرض آن ۶۰۰۰۰ میلی ثانیه ، یا ۶۰ ثانیه است .
همچنین خاصیت Enabled ، می تواند دارای یکی از دو مقدار true یا false باشد .

نکته : تعیین مدت زمان دوره تناوب کنترل Timer به مقدار خیلی کم ، می تواند باعث هنگ کردن صفحه و تحمیل بار اضافه به سرور شود . بنابراین ، فقط در زمانی که نیاز دارید ، صفحه را Postback نمایید .
شما می توانید بیش از یک کنترل Timer را بر روی صفحه قرار دهید ، چنانچه لازم باشد ، کنترل های مختلف UpdatePannel ، در فواصل زمانی گوناگون به روز شوند . اما یک کنترل Timer را می توانید برای کنترل چندین کنترل UpdatePannel ، به صورت همزمان به کار ببرید .

استفاده از یک کنترل Timer درون یک کنترل UpdatePannel :

هنگامی که یک کنترل Timer را درون یک کنترل UpdatePannel قرار دهید ، به صورت اتوماتیک کنترل Timer به عنوان رفرش کننده و آپدیت کننده آن عمل می کند .
شما می توانید این رفتار کنترل Timer را با قرار دادن مقدار خاصیت ChildrenAsTriggers کنترل UpdatePannel ، بر روی مقدار false کنسل کنید .
برای کنترل Timer ای که درون یک کنترل UpdatePannel قرار دارد ، مدت زمان شمارش دوره تناوب زمانی آغاز می شود ، که عمل Postback صفحه به طور کامل اجرا شود . بنابراین دوره تناوب آن تا زمانی که صفحه به طور کامل از سرور باز نگردد ، شروع نمی شود .
برای مثال ، اگر دوره تناوب کنترل تایمر را ۶۰ ثانیه تعیین کرده و انجام عمل Postback صفحه ۳ ثانیه طول بکشد ، سری بعدی دوره تناوب کنترل تایمر ، ۶۳ ثانیه بعد خواهد بود .
در مثال زیر ، کد قرار گرفتن یک کنترل Timer درون یک کنترل UpdatePannel را مشاهده می کنید :

کد

```
<asp:ScriptManager runat="server" ID="ScriptManager1" />

<asp:UpdatePanel runat="server" ID="UpdatePanel1"
UpdateMode="Conditional">
   <ContentTemplate>
       <asp:Timer ID="Timer1" runat="server" Interval="12000"
OnTick="Timer1_Tick">
       </asp:Timer>
    </ContentTemplate>
</asp:UpdatePanel>
```

استفاده از یک کنترل Timer بیرون از یک کنترل UpdatePannel :

اگر کنترل Timer را بیرون از تگ کنترل UpdatePannel باشد ، شما باید صراحتا تعیین کنید که این کنترل Timer به عنوان اجرا کننده و آپدیت کننده کنترل UpdatePannel به صورت متناوب است .
اگر کنترل Timer بیرون کنترل UpdatePannel باشد ، دوره تناوب کنترل تایمر همزمان با شروع عملیات Postback صفحه آغاز می شود .
برای مثال اگر دوره تناوب کنترل تایمر روی ۶۰ ثانیه تنظیم شده باشد و عمل Postback صفحه ۳ ثانیه طول بکشد ، Postback بعدی دقیقا ۶۰ ثانیه دیگر انجام می شود . بنابراین کاربر محتوی جدید صفحه را فقط برای ۵۷ ثانیه مشاهده خواهد کرد .
در مثال زیر ، کد قرار گرفتن یک کنترل Timer بیرون از یک کنترل UpdatePannel را مشاهده می کنید :

کد

```
<asp:ScriptManager runat="server" ID="ScriptManager1" />

<asp:Timer ID="Timer1" runat="server" Interval="12000" OnTick="Timer1_Tick">
</asp:Timer>
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
   <Triggers>
       <asp:AsyncPostBackTrigger ControlID="Timer1" EventName="Tick" />
```

```
    </Triggers>
    <ContentTemplate>
        <asp:Label ID="Label1" runat="server"></asp:Label>
    </ContentTemplate>
</asp:UpdatePanel>
```

