



www.mohandesyar.com

عنوان



واژه Ajax سرنام عبارت Asynchronous JavaScript and XML و به معنی < جاوا اسکریپت و XML آسنکرون یا نامتقارن > است. ای جکس فناوری جدیدی است که به کمک آن می توان اینترنتیسی یک برنامه تحت وب را طوری ساخت که وقتی کاربر روی دکمه یا لینکی کلیک می کند، کلیه عملیات ارسال اطلاعات و دریافت نتایج در پشت صحنه انجام شود و فقط آن قسمت از اینترنتیسی که قرار است اطلاعات جدید را به نمایش درآورد تغییر کند بدون اینکه تمام صفحه از نو بارگذاری شود. در این مقاله می خواهیم این فناوری نوین و تحول آفرین را از جنبه های مختلف مورد بررسی قرار دهیم و با مبانی فنی آن آشنا شویم.



جسی جیمز گرت پیشنهاد دهنده ای جکس

کمریندها را ببندید

آیا از دنیای کلاسیک وب خسته شده اید؟ آیا مایلید با چهره جدید وب آشنا شوید؟ آیا می خواهید با جدیدترین فناوری دنیای برنامه نویسی وب آشنا شوید؟ برای پرواز آماده باشید! مقصد ما یکی از استان های کشور جدید التاسیس Web 2.0 است. می خواهیم با هم به دنیای هیجان انگیز Ajax سفر کنیم.

در سرزمین ای جکس چهره سایت های وب مانند نرم افزارهای رومیزی به نظر می رسند. اکنون وب خانه دوم اطلاعات دیجیتالی شما است. حالا که با داشتن یک خط اینترنت پرسرعت می توانید در تمام طول شبانه روز به اینترنت متصل باشید، وقت آن فرارسیده که نرم افزارهای تحت وب نیز متحول شوند و کارکرد و قابلیت های مانند نرم افزارهای رومیزی داشته باشند. به این ترتیب فاصله و تفاوت میان نرم افزارهای دسکتاپ و نرم افزارهای آنلاین ناپدید می شود. دیگر لازم نیست وقتی روی دکمه submit کلیک می کنید منتظر بارگذاری مجدد صفحه وب بمانید. همه چیز فوراً و به سرعت اتفاق می افتد.

تصور کنید به صورت آنلاین در حال خواندن نامه های الکترونیکی خود در یکی از سایت های یاهو یا جی میل هستید. فوراً یادتان می افتد که باید به کسی نامه ای بفرستید. لازم نیست صفحه ای که در حال مطالعه آن هستید را ترک کنید و یا با استفاده از ترفندهایی مانند رایت کلیک و فرمان Open in New window یک پنجره جدید باز کنید و منتظر بارگذاری مجدد صفحه شوید. شما فقط می خواهید یک پنجره جدید باز کنید که داخلش نامه خود را تایپ کنید.

چه نیازی به بارگذاری مجدد آن همه اطلاعات جزئی و جنبی (مانند لوگوی سایت، نام آیدی شما و ...) است که در اطراف اطلاعات اصلی به نمایش درآمده اند؟ ای جکس همین کار را برای شما می کند. وقتی که نامه خود را ارسال کردید نیز فقط همان پنجره بسته می شود و چند بایت برای سایت مقصد ارسال می شود و اینترنتیسی صندوق پستی نامه های شما بدون این که تکان بخورد و یا دوباره به نمایش درآید همچنان پیش چشم شما و منتظر فرمان بعدی است. در دنیای ای جکس دکمه های Back و Forward مرورگر معنی و کاربرد خود را تقریباً از دست می دهند.

همه دکمه های مورد نیاز داخل اینترنتیسی سایت موجود است. برخلاف جهان Web 1.0، در دنیای ای جکس گشت و گذار در وب لزوماً خطی نیست. اگر می خواهید از یک فروشگاه الکترونیکی خرید کنید، می توانید محصول مورد نظران را به روش drag&drop داخل سبد ببندازید و با فشردن دکمه نهایی، خرید خود را ثبت

کنید. پیغام ثبت موفقیت آمیز سفارش برای شما نمایش داده می‌شود و کار تمام است. لازم نیست از صفحه الف به صفحه ب و سپس از آنجا به صفحه ج بروید.

ای‌جکس چیست؟

این واژه را اولین بار جسی جیمز گرت (Jesse James Garrett)، برنامه‌نویسی از شرکت Adaptive Path در مقاله‌ای با عنوان <ای‌جکس: رهیافت جدیدی در برنامه‌های تحت وب > به کار برد. ای‌جکس سرنام عبارت Asynchronous JavaScript and XML و به معنی ترکیب نامتقارن جاوا اسکریپت و XML است.

چرا نامتقارن؟ چون ترکیب این دو فناوری به شما کمک می‌کند قسمتی از یک صفحه وب را بروز کنید بدون اینکه لازم باشد <همزمان یا متقارن > با این عمل، کل آن صفحه از نو بارگذاری شود. این عملیات توسط شیء و فرمانی به نام XMLHttpRequest انجام می‌شود که سال‌ها پیش، هنگام معرفی نسخه 4 مرورگر اینترنت اکسپلورر وارد این نرم‌افزار شده و به دلیل قابلیت‌های جالبی که دارد، این روزها تمام مرورگرهای معروف و قدرتمند وب مانند فایرفاکس و اپرا از آن پشتیبانی می‌کنند. به کمک این شیء می‌توانید برنامه‌هایی به زبان جاوا اسکریپت بنویسید که در پشت صحنه یک صفحه وب اطلاعاتی را به سرور بفرستند و داده‌هایی را دریافت کنند.

ای‌جکس اساساً رهاورده‌ای از دنیای جاوا است و بار دیگر فناوری جاوا و زبان اسکریپت‌نویسی وابسته به آن را در کانون توجه برنامه‌نویسان قرار داده است. اما خبر خوب اینست که چون همه پلتفرم‌های برنامه‌نویسی وب از قبیل PHP، ASP.NET و JSP از جاوا اسکریپت و XML پشتیبانی می‌کنند، فناوری ای‌جکس در همه این پلتفرم‌ها پیاده‌سازی شده و حتی جالب است بدانید که تعداد مازول‌های ای‌جکس نوشته شده برای دات نت و PHP بیشتر از انواع جاوایی آن است.

مایکروسافت که سال‌ها پیش از شیء XMLHttpRequest در نرم‌افزار Outlook Web Access استفاده کرد و جزو اولین ترویج‌دهندگان این تکنیک به شمار می‌رود اخیراً نرم‌افزاری به نام Atlas معرفی کرده که مخصوص برنامه‌نویسی مبتنی بر ای‌جکس است. اما چون ای‌جکس بر اساس جاوا کار می‌کند بدیهی است که جنبش این سورس و خیل عظیم برنامه‌نویسان جاوا نیز بیکار نمانده و می‌کوشند تا دیر نشده Ajax را به قلب دنیای نرم‌افزارهای آزاد بیاورند و از انحصاری شدن آن جلوگیری کنند تا این فناوری به استاندارد جدید وب تبدیل شود. به نظر می‌رسد که آنها موفق بوده‌اند زیرا حتی پیاده‌سازی‌های دات نت ای‌جکس نیز اغلب به صورت این سورس است.

مکانیزم Ajax

جسی جیمز گرت در مقاله معروف خود شیوه عمل ای‌جکس را چنین توضیح می‌دهد: هر عمل از سوی کاربر که به طور معمول موجب تولید یک تقاضای HTTP شود به جای ارسال مستقیم به وب، موجب فراخوانی یک فرمان جاوا اسکریپتی و هدایت آن به موتور ای‌جکس می‌شود. هر نوع پاسخی به کاربر از سوی سرور (مانند کنترل صحت داده‌های وارد شده در یک فرم ورود اطلاعات، ویرایش اطلاعات در حافظه و حتی برخی از انواع هدایت کاربر در سایت) نیازی به ارسال یک صفحه جدید به سمت کاربر ندارد و تنها همان قسمتی که باید تغییر کند بروز می‌شود.

به طور سنتی وقتی کاربر فرمی را پر می‌کند و به سایت ارسال می‌کند، وب سرور با بارگذاری مجدد یا تازه‌سازی صفحه (refresh) و نمایش یک پیغام و یا نتیجه پردازش اطلاعات، به او پاسخ می‌دهد و به همین دلیل هم وقت سرور برای ارسال کل محتوای آن صفحه گرفته می‌شود و هم کاربر باید برای دریافت کامل آن صفحه منتظر بماند؛ که نتیجه آن کاهش بازده سرور، مصرف پهنای باند و تلف شدن وقت و هزینه است. اما به کارگیری تکنیک ای‌جکس این مشکلات را به طرز قابل‌ملاحظه‌ای کاهش می‌دهد.

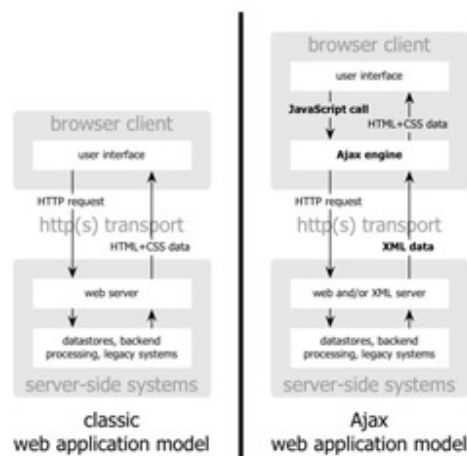
گوگل باز هم پیشتر



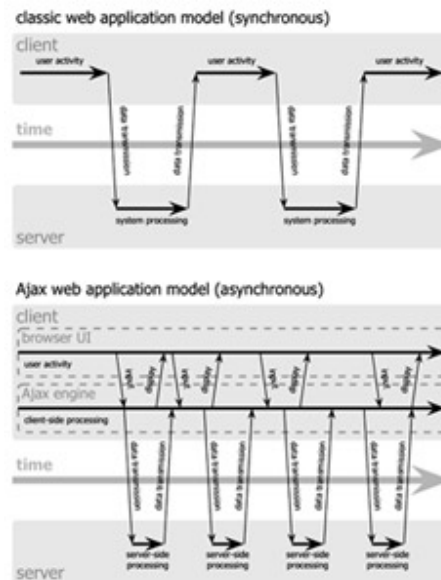
همانطور که جسی جیمز گرت می‌نویسد، گوگل در میان سایت‌های اینترنتی از نظر میزان به‌کارگیری فناوری ای‌جکس پیشتر است و این تکنیک را در نرم‌افزار سایت‌های **Orkut**، **Google Groups**، **Google Suggest**، سیستم **Google Maps** به کار برده است. همچنین به گفته گرت بسیاری از قابلیت‌های دوست‌داشتنی **Flickr**، سایت به اشتراک‌گذاری عکس **یاهو**، و نیز موتور جستجوی **A9** در سایت آمازون بر اساس ای‌جکس کار می‌کنند.