آموزش Ajax > کنترل های Ajax ASP.Net > معرفی کنترل UpdatePannel

آشنایی با کنترل UpdatePannel :

کنترل UpdatePannel این امکان را به شما می دهد ، تا برنامه های وب قوی و با رابط کاربری همانند نرم افزارهای دسکتاپ بسازید . به وسیله این کنترل می توانید بخش یا بخش هایی از صفحه را که می خواهید اطلاعات آن تغییر کرده را آپدیت کرده و مانع رفرش شدن و Postback کل صفحه به سرور شوید .
به این کار در اصطلاح آپدیت جزیی صفحه می گویند . در حالت ساده ، برای مثال زمانی که می خواهید اطلاعات نمایش داده شده توسط یک کنترل GridView را آپدیت کنید ، این کار مستلزم رفرش شدن و Postback کل صفحه به سرور است .
در حالی که با قرار دادن همان کنترل GridView درون یک کنترل UpdatePannel ، در زمان آپدیت فقط کنترل GridView آپدیت شده و بقیه صفحه بدون تغییر یا Postback باقی می مانند . این روش ، حس کار با یک نرم افزار دسکتاپ را به کاربر داده و باعث رضایت بیشتر وی و سرعت بالاتر پاسخگویی صفحه می شود .
تمامی این امور را می توانید با قرار دادن یک کنترل UpdatePannel به همراه یک کنترل ScriptManager بر روی صفحه و بدون نیاز به هیچ کدنویسی خاصی انجام دهید .

نحوه کار کنترل UpdatePannel :

کنترل UpdatePannel ، با جدا کردن بخش مورد نظر از صفحه ، در هنگام به روز رسانی خود ، فقط آن بخش یا بخش های مرتبط را آپدیت کرده و مانع رفرش شدن و Postback کل صفحه می شود .این کار به وسیله یک کنترل ScriptManager و کلاس PageRequestManager در صفحه انجام می شود .
در این حالت آپدیت صفحه فقط به بخش یا بخش هایی محدود می شود ، که درون کنترل UpdatePannel بوده و برای ارسال کد مشخص شده اند . مرورگر فقط محتویات آن بخش را برای سرور ارسال می کند ، سپس سرور درخواست را بررسی کرده و جواب آن را حاضر می کند . در آخر جواب حاضر شده به صفحه برگردانده شده و به وسیله متدهای DOM در بخش های HTML مورد نظر اعمال می شود .
مثال زیر ، انجام روند اجرای یک صفحه از ابتدا و سپس آپدیت شدن بخش قرار گرفته در کنترل UpdatePannel را به صورت یک دیاگرام نشان می دهد :

فعال کردن به روز رسانی بخشی (partial-page update) در صفحه :

کنترل UpdatePannel ، برای کارکرد صحیح در صفحه ، به یک کنترل ScriptManager نیاز دارد . به صورت پیش فرض ، به روز رسانی بخشی صفحه فعال است ، زیرا مقدار خاصیت EnablePartialRendering کنترل ScriptManager بر روی true تنظیم شده است .
کد زیر ، نحوه تعریف و استفاده از یک کنترل ScriptManager و UpdatePannel را به صورت همزمان نشان می دهد . کنترل UpdatePannel شامل یک دکمه فرمان است که در هنگام کلیک بر روی آن ، محتویات درون UpdatePannel به روز می شود .
به صورت پیش فرض مقدار خاصیت ChildrenAsTriggers کنترل UpdatePannel بر روی مقدار true تنظیم شده و همین باعث می شود تا کلیک بر روی دکمه فرمان که یک کنترل فرزند UpdatePannel است ، باعث آپدیت محتویات آن شود .