گرت می‌نویسد: اگر موتور ای‌جکس برای پاسخ دادن به کاربر نیازمند گرفتن اطلاعاتی از سمت سرور است، اگر قرار است داده‌ها برای پردازش به سرور ارسال شوند، اگر لازم است کدهای اضافی برای نمایش تغییرات اینترنتی بارگذاری شوند، اگر نیاز به بازیابی و بیرون کشیدن اطلاعات از بانک اطلاعاتی باشد، همه این کارها به‌طور آسنکرون و با استفاده از **XML**، بدون اینکه وقفه‌ای در تماس میان کاربر و اینترنتیسیس نرم‌افزار به‌وجود آید توسط موتور ای‌جکس انجام خواهد شد.

شکل 1



شکل 2



شکل‌های 1 و 2 تصاویری هستند که گرت در مقاله خود استفاده کرده است. شکل 1 مقایسه‌ای است که نشان می‌دهد که برنامه‌های تحت وب سنتی چگونه کنش‌های کاربر را مدیریت می‌کنند و ای‌جکس چگونه همین عملیات را کنترل می‌کند.

شکل 2 مقایسه دیگری است که نشان می‌دهد اگر تعامل کاربر با یک سایت را در محور زمان ترسیم کنیم، چه تفاوتی میان فرآیند ارسال و دریافت داده‌ها در نرم‌افزارهای سنتی و نرم‌افزارهای مبتنی بر فناوری ای‌جکس وجود دارد.

اگر به نمودار شکل 2 دقت کنید متوجه می‌شوید که هنگامی که اینترفیس یک برنامه وب از ای‌جکس استفاده می‌کند، تماس میان کاربر و اینترفیس هرگز قطع نمی‌شود. او همواره نرم‌افزار را در دسترس و پیش روی خود می‌بیند و موتور ای‌جکس در پشت صحنه عملیات ارسال و دریافت داده‌ها را مدیریت می‌کند.

مزایا و معایب Ajax

طی سال‌های اخیر صنعت نرم‌افزار در عرصه وب به سویی تولید سیستم‌هایی حرکت کرده‌است که هرچه بیشتر مستقل از نوع سیستم‌عامل و مرورگر باشد که کاربر استفاده می‌کند.

نگاهی به سیر تحولات مربوط به زبان‌های برنامه‌نویسی وب مانند PHP و ASP از یک سو و کاهش اقبال برنامه‌نویسان به فناوری‌هایی همچون ActiveX و Java Applet در سمت کلاینت از سوی دیگر، موید این نکته است.

در واقع یکی از دلایل مهم توجه دنیای نرم‌افزار به فناوری ای‌جکس همین است. این دستاورد برخلاف فناوری‌هایی مانند Macromedia Flash نیازمند نصب هیچ نرم‌افزار الحاقی و اضافی روی مرورگر نیست و همین حالا بستر لازم برای پیاده‌سازی ای‌جکس روی میلیون‌ها سایت و مرورگر وب وجود دارد. در این میان، اتکای ای‌جکس بر فناوری XML از اهمیت ویژه‌ای برخوردار است.

زیرا XML ماهیتاً یک فناوری باز است که انعطاف‌پذیری زیادی دارد و هم‌اکنون در مقیاس گسترده‌ای در نرم‌افزارهای تحت وب، از سیستم‌های تجارت الکترونیک گرفته تا نرم‌افزارهای مدیریت محتوای سایت‌های وب و فناوری‌هایی همچون وب‌سرویس، RSS و حتی پادکستینگ مورد استفاده قرار می‌گیرد.

ظهور ای‌جکس نشان می‌دهد که فناوری XML هنوز قابلیت‌های کشف‌نشده‌ای دارد که همچنان بکر مانده‌اند و می‌توانند منشا تحولات بزرگ در سیستم‌های اطلاعاتی و ارتباطی باشند.

طعم کد Ajax چگونه است؟

ورود به مباحث کدنویسی ای جکس مقوله پیچیده و مفصلی است که در این مقاله نمی‌گنجد، ولی خالی از لطف نیست که کمی با شکل و قیافه کدهای ای جکس آشنا شوید. کد زیر نمونه‌ای از اعلان شیء XMLHttpRequest را نشان می‌دهد:

```
<script language="javascript" type="text/javascript">
var xmlhttp = new XMLHttpRequest();
</script>
```

کد 1

کد 2 نمونه‌ای از اعلان شیء XMLHttpRequest را در مرورگرهای مایکروسافتی نشان می‌دهد:

```
var xmlhttp = false;
try {
    xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
    try {
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (e2) {
        xmlhttp = false;
    }
}
```

کد 2

کد 3 نمونه‌ای از کدنویسی جاوا اسکریپت را نشان می‌دهد. در ای جکس به وفور از متدولوژی DOM در جاوا اسکریپت و DHTML استفاده می‌کنیم:

```
// Get the value of the "phone" field and stuff it in a variable called phone
var phone = document.getElementById("phone").value;

// Set some values on a form using an array called response
document.getElementById("order").value = response[0];
document.getElementById("address").value = response[1];
```

کد 3

کد 4 نیز نمونه‌ای از روش ارسال یک تقاضا به وسیله Ajax را نشان می‌دهد:

```
function callServer() {
    // Get the city and state from the web form
    var city = document.getElementById("city").value;
    var state = document.getElementById("state").value;
    // Only go on if there are values for both fields
    if ((city == null) || (city == "")) return;
    if ((state == null) || (state == "")) return;

    // Build the URL to connect to
    var url = "/scripts/getZipCode.php?city=" + escape(city) + "&state=" + escape(state);

    // Open a connection to the server
    xmlhttp.open("GET", url, true);

    // Setup a function for the server to run when it's done
    xmlhttp.onreadystatechange = updatePage;

    // Send the request
    xmlhttp.send(null);
}
```


این تقاضا می‌تواند مثلاً از قسمتی از یک صفحه وب به صورت کد 5 باشد :

```
<form>
<p>City: <input type="text" name="city" id="city" size="25"
onchange="callServer();" /></p>
<p>State: <input type="text" name="state" id="state" size="25"
onchange="callServer();" /></p>
<p>Zip Code: <input type="text" name="zipCode" id="city" size="5" /></p>
</form>
```

کد 5

نهایتاً پاسخ دریافت شده از سمت سرور را نیز می‌توان به وسیله کدی شبیه به کد 6 مورد استفاده قرار داد :

```
function updatePage() {
if (xmlHttp.readyState == 4) {
var response = xmlHttp.responseText;
document.getElementById("zipCode").value = response;
}
}
```

کد 6

در همین ارتباط، رویکرد ای‌جکس به سمت فناوری وب‌سرویس قابل توجه است. از آنجا که Ajax تا حد زیادی به معماری XMLHttpRequest وابسته است، این قابلیت را دارد که فرامین ارسال و دریافت اطلاعات را نه فقط از طریق سایت اصلی نمایش دهنده یک صفحه وب، بلکه از طریق سایت‌های دیگری که احتمالاً اطلاعات خود را بر بستر وب سرویس و XML قرار داده‌اند نیز دریافت کند. بنابراین ای‌جکس بالقوه یک فناوری سازگار با وب سرویس نیز هستند. مزیت مهم دیگر Ajax بهره‌گیری آن از استاندارد CSS است.

استانداردی که به تنهایی منشأ تحولات دیگری در زمینه اینترنتیس نرم‌افزارهای وب است و در آینده بسیار نزدیک به وجه غالب صفحه‌آرایی و اینترنتیس سایت‌های وب تبدیل خواهد شد. توجه ای‌جکس به CSS نشان می‌دهد که این فناوری همزمان با به‌کارگیری جاوا اسکریپت و XML برای مدیریت داده‌ها، از استاندارد پیشرفته‌ای برای اینترنتیس وب استفاده می‌کند و می‌توان حقیقتاً آن را یک فناوری مرتبط با اینترنتیس (و نه فقط تعامل میان کلاینت و سرور) تلقی کرد.

با این وجود، ای‌جکس معایبی نیز دارد و این معایب موضوع بحث بسیاری در محافل برنامه‌نویسی روی

اینترنت است. یکی از مسائلی که به عنوان ایراد مطرح شده، دشواری تغییر عادت کاربران در استفاده از کلیدهای Back و Forward و Refresh در مرورگرهای وب است. یکی از مشکلات برنامه‌نویسان وب همواره این است که یا باید کاربر را عادت دهند که هرچه کمتر از این دکمه‌ها استفاده کند و یا نرم‌افزار خود را

طوری بنویسند که اگر کاربر سهواً یا عمداً از این دکمه‌ها استفاده کرد، نرم‌افزار دچار اشتباه و خطا در تفسیر عمل کاربر نشود.

به عنوان مثال هنوز بسیاری از سایت‌های تجارت الکترونیکی که به کار فروش محصولات مشغولند هنگام طی شدن مراحل نهایی خرید آنلاین به کاربر هشدار می‌دهند که حین پردازش یک سفارش (یعنی درست در لحظه‌ای که فرمان نهایی از سوی کاربر ارسال شده است و هنوز صفحه نمایش پیام ثبت موفقیت‌آمیز سفارش یا عدم ثبت آن برای وی نمایش داده نشده) از فشردن کلید Refresh جدا پرهیز کنند وگرنه ممکن است از کارت اعتبار آنها دوبار پول کسر شود. همچنین استفاده از دکمه‌های Back و Forward در عملیاتی که به آسانی برگشت‌پذیر نیستند ممکن است باعث گیج شدن کاربر است.