کد	<asp:Button ID="Button1" Text="Refresh Panel" runat="server" /> <asp:ScriptManager ID="ScriptManager1" runat="server" />

	<asp:UpdatePanel ID="UpdatePanel1" UpdateMode="Conditional" runat="server"> <Triggers> <asp:AsyncPostBackTrigger ControlID="Button1" /> </Triggers> <ContentTemplate> <fieldset> <legend>UpdatePanel content</legend> <%=DateTime.Now.ToString() %> </fieldset> </ContentTemplate> </asp:UpdatePanel>

تعیین محتویات برای کنترل UpdatePannel :

می توانید محتویات درونی را برای کنترل UpdatePannel به صورت مستقیم در بخش کد وارد کرده و یا از ابزار ContentTemplate در ویرایشگر ویژوال استودیو استفاده کنید .
محتویات کنترل UpdatePannel ، درون تگ باز و بسته <ContentTemplate> آن ، قرار می گیرد . برای اضافه کردن محتویات در حین اجرا و به صورت برنامه ریزی شده ، بایستی از خاصیت ContentTemplateContainer کنترل استفاده نمایید .

تعیین کنترل یا کد برای فعال کردن به روز رسانی در کنترل UpdatePannel :

به صورت پیش فرض ، هر کنترل درون کنترل UpdatePannel که قابلیت Postback داشته باشد ، در هنگام تغییر ، باعث رفرش شدن و آپدیت محتویات درونی کنترل UpdatePannel می شود .
همچنین می توانید کنترل دیگری که خارج از کنترل UpdatePannel وجود دارد را نیز ، برای رفرش کردن آن تنظیم نمایید . این کار با تعیین یک trigger برای کنترل UpdatePannel انجام می شود . trigger ، کنترل یا رویدادی است که باعث می شود تا محتویات درون کنترل UpdatePannel **رفرش شود** . برای مثال یک دکمه **فرمان** .
کد زیر نشان می دهد که چگونه یک دکمه فرمان را برای رفرش کردن محتویات یک کنترل UpdatePannel به کار ببرید :

کد	<asp:Button **ID="Button1"** Text="Refresh Panel" runat="server" /> <asp:ScriptManager ID="ScriptManager1" runat="server" /> <asp:UpdatePanel ID="UpdatePanel1" UpdateMode="Conditional" runat="server"> <Triggers> **<asp:AsyncPostBackTrigger ControlID="Button1" />** </Triggers> <ContentTemplate> <fieldset>

```
            <legend>UpdatePanel content</legend>
                <%=DateTime.Now.ToString() %>
            </fieldset>
        </ContentTemplate>
    </asp:UpdatePanel>
```

trigger به وسیله تگ <asp:AsyncPostBackTrigger> درون المنت <Triggers> کنترل UpdatePannel تعیین می شود . (در محیط ویژوال استودیو می توانید برای کنترل مورد نظرتان ، trigger هایی را به وسیله پنجره UpdatePanelTrigger Collection Editor ایجاد نمایید) .

کنترل های UpdatePannel چگونه رفرش می شوند:

لیست زیر تعیین می کند که چگونه و چه زمانی کنترل UpdatePannel رفرش‌شود :

- اگر مقدار خاصیت UpdateMode را روی Always تنظیم کنید ، کنترل UpdatePannel با هر Postbackای ، از هر نقطه یا کنترلی در صفحه رفرش می شود . این مسله شامل تمام کنترل هایی که چه درون UpdatePannel هستند یا بیرون آن ، صدق می کند.
- اگر مقدار خاصیت UpdateMode را روی Conditional کفه مقدار پیش فرض است ، فقط در حالات زیر محتویات کنترل به روز می شود:
  - هنگامی که یک Postback یا trigger برای کنترل UpdatePannel فعل‌شود.
  - هنگامی که به صورت مستقیم رویداد Update کنترل UpdatePannel را فراخوانی نمایید.
  - هنگامی که کنترل UpdatePannel در یک کنترل دیگر UpdatePannel قرار دارد و کنترل مادر parent آن ، به روز می شود.
  - هنگامی که مقدار خاصیت ChildrenAsTriggers کنترل روی true تنظیم شده باشد و یک کنترل فرزند درونی آن عمل Postback را انجام دهد.