مثلاً اگر نام‌های داخل صندوق پستی خود را پاک کنید، استفاده از دکمه Back هرگز این عمل را Undo نمی‌کند. همچنین اگر یک قلم کالا به سبد خرید آنلاین خود اضافه کنید، فشردن دکمه Back ممکن است در ظاهر چنین نشان دهد که آن قلم کالا مجدداً از سبد برداشته شده اما در سمت سرور همچنان در سبد خرید کاربر باشد. از آنجا که فناوری ای‌جکس عمل Navigation یا راهبری در یک سایت را به روندی غیرخطی تبدیل می‌کند، تمام این مشکلات به شکل حادثی ممکن است بروز کند.

در واقع با حضور Ajax، کارکرد سیستم History مرورگر به مساله‌ای بغرنج تبدیل می‌شود. زیرا برنامه‌نویس

یا باید با گنجاندن دکمه‌ها و فرامین اضافی، مکانیزم Undo را بازسازی کند و یا موتور ای‌جکس را طوری بنویسد که فشردن دکمه Back خود به خود موجب احضار فرمان Undo شود.

در هر دو صورت کار برنامه‌نویس آسان نخواهد بود. البته در این زمینه ترفندها و تکنیک‌هایی هم ابداع شده است. از جمله، استفاده از تگ IFRAME مخفی در اینترنت‌فیس صفحه که موتور ای‌جکس بتواند در صورت فشرده شدن دکمه Back از سوی کاربر، نسخه‌های پیشین نمایش داده شده از اینترنت‌فیس را از انبار History مرورگر بیرون بکشد و دوباره در چرخه عملیات موتور ای‌جکس وارد کند. این ترفند همانکون در سرویس Google Maps استفاده می‌شود. مشکل دیگری که در رابطه با ای‌جکس وجود دارد، بی معنی شدن مفهوم Bookmark است.

بازهم مکانیزم غیرخطی ای‌جکس مشکل ساز می‌شود. آیا شما می‌توانید در نرم‌افزارهای معمولی دسکتاپ یک لحظه مشخص را Bookmark کنید و دوباره به آن برگردید؟ بعضی از نرم‌افزارها مانند فتوشاپ (سیستم Action history) این کار را انجام می‌دهند، ولی اغلب نرم‌افزارها چنین نیستند. در واقع به سختی می‌توان از لحظات مختلف یک نرم‌افزار عکس گرفت و snapshot تهیه کرد.

این کار به یک انباره پیچیده نیاز دارد که هم حالت‌های مختلف اینترنت‌فیس و هم وضعیت‌های متفاوت داده‌ها را قبل و پس از تغییرات در خود نگه دارد. البته درمورد Ajax راهکارهایی در این زمینه پیشنهاد شده است. از جمله استفاده از anchor در HTML که پس از علامت در یک URL ظاهر می‌شود. از آنجایی که جاوا اسکریپت امکان به روزرسانی داینامیک آنکورها را دارد، بعضی برنامه‌نویسان پیشنهاد کرده‌اند که از این ترفند برای نگهداری حالات مختلف یک برنامه مبتنی بر ای‌جکس استفاده شود. تکنیکی که ممکن است در زمینه حل مشکل دکمه‌های Back و Forward نیز سودمند واقع شود.

یک مشکل عمده دیگر نیز در ارتباط با ای‌جکس وجود دارد. این فناوری به شدت متکی به XMLHttpRequest است و این شیء به دلایل امنیتی طی ماه‌های اخیر هرچه بیشتر و بیشتر در نسخه‌های جدید مرورگرها محدود شده است. زیرا اگر هر کلاینتی بتواند از هر نقطه‌ای به هر سروری این فرمان را بفرستد، آنگاه تهدیدهای امنیتی علیه سایت‌ها افزایش می‌یابد. محدودیت‌های جدید اعمال‌شده در نسخه‌های اخیر مرورگرها موجب شده که فرمان XMLHttpRequest به غیر از سایتی که صفحه وب از آنجا آمده است نتواند با سایت دیگری دیالوگ داشته باشد و این مسئله در تناقض با کاربرد ای‌جکس در زمینه وب‌سرویس است.

البته برای غلبه بر این مشکل راه‌حل‌هایی هم پیشنهاد شده است، از جمله این که شیء XMLHttpRequest می‌تواند تقاضای دیالوگ با سایت‌های دیگر را به یک وب سرویس روی سایتی که صفحه وب از آنجا آمده است بفرستد و این وب‌سرویس (که روی میزبان سایت قرار دارد و با تمام اینترنت در ارتباط است) به صورت یک واسطه عمل کند و تقاضاهای مورد نظر را برای سایت مقصد ارسال کند. این وب سرویس‌های واسطه اصطلاحاً Application Proxy نامیده می‌شوند.

البته ای‌جکس مشکلات کوچک و جزیی دیگری هم دارد که چندان مایه نگرانی نیست ولی به هر حال قابل لمس هستند. به عنوان نمونه، نرم‌افزارهای مبتنی بر ای‌جکس از حجم زیادی جاوا اسکریپت استفاده می‌کنند که همه اینها در هر session دست کم یک بار باید روی مرورگر بارگذاری شوند. بنابراین اولین باری که چنین اینترنت‌فیزی بارگذاری می‌شود صفحه وب آنقدر سنگین می‌شود که حتی با ارتباط باندپهن هم چند لحظه طول می‌کشد که صفحه بارگذاری شود.

در چنین شرایطی نوشتن یک موتور ای‌جکس هوشمند که با کمتر میزان کد بتواند بهترین کارکرد را داشته باشد، خود به یک چالش برنامه‌نویسی تبدیل می‌شود؛ ضمن این که کاربران وب در کشورهایی که سرعت دسترسی به اینترنت در آنها به طور معمول زیاد نیست باید هنگام بارگذاری صفحات وب این چنینی صبر پیشه کنند و برنامه‌نویسان نیز مراقب باشند تا در صورتی که به دلیل کندی خط یا قطع شدن‌ها لحظه‌ای آن، کدهای جاوا اسکریپت به طور کامل روی مرورگر بارگذاری نشد، نرم‌افزار دچار خطا و اشتباه نشود و بتواند این مشکلات را از طریق بارگذاری مجدد و هوشمندانه کد جاوا اسکریپت روی کلاینت، مدیریت کند.

جنبش Open Ajax هم از راه رسید

در کنار همه تحولاتی که به دنبال ظهور پدیده ای جکس در صنعت نرم افزار پدید آمده است، یک خبر مهم نیز قابل توجه است. شرکت IBM که خود از پیشگامان فناوری XML است، در ابتدای ماه فوریه 2006 اعلام کرد یک ائتلاف بزرگ صنعتی متشکل از شرکت های بزرگ، Zend، BEA Systems، ناول، یاهو، موزیلا، ردهت، اوراکل و بنیاد اکلیپس ایجاد کرده است که در نظر دارد برای توسعه یک پیاده سازی فراگیر و این سورس از ای جکس تلاش کند. پیش بینی می شود که این پیاده سازی به یکی از استانداردهای اصلی ای جکس در صنعت نرم افزار تبدیل شود. آی بی ام در نظر دارد مجموعه Toolkit پیشنهادی خود را از طریق

بنیادهای موزیلا و اکلیپس در دسترس جامعه این سورس قرار دهد. در همین رابطه شرکت Zimbra، یک سازنده نرم افزارهای مبتنی بر ای جکس در نظر دارد مجموعه Toolkit خود را تحت لیسانس موزیلا و Apache در اختیار برنامه نویسان قرار دهد.

در عین حال وقتی حجم جاوا اسکریپت بارگذاری شده روی کلاینت افزایش می یابد سرعت پردازش کامپیوتری که کاربر استفاده می کند نیز به یک مسئله تبدیل می شود. به بیان دیگر، ای جکس شدیداً منابع و resource های سیستم را به کار می گیرد، چنانکه به جرات می توان گفت نرم افزارهای مبتنی بر Ajax برای کاربرانی که از کامپیوترهای ضعیف و قدیمی استفاده می کنند قابل استفاده نیست.

اگر فرض را بر این بگذاریم که این یک مشکل عمومی است و نه مختص وب، یعنی کامپیوترهای قدیمی از اجرای نسخه جدید نرم افزارهای مخصوص دسکتاپ نیز ناتوانند تا چه برسد به وب، آنگاه این مشکل چندان اهمیتی ندارد زیرا به مرور زمان، با جایگزین شدن سیستم های جدید، مسئله حل خواهد شد. در چنین شرایطی ممکن است کندهای جاوا اسکریپت در نرم افزارهای مبتنی بر ای جکس عملاً مزیت کاهش زمان انتظار برای بارگذاری مجدد صفحه را خنثی کند و نقض غرض شود. خوشبختانه مرتباً پردازنده های سریع تری وارد بازار می شوند و می توان امیدوار بود که این مشکل قبل از آنکه جدی شود، ناپدید شود.

پیاده سازی Ajax

یکی دیگر از مشکلاتی که در ارتباط با ای جکس به نظر می رسد اینست که کار زیادی از برنامه نویسی می طلبد و در حال حاضر اجرای موفقیت آمیز یک اینترفیس حرفه ای مبتنی بر ای جکس کاری بسیار دشوار و وقت گیر است. یعنی Ajax به همان اندازه که کار بازدید کننده ها را آسان می کند، کار برنامه نویسان را مشکل می کند! با این حال ای جکس روی پلتفرم های مختلف پیاده سازی شده و مجموعه ابزارهای مختلفی برای تسهیل کار برنامه نویسان عرضه شده است تا تجربه شیرین تری از Ajax داشته باشند.

برنامه نویسان ASP.NET در صورتی که از نسخه دات نت 1.1 استفاده می کنند می توان از ماژول Ajax.NET استفاده کنند که اولین پیاده سازی دات نت ای جکس و رایگان است. مایکروسافت در ASP.NET 2.0 این فناوری را با نام Atlas وارد پلتفرم کرده است که علاقه مندان می توانند یکراست سراغ آن بروند.

برای پلتفرم جاوا نیز چندین پیاده سازی مختلف، اغلب این سورس، وجود دارد. از جمله AjaxAnywhere و AjaxTags را ببینید. همچنین چندین پیاده سازی مخصوص PHP می توانید پیدا کنید. ماژول های CPaint و Ajaxax آن جمله اند. ماژول CPaint برای ASP نیز قابل استفاده است. چند ماژول دیگر هم مانند GeneXus وجود دارند که مالتی پلتفرم هستند. برای هریک از پلتفرم های Perl، ColdFusion و Python نیز دست کم یک پیاده سازی معروف وجود دارد. فهرست کامل همه این پیاده سازی ها را به همراه آدرس سایت و لینک دانلود آنها را می توانید در ضمیمه آنلاین همین مقاله که همزمان با انتشار این شماره در بخش دانلود سایت ماهنامه شبکه قرار می گیرد، پیدا کنید.

بهروز نوعی پور

واژه **Ajax** با تلفظ <ای جکس> یا <ای ژاکس> سرنام عبارت XMLHttpRequest و Asynchronous JavaScript به معنی <ترکیب نامتقارن جاوا اسکریپت و XML> است. ماهیت صفحات وب و پروتکل HTTP به گونه‌ای است که به طور معمول وقتی در حال وب‌گردی هستیم، به ازای هر کنش و واکنش میان ما و سایتی که در حال کار با آن هستیم، کل یک صفحه وب از نو بارگذاری و تازه‌سازی (refresh) می‌شود.

ای جکس فناوری جدیدی است که تغییر محسوس را در این سناریو به وجود می‌آورد؛ به این ترتیب که به جای بارگذاری مجدد کل صفحه، فقط قسمتی تغییر می‌کند که قرار است اطلاعات جدید را به نمایش درآورد و کلیه عملیات ارسال اطلاعات و دریافت نتایج در پشت صحنه انجام می‌شود. در نتیجه هیچ‌گاه صفحه سفید و خالی وب در فواصل کنش و واکنش‌های هنگام کار با مرورگر دیده نمی‌شود و احساسی مشابه تجربه کار با یک نرم‌افزار دسکتاپ به کاربر دست می‌دهد.

جادوی ای جکس چیزی نیست جز یک فکر بکر و آن هم ترکیب کردن جاوا اسکریپت و XML در قالب یک موجود نرم‌افزاری جدید.

اصطلاح Ajax را اولین بار کارشناسی از شرکت Adaptive Path به نام جسی جیمزگرت در مقاله‌ای با عنوان <ای جکس؛ رهیافت جدیدی در برنامه‌های تحت وب> مطرح کرد و خیلی سریع مورد استقبال گسترده برنامه نویسان وب در سراسر جهان قرار گرفت. اعتقاد عمومی این است که تاریخچه به‌کارگیری تکنیک مذکور به پیدایش نرم‌افزار Outlook WebAccess و استفاده از فرمان XMLHttpRequest که مایکروسافت ابداع کرده و در نرم‌افزار اینترنت اکسپلور به‌کار رفته است، برمی‌گردد. اما امروزه اغلب مرورگرهای مهم و شناخته‌شده (از جمله فایرفاکس) از آن پشتیبانی می‌کنند و دیگر یک فناوری محدود به اینترنت اکسپلور نیست.

چرا Ajax این قدر اهمیت دارد؟

فناوری ای جکس جدید است و شگفت‌انگیز به نظر می‌رسد. ولی در حقیقت کل این شعبده بر اساس فناوری‌هایی بنا شده است که هم‌اکنون موجودند: جاوا اسکریپت و XML هر دوی این فناوری‌ها تا حد زیادی باز هستند و منحصر به شرکت خاصی نیستند. به همین دلیل این روزها تمام محافل دنیای برنامه نویسی مملو از مقالات و تحلیل‌هایی درباره **Ajax** است. به عنوان نمونه می‌توانید صفحه نخست سایت موسوم به <کانال شبکه برنامه نویسان سان> را باز کنید.

بیشتر این صفحه (در زمان نگارش این یادداشت) به مقالات و مطالب متنوعی درباره **Ajax** اختصاص یافته است. مجلات برنامه‌نویسی نیز مقالات متعددی در این زمینه منتشر کرده‌اند و سایت‌های معروفی مانند "O'Reilly" و "xml.com" نیز در این باره مطلب دارند.

این فناوری از یک جنبه دیگر نیز اهمیت دارد. به دلیل عدم نیاز به بارگذاری مجدد کل یک صفحه وب، مقدار داده‌هایی که لازم است برای تکمیل یک Interaction میان کاربر و سایت مبادله شود، به شدت کاهش می‌یابد و این به معنی افزایش محسوس سرعت نرم‌افزارهای تحت وب، سهولت به‌کارگیری اینترفیس‌های مبتنی بر ای جکس و کاربرپسندتر شدن آن‌ها می‌شود. به همین دلیل این روزها اکثر پورتال‌های بزرگ (مانند یاهو) مشغول طراحی اینترفیس‌های مبتنی بر **Ajax** هستند.

با این همه مزایای ای جکس به همین موارد ختم نمی‌شود. تصور کنید وقتی در طول شبانه روز میلیاردها بار فرآیند بارگذاری مجدد صفحات وب تعاملی در شبکه اینترنت تکرار می‌شود، استفاده از ای جکس چگونه موجب آزاد شدن پهنای‌بند اینترنت و در نتیجه میلیاردها دلار صرفه‌جویی اقتصادی در این زمینه می‌شود و این به نوبه خود انقلابی در عرصه وب و فضای سایبر به شمار می‌رود. این تحول در شرایطی اتفاق می‌افتد که همزمان فناوری‌های اینترنت پرسرعت همچون ADSL به شدت در حال رشد هستند.

حال تصور کنید اگر فناوری ای جکس در مقیاس گسترده‌ای مورد استفاده نرم‌افزارهای تحت وب قرار گیرد. آنگاه افزایش سرعت کار با اینترنت چند برابر خواهد شد. چنین تحولی می‌تواند اینترنت را به کامپیوتر دوم کاربران تبدیل کند. به گونه‌ای که برای آن‌ها اجرای یک نرم‌افزار از روی وب تفاوت محسوسی با اجرای آن از روی کامپیوتر دسکتاپ نداشته باشد. در این صورت ممکن است واقعا بخش مهمی از توان پردازشی نرم‌افزارها، چه از نوع دسکتاپ و چه از نوع تحت وب به سیستم‌های موسوم به Web Service واگذار شود.

از این رو کسانی که مبتکر و مشوق ایده وب سرویس بوده‌اند، این روزها انگیزه تازه‌ای برای سرعت بخشیدن به این فناوری پیدا کرده‌اند و تلاش می‌کنند دو فناوری ای جکس و وب سرویس را به یکدیگر پیوند دهند و راهکارهای تازه‌ای بیافرینند.

کدام سایت ها از ای جکس استفاده کرده اند؟

دنیا برنامه نویسی وب هنوز درگیر هیجانات مربوط به آشنایی با این پدیده است و در حال بررسی مشکلات تکنیکی و هضم و جذب آن در بافت برنامه های تحت وب می باشد. به همین دلیل میزان استفاده عملی از آن چندان گسترده نیست. اما چون کارایی ای جکس دیگر برای همه ثابت شده است، حرکت های بزرگی در سراسر اینترنت به سمت استفاده از این فناوری در جریان است. از جمله اینترنتیس نسخه دوم Yahoo Mail که به نسخه بتا معروف است و فعلا در دسترس کاربران سرویس غیر رایگان یاهو قرار دارد، از همین فناوری استفاده می کند که به زودی اینترنتیس جدید در دسترس عموم قرار می گیرد. سرویس Gmail و Google Map نیز از این فناوری استفاده می کنند.

جهت دریافت نمونه برنامه ها و مقالات **ajax** به آدرس زیر مراجعه
نمائید: <http://www.zamanphp.blogspot.com/?PostID=34>

IranPHP

Ajax یا جاوا اسکریپت و xml غیر همزمان (Asynchronous Javascript and XML)، جدیدترین همه چه موجود در برنامه نویسی وب است. با استفاده از شیء XMLHttpRequest، HTML، و JavaScript و CSS، می توانید refresh صفحه را که (تا چند ماه پیش) یک بخش اجباری از ارسال داده به سمت سرور بود، عوض کنید.

در این مقاله قصد داریم تا 3 کاربرد عملی و ساده از Ajax بر روی سرور را به شما نشان دهیم. برای ادامه نیاز است که تا حدی با HTML، و Javascript و PHP آشنا باشید. با این حال من سعی می کنم که همه چیز را به شکل ساده بیان کنم تا چیزی را از دست ندهید.

توجه: در این مقاله فرض می کنیم که شما یک وب سرور PHP را در حال اجرا دارید.

فعال سازی ارتباطات کلاینت-به-سرور

2 تابع ساده لازم است تا Ajax کار کند. آندو همه آنچه را که برای انتقال داده به سرور و همچنین پردازش پاسخ فرستاده شده از سرور لازم است، کنترل می کنند. ما یک فایل js که شامل این دو تابع است می سازیم که در هر 3 مثال Ajax ما استفاده خواهد شد.

یک فایل جدید در ادیتور متن دلخواه خود ایجاد کنید و آنرا ajax.js بنامید. کد زیر را کپی و در آن فایل paste کرده و آنرا save کنید:

```
function DoCallback(data) {  
    // branch for native XMLHttpRequest object  
    if (window.XMLHttpRequest) {  
        req = new XMLHttpRequest();  
        req.onreadystatechange = processReqChange;  
        req.open('POST', url, true);  
        req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');  
        req.send(data);  
        // branch for IE/Windows ActiveX version  
    }  
    else if (window.ActiveXObject) {  
        req = new ActiveXObject('Microsoft.XMLHTTP');  
        if (req) {  
            req.onreadystatechange = processReqChange;  
            req.open('POST', url, true);  
            req.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
```

```
urlencoded');
    req.send(data);
  }
}
```

```
function processReqChange() {
  // only if req shows 'loaded'
  if (req.readyState == 4) {
    // only if 'OK'
    if (req.status == 200) {
      eval(what);
    }
    else {
      alert('There was a problem retrieving the XML data: ' +
req.responseText);
    }
  }
}
```

حالا که فایل is تان را ایجاد کرده اید، بیایید به اولین کاربرد Ajax در وب سایتتان می پردازیم.