آموزش Ajax > کنترل های Ajax ASP.Net > معرفی کنترل UpdateProgress

آشنایی با کنترل UpdateProgress :

از کنترل UpdateProgress ، برای نمایش یک نمایه تصویری یا یک پیام متنی ، که میزان پیشرفت پروسه عملیات لود شدن و به روز رسانی صفحه را نشان می دهد ، استفاده می شود .
شما می توانید ظاهر و محتویان کنترل UpdateProgress را به راحتی تعیین نمایید . نمونه این کنترل را تاکنون زیاد مشاهده کرده اید . برای مثال هنگامی که صفحه در حال لود کردن اطلاعات از یک منبع داده ای است ، یک ساعت دوار به کاربر نشان می دهد ، که عملیات در حال انجام است و باید منتظر بماند .
برای جلوگیری از نمایش فوق سریع کنترل در زمانی که عملیات به روز رسانی صفحه بسیار سریع است ، می توان یک مدت زمان تاخیری را به عنوان Delay برای نمایش کنترل در نظر گرفت .
شما می توانید یک یا چند کنترل UpdateProgress را بر روی صفحه قرار دهید ، که هر کدام مربوط به یک کنترل UpdateProgress باشند . یا اینکه یک کنترل UpdateProgress واحد را برای چندین کنترل UpdatePannel به کار ببرید .

نحوه کار کنترل UpdateProgress :

کنترل UpdateProgress ، در هنگام اجرا بر روی صفحات وب ، به صورت یک تگ قرار می گیرد ، که بسته به اینکه کنترل UpdatePannel مربوط به آن ، در حال به روز رسانی است یا خیر ، مخفی یا نمایان می شود .
در هنگام لود اولیه صفحه یا رفرش شدن کل صفحه ، این کنترل نمایش داده نخواهد شد .

اتصال یک کنترل UpdateProgress به یک کنترل UpdatePannel :

شما می توانید به وسیله خاصیت AssociatedUpdatePanelID کنترل UpdateProgress را به یک کنترل UpdatePannel متصل کنید . هنگامی که عمل Postback یا رفرش شدن توسط کنترل UpdatePannel انجام می شود ، کنترل UpdateProgress مرتبط با آن نمایش داده می شود .
اگر کنترل UpdateProgress را به هیچ کنترل UpdatePannel خاصی مرتبط نکنید ، با هر بار Postback یا به روز رسانی ، آن کنترل در صفحه نمایش داده خواهد شد .

تعیین محتویات مورد نظر برای یک کنترل UpdateProgress :

می توانید تعیین کنید تا متن یا پیام خاصی توسط کنترل UpdateProgress در هنگام نمایش ، نشان داده شود .
برای این منظور بایستی متن یا کد HTML مورد نظر خود را درون تگ باز و بسته ProgressTemplate کنترل قرار دهید . همانند مثال زیر :

کد

```
<asp:UpdateProgress ID="UpdateProgress1" runat="server">
    <ProgressTemplate>
        An update is in progress...
    </ProgressTemplate>
</asp:UpdateProgress>
```

تعیین چگونگی نمایش محتویات کنترل :

هنگامی که مقدار خاصیت DynamicLayout کنترل UpdateProgress را بر روی true تنظیم نمایید ، کنترل در هنگام اجرا بر روی صفحه ، هیچ فضایی را اشغال نمی کند . در عوض ، صفحه به صورت دینامیکی آن را در مواقع لازم نمایش خواهد دارد . برای پشتیبانی از نمایش دینامیک ، کنترل UpdateProgress بر روی صفحه در هنگام اجرا به صورت یک تگ رندر می شود ، که مقدار خاصیت dispaly آن برابر با none است .
اما هنگامی که مقدار خاصیت DynamicLayout روی false باشد ، کنترل UpdateProgress فضایی را بر روی صفحه اشغال می کند ، حتی اگر مخفی باشد . در این حالت ، کنترل به صورت یک تگ رندر شده ، که مقدار خاصیت dispaly آن روی block و مقدار خاصیت visibility اش روی hidden تنظیم شده است .

قرار دادن کنترل UpdateProgress بر روی صفحه :

می توانید کنترل UpdateProgress را درون یا بیرون از کنترل UpdatePannel قرار دهید . یک کنترل UpdateProgress هر زمان که کنترل UpdatePannel مرتبط به آن ، در حال به روز رسانی باشد ، نمایش داده می شود . حتی اگر درون کنترل UpdatePannel دیگری قرار گرفته باشد .