کاربرد شماره 1 اجکس: تایید اعتبار یک login

بیایید ببینم چگونه می توانید بجای بازگشت دوباره به سرور جهت تایید اعتبار نام کاربری/رمز عبور برای یک فرم login از Ajax استفاده کنید.

بیاید با ایجاد اسکریپت PHP که Ajax با آن "صحبت" خواهد کرد، شروع کنیم. یک فایل به نام login.php بر روی وب سرور در همان دایرکتوری که اسکریپت ajax.js را قبلا ایجاد کرده بودیم، بسازید. کد PHP زیر را در فایل login.php کپی و paste کرده و آنرا save نمایید:

```
<?php
if (isset($_POST["username"]) && isset($_POST["password"])) {
    $username = $_POST["username"];
    $password = $_POST["password"];
    if ($username = "user" && $password == "password")
        echo 1;
    else
        echo 0;
}
else {
    echo 0;
}
?>
```

توجه: واضح است که این مثال خیلی ساده شده است. در حالت معمول شما از یک بانک اطلاعات MySQL برای ذخیره login کاربران و مراحل از رمزگذاری برای ارسال داده از کلاینت به سرور استفاده می کنید.

در مرحله بعد بیایید صفحه login خود را بسازیم. یک صفحه جدید با نام login.html ایجاد کنید و کد زیر را در آن کپی و paste نمایید:

```
<html>
<head>
```

```

<script>

var url = "login.php";
var what = "LoginStatus(req.responseText)";
function CheckLogin() {
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;
    DoCallback("username=" + username + "&password=" + password);
}

function LoginStatus(Status) {
    if(Status == 0)
        alert("Bad login!");
    else
        alert("Login OK!");
}
}
</script>
<script src="ajax.js" type="text/javascript"></script>
</head>
<body>
<pre>Username: <input id="username" type="text"><br>
Password <input id="password" type="password"><br><br>
<input type="button" value="Check Login" onClick="CheckLogin()">
</pre>
</body>
</html>

```

برای تست اولین مثالمان، کلمه "user" را در فیلد نام کاربری و کلمه "password" را (بدون گیومه) در فیلد رمز عبور وارد کنید و دکمه "CheckLogin" را کلیک نمایید.

ابتدا ما تابع DoCallback را با پاس دادن مقادیر فیلدهای نام کاربری و رمز عبور فراخوانی می کنیم، شبیه زیر:

```
DoCallback("username="+username+"&password="+password);
```

تابع DoCallback از Ajax برای فراخوانی صفحه login.php با مقادیر نام کاربری و رمز عبور گرفته شده از فرم، استفاده می کند. 2 تا متغیر در بالای فایل login.html تعریف شده است که به تابع Ajax می گوید که چه کار کند:

```

var url = "login.php";
var what = "LoginStatus(req.responseText)";

```

متغیر url به تابع Ajax می گوید که داده هایش را کجا بفرستد و متغیر what به Ajax می گوید که چه تابعی باید پس از آماده شدن سرور برای فرستادن داده ها فراخوانی شود. متغیر req.responseText حاوی مقادیر بازگشت داده شده توسط سرور خواهد بود که در این حالت 0 برای login ناموفق و 1 برای login موفق است. همانگونه که در اسکریپت login.php نشان داده شده است:

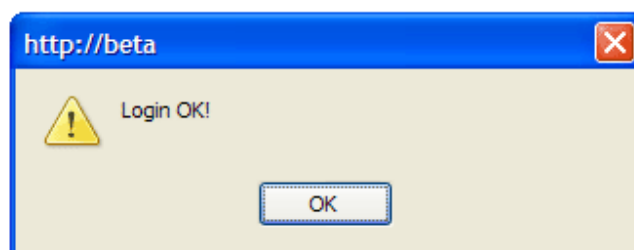
```

if($username = "user" && $password == "password") echo 1;
else echo 0;

```

Username:

Password:



کاربرد شماره 2 اجکس Drop Down: لیست کشور و ایالت ها

بیشتر فروشگاههای eCommerce بر روی اینترنت، به منظور محاسبه هزینه حمل و نقل از شما می خواهند تا هنگام سفارش دادن، کشور و ایالت خود را انتخاب نمایید. به جای لود شدن دوباره صفحه بعد از انتخاب کشور به منظور بازیابی لیست ایالتها، می توانیم از اجکس برای انجام چنین کاری استفاده کنیم، اما بدون refresh شدن صفحه.

یک صفحه جدید به نام country.php ایجاد کرده و کد زیر را کپی و در آن paste نمایید:

```
<?php
$states = array();
$states["Australia"] = array("New South Wales", "Queensland",
"Victoria",
"South Australia", "Tasmania", "Western Australia", "Northern
Territory");
$states["USA"] = array("California", "New York", "Ohio",
"Colorado", "Florida",
"Texas", "Arizona", "Washington");
if (isset($_POST["country"]) && isset($states[$_POST["country"]]))
{
    foreach($states[$_POST["country"]] as $state) {
        printf("%s,", $state);
    }
}
?>
```

PHP بالا یک آرایه از ایالت ها ایجاد می کند: یکی برای Australia و دیگری برای USA. زمانیکه یک کشور به اسکریپت فرستاده شد (از طرف تابع اجکس با استفاده از جاوا اسکریپت)، ایالتها مربوطه به شکل یک رشته string که با کاما از هم جدا شده اند، بازگشت داده می شوند؛ نظیر زیر:

state1,state2,state3

فایل HTML ما country.html خواهد بود که شبیه زیر است:

```
<html>
<head>
<script>
    var url = "country.php";
    var what = "SetStates(req.responseText)";
    function GetStates(Country) {
        DoCallback("country=" + Country);
    }
}
```



```

function SetStates(States) {
    var stateBox = document.getElementById("state");
    stateBox.options.length = 0;
    if(States != "") {
        var arrStates = States.split(",");
        for(i = 0; i < arrStates.length; i++) {
            if(arrStates[i] != "") {
                stateBox.options[stateBox.options.length] = new
Option(arrStates[i], arrStates[i]);
            }
        }
    }
}
</script>
<script src="ajax.js" type="text/javascript"></script>
</head>
<body>
<pre>Country:
<select onChange="GetStates(this.options[this.selectedIndex].text)"
id="country">
<option value=""></option>
<option>Australia</option>
<option>USA</option>
</select><br>State: <select id="state"></select>
</pre>
</body>
</html>

```

زمانیکه شما یک کشور انتخاب می کنید، تابع Ajax یک لیست از ایالت‌های آن کشور را از فایل country.php بازگشت می دهد. سپس لیست ایالت‌ها را به تابع SetStates ما پس می فرستد:

```

var url = "country.php";
var what = "SetStates(req.responseText)";

```

Country:

State:

- California
- New York
- Ohio
- Colorado
- Florida
- Texas
- Arizona
- Washington

کاربرد شماره 3 اجکس: نتایج جستجو

آخرین مثال من درباره Ajax نشان می دهد که چگونه می توان لیستی از نتایج جستجو از سمت سرور را نشان داد. بعد اینکه [Google Suggest](#) را در عمل دیدم، فکر کردم که این می تواند ایده خوبی برای انجام کارهای مشابه باشد.

برای شروع بایید اسکریپت PHP مان را بسازیم که نتایج جستجو را برگشت می دهد. چون می خواهیم همه چیز ساده باشد، ما فقط یک آرایه اصلی ساخته و از یک الگوریتم ساده استفاده می کنیم. کد زیر را به یک فایل جدید با نام search.php اضافه کنید:

```
<?php
$results = array();
$links = "";
$results["http://www.cats.com"] = array("cat", "cats", "kitten",
"kittens", "pet", "pets");
$results["http://www.google.com"] = array("search", "search engine",
"google");
$results["http://www.mtv.com"] = array("music", "tv", "music
charts");
if(isset($_POST["query"])) {
    $q = strtolower($_POST["query"]);
    $arrQueries = explode(" ", $q);
    foreach($arrQueries as $query) {
        foreach($results as $link=>$keywords) {
            foreach($keywords as $word) {
                if(strtolower($word) == $query) {
                    if(!is_numeric(strpos($links, $link)))
                        $links .= sprintf("%s,", $link);
                }
            }
        }
    }
    echo $links;
}
?>
```

من یک آرایه ساختگی از لینک ها و کلید واژه ها ساخته ام و بسته به اینکه چه کلماتی در textbox تایپ شده است، لینک های مرتبط بازگشت داده می شوند.

حال بایید فایل HTML مان را بسازیم. آنرا search.html بنامید و کد HTML زیر را در آن کپی نمایید:

```
<html>
<head>
<style>
    .box {
        border: solid 1px black;
        width: 200px;
        height: 80px;
        background-color: lightyellow;
        padding: 10px;
        margin-top: 5px;
        display: none; }
</style>
<script>
    var url = "search.php";
    var what = "ShowResults(req.responseText)";
    function GetResults(Query) {
        DoCallback("query="+Query);
    }
    function ShowResults(Links) {
        var results = document.getElementById("results");
        if(Links != "") {
            results.style.display = "block";
            results.innerHTML = "";
```

```

arrLinks = Links.split(",");

for(i = 0; i < arrLinks.length; i++) {
    if(arrLinks[i] != "")
        results.innerHTML = results.innerHTML + "<a href='" +
arrLinks[i] + ">"
            + arrLinks[i] + "</a><br>";
    }
}
else {
    results.style.display = "none";
}
}
</script>
<script src="ajax.js" type="text/javascript"></script>
</head>
<body> Search:
<input id="query" type="text"
onKeyUp="GetResults(document.getElementById('query').value)">
<div id="results" class="box">
</div>
</body>
</html>

```

مهمترین بخش از کد HTML ما، textbox جستجو است. چیزی شبیه این:

```

Search:
<input id="query" type="text"
onKeyUp="GetResults(document.getElementById('query').value)">

```

وقتی که چیزی در textbox تایپ شده است، رویداد onKeyUp رخ داده شده است که تابع جاوا اسکریپتی GetResults را فراخوانی می کند.

```

function GetResults(Query) {
DoCallback("query="+Query);
}

```

هر چیزی که در textbox تایپ شده باشد، به اسکریپت search.php که قبلاً ایجاد کرده بودیم، فرستاده می شود. query جستجو توسط توسط space از هم تفکیک شده است، و هر کلمه بر روی یک لیست ساختگی از نتایج جستجو چک می شود.

```

$results["http://www.cats.com"] = array("cat", "cats", "kitten",
"kittens", "pet", "pets");
$results["http://www.google.com"] = array("search", "search engine",
"google");
$results["http://www.mtv.com"] = array("music", "tv", "music charts");

```

اگر query جستجو نتیجه ای از کلمات موجود در براکت ها داشته باشد، لینک آن واژه به browser پس فرستاده می شود. وقتی که همه نتایج جستجو چک شد، لینک ها به شکل یک جعبه نتایج جستجوی زیبا نشان داده می شوند -- که همه این ها به صورت بلادرنگ (realtime) انجام می پذیرد.