تعیین اینکه چه زمانی کنترل UpdateProgress نمایش داده شود :

شما می توانید به صورت برنامه ریزی شده ، تعیین نمایید که چه زمانی کنترل UpdateProgress نمایش داده شود . برای این منظور بایستی از رویدادهای beginRequest و endRequest در جاوا اسکریپت استفاده نمایید (متعلق به کلاس PageRequestManager) .
در رویداد beginRequest عنصر DOM ای که جانشین کنترل UpdateProgress است را نمایش داده و در رویداد endRequest آن را مخفی نمایید .

مثال عملی کار با کنترل تایمر Timer ، UpdatePannel و ScriptManager :

در این بخش قصد داریم تا با ارائه یک مثال عملی و تشریح کد آن ، نحوه استفاده از کنترل تایمر Timer در ASP.Net و ScriptManager،UpdatePannel را به شما آموزش دهیم .

کد مثال زیر را مرور نمایید . هر بخش از کد که نیاز به توضیح داشته ، را با یک رنگ مجزا مشخص کرده وسپس آن را در پایان مثال توضیح داده ایم ::

کد

```
<%@ Page Language="C#" AutoEventWireup="true" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
  <html xmlns="http://www.w3.org/1999/xhtml">
   <head id="Head1" runat="server">
        <title> Timer مثال عملی کار با کنترل تایمر</title>
        <script runat="server">
            protected void Page_Load(object sender, EventArgs e)
                {
                    OriginalTime.Text = DateTime.Now.ToLongTimeString();    // این تابع
زمان ایجاد شدن اولیه صفحه را در کنترل مربوطه قرار می دهد
                    StockPrice.Text = "50";        // تعیین مقادیر اولیه قیمت و ساعت
                    TimeOfPrice.Text = "00:00:00";
                }
            protected void Timer1_Tick(object sender, EventArgs e)
                {
                    StockPrice.Text = GetStockPrice( );      // این خط کد یک قیمت جدید را از تابع
مربوط به آن درخواست می کند
                    TimeOfPrice.Text = DateTime.Now.ToLongTimeString( );    // این خط
کد نیز ،قیمت جدید ارسالی را در کنترل مربوط به آن آپدیت می کند
                }
             private string GetStockPrice()
                {
                    double randomStockPrice = 50 + new Random().NextDouble( );   //
این خط کد یک قیمت شانسی تولید می کند
                    return randomStockPrice.ToString("C");     // این خط کد ، قیمت جدید ایجاد
شده را به تابع درخواست کننده ، برمی گرداند
                }
        </script>
```

```
  </head>
  <body style ="text-align : right ; direction : rtl">
    <form id="form1" runat="server">
        <asp:ScriptManager ID="ScriptManager1" runat="server" />
<%-- وجود یک نمونه از این کنترل برای اجرای کد ایجکس ضروری است --%>
        <asp:Timer ID="Timer1" OnTick="Timer1_Tick" runat="server" Interval="10000"
/>          <%-- کد کنترل تایمر --%>
        <asp:UpdatePanel ID="StockPricePanel" runat="server"
UpdateMode="Conditional">
            <Triggers>
                <asp:AsyncPostBackTrigger ControlID="Timer1" />
            </Triggers>
          <ContentTemplate>
            قیمت برابر است با :  <asp:Label id="StockPrice" runat="server"></asp:Label><BR
/>
            در زمان:<asp:Label id="TimeOfPrice" runat="server"></asp:Label>
          </ContentTemplate>
        </asp:UpdatePanel>
        <div>
            صفحه در این زمان ساخته شده است : <asp:Label ID="OriginalTime"
runat="server"></asp:Label>
        </div>
      </form>
  </body>
</html>
```

خروجی

مشاده خروجی مثال

- کنترل trigger یا رفرش کننده کنترل UpdatePannel را تعیین می کند ، که در این مثال Timer1 بوده و هر ۱۰ ثانیه آن را رفرش می کند.
- محتویات درونی کنترل UpdatePannel را تعیین می کند ، که هر ۱۰ ثانیه یک بار رفرش شده و قیمت جدید را نمایش می دهند.
- خط اول این بخش کد یک کنترل ScriptManager را تعیین می کند . برای عملکرد صحیح کنترل های Timerو UpdatePannel وجود یک کنترل ScriptManager بر روی صفحه اجباری است .
خط دوم نیز کد کنترل Timer است که هر ۱۰ ثانیه یکبار triggerشده و تابع Timer1_Tick خود را فراخوانی و اجرا می کند.

مثال عملی کار با کنترل UpdateProgress :

در این بخش قصد داریم تا با ارائه یک مثال عملی و تشریح کد آن ، نحوه استفاده از کنترل UpdateProgress در ASP.Net را به شما آموزش دهیم .

کد مثال زیر را مرور نمایید . هر بخش از کد که نیاز به توضیح داشته ، را با یک رنگ مجزا مشخص کرده و سپس آن را در پایان مثال توضیح داده ایم :

در مثال زیر یک کنترل UpdateProgress ، وضعیت عملیات آپدیت شدن ۲ کنترل UpdatePannel را نمایش می دهد .

کد

```
<%@ Page Language="C#" AutoEventWireup="true" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
    <script runat="server">
        protected void Button_Click(object sender, EventArgs e)        // کد تابعی که در هنگام کلیک بر روی دکمه فرمان ها اجرا می شود
            {
                System.Threading.Thread.Sleep(3000);      //    سرور یک تاخیر ۳ ثانیه ای را انجام می دهد
            }
    </script>
  <html xmlns="http://www.w3.org/1999/xhtml">
   <head id="Head1" runat="server">
        <title> UpdateProgress  مثال عملی کار با کنترل تایمر</title>
        <style type="text/css">
            #UpdatePanel1, #UpdatePanel2, #UpdateProgress1 {      //  تنظیم خواص مورد نظر برای کنترل های مختلف
                border-right: gray 1px solid; border-top: gray 1px solid;
                border-left: gray 1px solid; border-bottom: gray 1px solid;
            }
            #UpdatePanel1, #UpdatePanel2 {
                width:200px; height:200px; position: relative;
                float: left; margin-left: 10px; margin-top: 10px;
            }
            #UpdateProgress1 {
                width: 400px; background-color: #FFC080;
                bottom: 0%; left: 0px; position: absolute;
```

```
            }
        </style>
    </head>
    <body>
        <form id="form1" runat="server">
            <div>
                <asp:ScriptManager ID="ScriptManager1" runat="server" />    <%-- برای کارکرد صحیح سایر کنترل ها وجود یک نمونه از این کنترل بر روی صفحه اجباری است --%>
                <asp:UpdatePanel ID="UpdatePanel1" UpdateMode="Conditional" runat="server">
                    <ContentTemplate>
                        <%=DateTime.Now.ToString() %> <br />
                        <asp:Button ID="Button1" runat="server" Text="Refresh Panel" OnClick="Button_Click" />
                    </ContentTemplate>
                </asp:UpdatePanel>
                <asp:UpdatePanel ID="UpdatePanel2" UpdateMode="Conditional" runat="server">
                    <ContentTemplate>
                        <%=DateTime.Now.ToString() %> <br />
                        <asp:Button ID="Button2" runat="server" Text="Refresh Panel" OnClick="Button_Click"/>
                    </ContentTemplate>
                </asp:UpdatePanel>
                <asp:UpdateProgress ID="UpdateProgress1" runat="server">
                    <ProgressTemplate>
                        Update in progress...
                    </ProgressTemplate>
                </asp:UpdateProgress>
            </div>
        </form>
    </body>    </body>
</html>
```

خروجی

مشاهده خروجی مثال

- کد تابع جاوا اسکریپتی که در هنگام کلیک بر روی دکمه فرمان های موجود در دو کنترل UpdatePannel، اجرا می شود.
- کد کنترل UpdatePannel اول.
- کد کنترل UpdatePannel دوم.
- کد کنترل UpdateProgress ، که یک نمایه تصویری را در زمان آپدیت شدن کنترل های UpdatePannel، بر روی صفحه نمایش می دهد.