Search:

<http://www.cats.com>
<http://www.google.com>
<http://www.mtv.com>

نتیجه گیری

در این مقاله به شما 3 روش عملی برای استفاده از Ajax برای بهبود سودمندی وب سایتتان نشان دادم. به خاطر داشته باشید که مباحثی نظیر امنیت و بار سرور، در تصمیم گیری درباره اینکه کجا و چگونه از اجکس بر روی وب سایتتان استفاده کنید، نقشی ندارند.

اگر مایلید تا بیشتر درباره Ajax بدانید، همه کد های مثال های گفته شده در این مقاله را به صورت کامل مطالعه کرده و لینک های زیر را نیز بررسی نمایید:

http://dhtmlnirvana.com/ajax/ajax_tutorial/
<http://www.sitepoint.com/article/remote-scripting-ajax>
http://en.wikipedia.org/wiki/Ajax_%28programming%29

مهارت در آجاکس، قسمت اول: مقدمه ای بر آجاکس

فراگیری Ajax، روشی مؤثر برای ایجاد وب سایت ها و چگونگی کارکرد آن

Ajax که ترکیبی از HTML، تکنولوژی جاوا اسکریپت، DHTML و DOM است، یک روش عالی است که به شما کمک می کند تا UI های سنگین وب را به برنامه های Ajax اینترنتیو تبدیل نمایید. نویسندگان در این سری مقالات نشان می دهد که چگونه این تکنولوژی ها با هم کار می کنند - از مقدمه تا یک نگاه دقیق - تا توسعه وب به صورت مؤثر به یک واقعیت آسان تبدیل شود. وی همچنین مفاهیم پایه Ajax شامل شیء XMLHttpRequest را تشریح می کند.

پنج سال پیش اگر چیزی از XML نمی دانستید، شما یک جوجه اردک زشت بودید که هیچ کس حاضر نبود با شما صحبت کند. هجده ماه پیش، زمانی که Ruby به صحنه آمد، هیچ کس برنامه نویسی را که نمی دانست چه چیزی به همراه Ruby آمده است، تحویل نمی گرفت. امروزه اگر می خواهید وارد آخرین موج تکنولوژی شوید، Ajax جایی برای این کار است.

Ajax بسیار فراتر از یک مد زود گذر است؛ در واقع یک روش قدرتمند برای ساختن وب سایت ها است که یادگیری آن به عنوان یک زبان جدید خیلی سخت نیست.

قبل از اینکه وارد این بحث شوم که Ajax چیست، بیاید چند دقیقه ای به این موضوع بپردازیم که Ajax چه کاری را انجام می دهد. امروزه هنگامیکه می خواهید یک برنامه بنویسید، شما دو انتخاب اصلی دارید:

- برنامه های Desktop
- برنامه های وب

هر دو اینها آشنا هستند؛ برنامه های desktop معمولاً بر روی یک CD یا بعضی اوقات قابل دانلود شدن از وب سایت هستند و به صورت کامل بر روی کامپیوتر شما نصب می شوند. آنها ممکن است جهت به روز رسانی از اینترنت هم استفاده کنند اما کدی که برنامه را اجرا می کند بر روی کامپیوتر شما باقی می ماند. برنامه های وب جایی بر روی یک سرور اجرا می شوند و شما با استفاده از browser تان می توانید به آنها دسترسی داشته باشید.

مهم تر از اینکه کد برنامه کجا اجرا می شود، این است که برنامه چگونه رفتار می کند و شما چگونه می توانید با آن کار کنید. برنامه های desktop معمولاً به خوبی سریع هستند (آنها بر روی کامپیوتر شما اجرا می شوند و برای اتصال به اینترنت نیازی به صبر کردن نیست)، رابط کاربری (UI) قوی دارند (معمولاً با سیستم عامل در ارتباطند) و به طور باور نکردنی داینامیک هستند. شما می توانید بدون نیاز به صبر کردن کلیک کنید، با موس اشاره کنید، تایپ کنید و ...

در عوض برنامه های وب معمولاً در هر لحظه به روز هستند و سرویس هایی را ارائه می دهند که هرگز نمی توانید بر روی desktop تان داشته باشید (Amazon.com و eBay را تصور کنید). اما به همراه توانایی های وب، انتظار نیز آمده است - انتظار برای پاسخ سرور، انتظار برای refresh شدن صفحه، انتظار برای اینکه یک درخواست فرستاده شود و صفحه جدیدی تولید شود.

بدیهی است که تا حدودی زیادی ساده سازی شده است اما شما نکته اصلی را بگیرید. همانگونه که ممکن است حدس زده باشید، Ajax سعی دارد تا پلی میان فاصله موجود بین عملیاتی و اینترنتی بودن یک برنامه desktop و همیشه به روز بودن برنامه های وب باشد. شما می توانید از UI های داینامیک و کنترل های زیبا شبیه آنچه در یک برنامه desktop می بینید، در برنامه های وب استفاده کنید.

خوب، منتظر چه هستید؟ به Ajax نگاه کنید و ببینید که چگونه UI های سنگین شما را به برنامه های پاسخی Ajax تبدیل می کند.

تکنولوژی قدیم، مهارت جدید

واقعیت این است که به همراه Ajax تکنولوژی های متعددی هم آمده است - برای اینکه از مبانی پایه فراتر بروید، نیاز دارید که وارد جزئیات تکنولوژی های مختلف شوید (و این دلیلی است که من چند مقاله اول خود را به تفکیک درباره هر کدام از آنها نوشته ام). خبر خوش این است که شما ممکن است در حد مناسبی با بعضی از این تکنولوژی ها آشنا باشید و از این بهتر اینکه یادگیری بیشتر این تکنولوژی ها آسان است. قطعاً به سختی یک زبان کامل نظیر Java یا Ruby نیست.

تکنولوژی های پایه موجود در برنامه های Ajax عبارتند از:

- HTML که به منظور ساختن فرم های وب و شناسایی فیلد های مورد استفاده در برنامه استفاده شده است .
- کدهای جاوا اسکریپت که کدهای هسته اجرایی برنامه های Ajax هستند و در تسهیل برقراری ارتباط با برنامه های سمت سرور کمک می کنند .
- HTML یا DHTML داینامیک به شما کمک می کند تا به صورت داینامیک فرم های تان را update نمایید. شما از div ، span و سایر المانهای HTML داینامیک به منظور علامت گذاری سند HTML تان استفاده می کنید .
- DOM (یا Document Object Model از طریق کد جاوا اسکریپت) به منظور کار با ساختار HTML شما و (در بعضی موارد XML (برگشتی از سمت سرور مورد استفاده قرار می گیرد .

بیا باید اینها را تقسیم بندی کنیم و ببینیم هرکدام چه کار می کنند. من در مقالات بعدی خود، هرکدام از آنها را تشریح خواهم کرد؛ اکنون بر روی آشنایی با این کامپوننت ها و تکنولوژی ها تمرکز می کنیم. هر قدر شما با این کد بیشتر آشنا شوید، حرکت از دانش موقتی درباره این تکنولوژی ها به سوی حرفه ای شدن در هر کدام از آنها آسان تر خواهد بود.

شیء XMLHttpRequest

اولین شئی که قرار است یاد بگیرید احتمالاً جدید ترین آنهاست؛ که XMLHttpRequest نامیده می شود. این شیء یک آبجکت جاوا اسکریپت است که به همان سادگی که در لیست زیر نشان داده شده است، ایجاد می شود:

لیست 1- ایجاد یک شیء جدید XMLHttpRequest

```
<script language="JavaScript" type="text/JavaScript">
var xmlhttp = new XMLHttpRequest();
</script>
```

من در مقاله بعدی خودم درباره این شیء بیشتر توضیح خواهم داد، اما برای اکنون توجه داشته باشید که این آبجکتی است که همه ارتباطات سمت سرور شما را مدیریت می کند. قبل از اینکه ادامه دهید، مکث کرده و درباره آن فکر کنید. این تکنولوژی جاوا اسکریپت است که از طریق شیء XMLHttpRequest با سرور ارتباط برقرار می نماید. چنین چیزی جزء جریان معمول برنامه نیست و جایی است که Ajax جادوی خود را بیشتر نمایان می کند.

در یک برنامه معمولی وب، کاربران فیلد های فرم را پر کرده و بر روی دکمه ارسال کلیک می کنند. سپس کل فرم به سرور فرستاده می شود، سرور اسکریپتی را پردازش می کند (معمولاً PHP یا Java یا شاید یک پروسه CGI یا کارهای مشابه)، و زمانیکه اسکریپت اجرا شد، سرور یک صفحه کاملاً جدید را پس می فرستد. آن صفحه می تواند HTML با یک فرم به همراه داده های پر شده یا یک تایید یا شاید صفحه ای با گزینه های انتخاب شده بر اساس داده های ارسال شده در فرم اصلی باشد. البته زمانیکه اسکریپت یا برنامه در سمت سرور در حال پردازش و فرستادن فرم جدید است، کاربران مجبورند صبر نمایند. صفحه نمایش خالی می شود و زمانیکه داده ها از سمت سرور برگشت داده شدند، صفحه دوباره ترسیم می شود. این همان جایی است که تعامل پایین نمود پیدا می کند -- کاربران بازخورد فوری دریافت نمی کنند و بدون شک همانند کار با برنامه های desktop راحت نمی باشند.

Ajax ذاتاً تکنولوژی جاوا اسکریپت و شیء XMLHttpRequest را ما بین فرم های وب شما و سرور قرار می دهد. هنگامی که کاربران فرم ها را پر می کنند، آن داده ها مستقیماً به سرور فرستاده نمی شوند بلکه به کدهای جاوا اسکریپت فرستاده می شوند. در عوض کدهای جاوا اسکریپت داده های فرم را گرفته و یک درخواست به سرور ارسال می کنند. وقتی چنین چیزی اتفاق می افتد، دیگر فرم نمایش داده شده به کاربر، رفت و برگشت ندارد و پنهان و پیدا نمی شود. به عبارت دیگر، کد جاوا اسکریپت درخواست را در پس زمینه ارسال می کند؛ کاربر حتی متوجه نمی شود که ارسال درخواست شروع شده است. به عبارتی بهتر، درخواست به صورت غیر همزمان (Asynchronously) ارسال شده است، به این معنی که کد جاوا اسکریپت شما (و کاربر) منتظر پاسخ سرور نمی مانند. بنابراین کاربران می توانند به ادامه ورود اطلاعات، مرور صفحه و استفاده از برنامه بپردازند.

سپس سرور داده هایی را به کد جاوا اسکریپت شما (که همچنان جانشین وب فرم است) پس می فرستد. کد جاوا اسکریپت تصمیم می گیرد که با آن داده ها چه کند. می تواند فیلدهای فرم را همان موقع update کند، که به برنامه شما احساس آنی بودن می دهد -- کاربران بدون اینکه فرمشان را ارسال یا refresh کرده باشند، داده های جدیدی دریافت می کنند. کد جاوا اسکریپت حتی می تواند داده ها را دریافت کرده، محاسباتی بر روی آن انجام داده و درخواست دیگری را ارسال کند، و همه اینها بدون دخالت کاربر است. چنین چیزی قدرت XMLHttpRequest است که می تواند هر آنچه را که می خواهد در سمت سرور یا کلاینت، با سرور مبادله نماید، حتی بدون آنکه کاربر از اتفاقاتی که در حال انجام است باخبر شود. نتیجه، تجربه ای دینامیک، پاسخگو، با تعامل بالا، شبیه یک برنامه desktop است، اما همه با نیروی اینترنت پشت آن.

افزودن چند خط جاوا اسکریپت

زمانیکه شما XMLHttpRequest را به کار می برید، باقی کد جاوا اسکریپت شما خیلی پیش پا افتاده می شود. در واقع، شما کد جاوا اسکریپت را تنها برای کارهای اساسی زیر استفاده خواهید کرد:

- دریافت داده های فرم: کد جاوا اسکریپت، بیرون کشیدن داده ها از فرم HTML شما و فرستادن آنها به سرور را آسان می نماید.
- تغییر مقادیر بر روی فرم: کد جاوا اسکریپت همچنین به روز رسانی یک فرم، از ست کردن مقادیر فیلدها تا جایگزین کردن تصاویر را آسان می نماید.
- تجزیه (Parse) کردن HTML و XML: شما از کد جاوا اسکریپت به منظور دستکاری DOM بخش بعد را ببینید) و کار با ساختار فرم HTML و هر داده XML ی که سرور بازگشت می دهد، استفاده خواهید کرد.

برای دو مورد اول، شما باید به خوبی با متد () `getElementById` ، آنچنان که در لیست 2 نشان داده شده است، آشنا باشید.

لیست 2. گرفتن و ست کردن مقادیر فیلد با استفاده از کد جاوا اسکریپت

```
// Get the value of the "phone" field and stuff it in a variable called phone
var phone = document.getElementById("phone").value;

// Set some values on a form using an array called response
document.getElementById("order").value = response[0];
document.getElementById("address").value = response[1];
```

عملا در اینجا هیچ چیز قابل توجهی وجود ندارد و این خوب است! شما باید درک کنید که هیچ چیز پیچیده و ترسناکی درباره این وجود ندارد. زمانیکه شما XMLHttpRequest را خوب یاد گرفتید، بقیه برنامه Ajax شما کدهای ساده جاوا اسکریپت شبیه آنچه در لیست 2 نشان داده شده خواهد بود، که با کمی HTML، هوشمندانه مخلوط شده است. پس بعضی اوقات کمی DOM مؤثر واقع می شود... بیایید نگاهی به آن بیاندازیم.

نازک کاری با DOM

آخرین مورد از لحاظ ترتیب، DOM یا Document Object Model است. برای بعضی از شما، DOM کمی ترسناک به نظر می آید -- اغلب توسط طراحان HTML استفاده نمی شود و حتی برای کد نویسان جاوا اسکریپت کمی غیر معمول است، مگر اینکه شما وارد موارد حرفه ای برنامه نویسی شده باشید. جایی که شما DOM را در برنامه های سنگین Java و C/C++ خواهید یافت، در حقیقت همان جایی است که DOM در سختی و مشکل بودن یادگیری کمی شهرت پیدا کرده است.

خوشبختانه استفاده از DOM در تکنولوژی جاوا اسکریپت ساده و بیشتر قابل درک است. من به شما نشان خواهم داد که چگونه از DOM استفاده کنید یا حداقل چند نمونه کد به شما ارائه خواهم کرد، حتی اگر به نظر گمراه کننده آید. خواهید فهمید که می توانید بدون اینکه با DOM کار کنید، به خوبی وارد Ajax شوید و آن مسیری است که من قصد دارم به شما نشان دهم. من در مقالات بعدی دوباره به DOM می پردازم اما برای اکنون فقط بدانید که خارج از این بحث است. زمانیکه شما شروع به انتقال XML ما بین کد جاوا اسکریپت و سرور کرده و فرم HTML را تغییر دادید، دوباره در DOM عمیق می شوید. برای اکنون انجام بعضی پیشرفت های مؤثر Ajax بدون DOM ممکن است، پس فعلا آن را کنار بگذارید.

بدست آوردن یک شیء Request

با مقدمه پایه ای که داشتید، وقت آن رسیده که نگاهی به جزئیات داشته باشید. از آنجاییکه XMLHttpRequest محور اصلی برنامه های Ajax است -- و شاید برای بسیاری از شما جدید باشد -- من از آنجا شروع می کنم. همانگونه که در لیست 1 دیدید، ایجاد و استفاده از این شیء بسیار ساده است، اینطور نیست؟ کمی صبر کنید.

نبرد رنج آور مرورگرها را در چند سال پیش به خاطر بیاورید و اینکه چطور هیچ چیزی در مرورگرها به یک شکل کار نمی کرد. البته ممکن است شما اعتقاد نداشته باشید ولی آن جنگ ها هنوز ادامه دارند، البته در یک مقیاس بسیار کوچکتر. و غافلگیری XMLHttpRequest: یکی از قربانیان این نبرد است. بنابراین شما برای به دست آوردن یک شیء XMLHttpRequest نیاز دارید که کارهای متفاوتی را انجام دهید. من این کار را گام به گام به شما نشان خواهم داد.

کار با مرورگرهای مایکروسافت

مرورگر مایکروسافت یا Internet Explorer، برای کار با XML از MSXML Parser استفاده می کند (برای اطلاعات بیشتر درباره MSXML به بخش منابع این مقاله رجوع کنید). بنابراین زمانیکه شما برنامه های Ajax می نویسید که نیاز دارد بر روی IE کار کند، باید این شیء را به روش خاصی ایجاد کنید. اما این کار آسان نیست. در حقیقت MSXML، بسته به اینکه چه ورژنی از تکنولوژی جاوا اسکریپت بر روی IE نصب شده است، دو ورژن متفاوت دارد، بنابراین شما مجبورید برای کنترل هر دو روش کد بنویسید. برای دیدن کد لازم جهت ایجاد یک شیء XMLHttpRequest بر روی مرورگرهای مایکروسافت، لیست 3 را ببینید.

لیست 3. ایجاد یک شیء XMLHttpRequest بر روی مرورگرهای مایکروسافت

```
var xmlhttp = false;
try {
    xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
    try {
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (e2) {
        xmlhttp = false;
    }
}
```

هنوز همه این کد ها قابل قبول به نظر نمی رسد اما خوب است. شما قبل از اینکه این سری مقالات تمام شود، در برنامه نویسی جاوا اسکریپت، کنترل خطا، کامپایل شرطی و غیره وارد خواهید شد. اکنون شما باید دو خط اصلی را مد نظر داشته باشید:

```
xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
```

9

```
xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
```

بطور خلاصه این کد سعی می کند تا شیء را با استفاده از یکی از ورژن های MSXML ایجاد کند؛ اگر انجام نشد با استفاده از ورژن دیگر شیء را ایجاد می کند. خوب است نه؟ اگر هیچ کدام از آنها کار نکنند، متغیر xmlhttp مقدارش false می شود تا به کد شما بفهماند که یک چیزی کار نکرده است. اگر اینگونه باشد، شما احتمالاً با یک مرورگر غیر مایکروسافتی مواجه هستید و نیاز است برای انجام این کار از کد متفاوتی استفاده نمایید.

کار با مرورگرهای Mozilla و غیر مایکروسافتی

اگر مرورگر IE انتخاب شما نیست یا برای مرورگرهای غیر مایکروسافتی کد نویسی می کنید، به کد متفاوتی نیاز دارید. در حقیقت یک خط کد خیلی ساده است که قبلاً در لیست 1 دیده بودید:

```
var xmlhttp = new XMLHttpRequest object;
```

این خط کد به روشی خیلی ساده تر یک شیء XMLHttpRequest را در مرورگرهای Mozilla، فایرفاکس، Opera، Safari و تقریباً همه مرورگرهای غیر مایکروسافتی که Ajax را به هر فرم یا روشی پشتیبانی می کنند، می سازد.

کنار هم گذاشتن اینها

هدف، پشتیبانی از همه مرورگرها است. چه کسی حاضر می شود برنامه ای بنویسد که فقط برای IE کار کند یا برنامه ای که فقط بر روی مرورگرهای غیر مایکروسافتی کار کند؟ بدتر اینکه آیا شما می خواهید برنامه تان را دو بار بنویسید؟ البته که نه! بنابراین کد شما پشتیبانی از IE و مرورگرهای غیر مایکروسافتی را ترکیب می کند. لیست 4، کد لازم جهت انجام این کار را نشان می دهد.

لیست 4. ایجاد یک شیء XMLHttpRequest به روش چند مرورگری

```
/* Create a new XMLHttpRequest object to talk to the Web server */
var xmlhttp = false;
/*@cc_on @*/
/*if (@_jscript_version >= 5)
try {
    xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
    try {
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    } catch (e2) {
        xmlhttp = false;
    }
}
@end @*/

if (!xmlhttp && typeof XMLHttpRequest != 'undefined') {
    xmlhttp = new XMLHttpRequest();
}
```

فعلاً توضیحات و تگ های مرموز شبیه @cc_on را نادیده بگیرید. آنها دستورات خاص کامپایلر جاوا اسکریپت هستند که در مقاله بعدی من، که منحصر به شیء XMLHttpRequest می پردازد، شما با آن بیشتر آشنا خواهید شد. هسته اصلی این کد به سه مرحله تقسیم می شود:

1. ایجاد یک متغیر به نام xmlhttp تا به شیء XMLHttpRequest که شما ایجاد خواهید کرد، ارجاع دهد.
2. تلاش برای ایجاد شیء در مرورگرهای مایکروسافت
 - o تلاش برای ایجاد شیء با استفاده از شیء Msxml2.XMLHTTP
 - o اگر انجام نشد، تلاش برای ایجاد شیء با استفاده از شیء Microsoft.XMLHTTP
3. اگر xmlhttp هنوز ایجاد نشده است، ایجاد شیء به روش غیر مایکروسافتی

در انتهای این پروسه، متغیر xmlhttp باید به یک شیء XMLHttpRequest معتبر، رفرنس دهد، بدون اهمیت به اینکه کاربران شما چه مرورگری را اجرا کرده اند.

یک کلمه درباره امنیت

چه خبر از امنیت؟ مرورگرهای امروزی به کاربران این امکان را می دهند تا با غیر فعال کردن تکنولوژی جاوا اسکریپت و هر تعداد اختیاراتی که در مرورگرشان وجود دارد، امنیتشان را بالا ببرند. در این موارد، کد شما احتمالاً در هیچ موقعیتی کار نخواهد کرد. برای این حالت شما باید مشکل را به دقت حل کنید -- چنین چیزی خودش حداقل یک مقاله است، چیزی که من بعداً درگیر خواهم شد. برای اکنون، شما کدی قوی ولی ناتمام می نویسید که برای انجام Ajax توانمند است. شما بر خواهید گشت و به جزئیات کوچکتر خواهید پرداخت.

درخواست/پاسخ در دنیای آجاکس

خوب، شما اکنون Ajax را می فهمید و یک تصور پایه از شیء XMLHttpRequest و چگونگی ایجاد آن دارید. اگر شما مطالب را به دقت خوانده باشید، حتماً متوجه شده اید که این تکنولوژی جاوا اسکریپت

است که با برنامه وب موجود در سرور ارتباط برقرار می کند، به جای اینکه فرم HTML شما مستقیماً به آن برنامه submit شود.

تکه گمشده چیست؟ چگونه عملاً از XMLHttpRequest استفاده کنیم؟ از آنجاییکه این یک کد مهم است که شما به فرم های مختلف در همه برنامه های Ajax که خواهید نوشت، استفاده می کنید، نگاهی سریع بر این داشته باشید که مدل پایه درخواست/پاسخ در Ajax چگونه است.

ساختن یک درخواست (Request)

شما شیئی جدید و درخشان XMLHttpRequest را دارید؛ حال آنرا برای یک چرخش نگه دارید. ابتدا شما نیاز به یک متد جاوا اسکریپت دارید که صفحه وب شما بتواند فراخوانی کند (همانند زمانیکه کاربر متنی را تایپ می کند یا آیتمی را از یک منو انتخاب می نماید). سپس شما از مراحل پایه زیر در تقریباً همه برنامه های Ajax خود پیروی خواهید کرد:

1. هر داده ای که نیاز دارید، از وب فرمتان دریافت کنید.
2. URLی که می خواهید به آن وصل شوید را بسازید.
3. یک اتصال با سرور برقرار کنید.
4. تابعی را که در سمت سرور باید اجرا شود، مشخص نمایید.
5. درخواست را بفرستید.

لیست 5 یک متد ساده Ajax است که همه اینها را در این روش نشان می دهد:

لیست 5. ایجاد یک درخواست با Ajax

```
function callServer() {
    // Get the city and state from the web form
    var city = document.getElementById("city").value;
    var state = document.getElementById("state").value;
    // Only go on if there are values for both fields
    if ((city == null) || (city == "")) return;
    if ((state == null) || (state == "")) return;

    // Build the URL to connect to
    var url = "/scripts/getZipCode.php?city=" + escape(city) + "&state=" +
    escape(state);

    // Open a connection to the server
    xmlhttp.open("GET", url, true);

    // Setup a function for the server to run when it's done
    xmlhttp.onreadystatechange = updatePage;

    // Send the request
    xmlhttp.send(null);
}
```

بیشتر این کد، واضح و روشن است. خطوط اول کد، از کدهای پایه جاوا اسکریپت به منظور جمع آوری مقادیر تعدادی از فیلدهای فرم استفاده می کند. سپس کد، یک اسکریپت PHP را به عنوان مقصدی که باید به آن متصل شود، مشخص می نماید. توجه داشته باشید که چگونه URL اسکریپت تعریف شده است و سپس شهر و استان (دریافت شده از فرم) با استفاده از پارامترهای ساده متد GET به آن اضافه شده است.

در مرحله بعد، یک اتصال باز شده است. این اولین جایی است که دوباره XMLHttpRequest را در عمل می بینید. متد اتصال تعیین شده است (GET)، همچنین URLی که باید به آن متصل شود. پارامتر آخر، زمانیکه true باشد، یک اتصال غیر همزمان (آسنکرون) را درخواست می دهد (پس این است که Ajax را می سازد). اگر به صورت false استفاده کنید، کد، منتظر سرور می ماند تا درخواست ساخته شود و تا زمانیکه پاسخی داده نشده باشد، ادامه نمی دهد. با ست کردن این پارامتر به true، کاربران شما می توانند همزمان از فرم استفاده نمایند (و حتی می توانند سایر فراخوانی های جاوا اسکریپت را انجام دهند) تا زمانیکه سرور این درخواست را در پشت زمینه پردازش کند.

خاصیت onreadystatechange از xmlhttp به خاطر داشته باشید که نمونه ساخته شده از شیئی XMLHttpRequest است) به شما این امکان را می دهد تا به سرور اطلاع دهید که زمانیکه کار اجرائیش را تمام کرد، چه کار کند (حال این کار 5 دقیقه طول بکشد یا 5 ساعت). از آنجاییکه منتظر سرور نمی ماند،

نیاز است که به سرور اجازه دهید که بداند چه کاری را انجام دهد تا شما بتوانید عکس العمل نشان دهید. در این حالت، هنگامیکه سرور پردازش درخواست شما را تمام کرد، یک متد خاص --به نام updatePage() خواهد شد.

در پایان، متد send() با مقدار null فراخوانی شده است. از آنجاییکه شما داده های ارسالی به سرور را (شهر و استان) به URL درخواستی اضافه کرده اید، نیازی ندارید که چیز دیگری را در درخواست ارسال نمایید. بنابراین این متد درخواست شما را ارسال کرده و سرور می تواند آنچه را که می خواهید، انجام دهد.

اگر چیزی از این مورد متوجه نمی شوید، به سادگی آن توجه نمایید. به جز طبیعت غیر همزمانی Ajax که باید آنرا در ذهن داشته باشید، Ajax خود چیز ساده ای است. شما Ajax را تحسین خواهید کرد که چگونه به شما امکان می دهد تا به جای تمرکز بر کدهای پیچیده درخواست/پاسخ HTTP، بر روی برنامه ها و رابط کاربری خوب تمرکز نمایید.

کد نشان داده شده در لیست 5 به همان سادگی که کار می کند است. داده ها متن ساده هستند و می توانند به عنوان بخشی از URL درخواست اضافه شوند. به جای متد پیچیده POST، متد GET درخواست را ارسال می کند. در این روش هیچ XML یا محتوای Headerی اضافه نمی شود و هیچ داده ای در متن درخواست ارسال نمی گردد --به عبارت دیگر این همان Ajax ایده آل است.

نترسید؛ در ادامه این سری از مقالات، مباحث پیچیده تر می شوند. شما یاد می گیرید که چگونه درخواست های POST را ارسال کنید، چگونه Header درخواست و نوع محتوا (Content Type) را ست کنید، و چگونه XML را در پیغامتان قرار دهید، چگونه امنیت را به درخواستتان اضافه نمایید --این لیست خیلی طولانی است. الان به خاطر سختی مباحث نگران نباشید. بر روی مباحث پایه جلو بروید و به زودی یک زرادخانه کامل از ابزارهای Ajax خواهید ساخت.

کار با پاسخ(Response)

وقت آن رسیده که شما با پاسخ سرور کار کنید. شما در حقیقت دو چیز را نیاز است بدانید:

- هیچ کاری انجام ندهید تا زمانیکه خاصیت xmlHttpRequest.readyState مقدارش برابر 4 شود.
- سرور پاسخ خود را در خاصیت xmlHttpRequest.responseText قرار خواهد داد.

اولی --وضعیت آمادگی-- در مقاله بعدی ادامه داده می شود. شما در آنجا درباره مراحل یک درخواست HTTP بیشتر یاد خواهید گرفت. فعلاً اگر شما وضعیت را برای یک مقدار مشخص چک کنید (4)، همه چیز کار خواهد کرد (و شما باید یکسری موارد را در مقاله مدنظر قرار دهید). مورد دوم که آسان است از خاصیت xmlHttpRequest.responseText به منظور دریافت پاسخ سرور استفاده می کند. لیست 6 یک مثال از یک متدی است که سرور می تواند بر اساس مقادیر ارسالی در لیست 5 فراخوانی نماید.

لیست 6. کار با پاسخ سرور

```
function updatePage() {
    if (xmlHttpRequest.readyState == 4) {
        var response = xmlHttpRequest.responseText;
        document.getElementById("zipCode").value = response;
    }
}
```

دوباره می بینید که این کد خیلی مشکل یا پیچیده نیست. این متد صبر می کند تا سرور با یک وضعیت مناسب آن را فراخوانی کند و سپس مقادیری را که سرور فرستاده است، به منظور ست کردن مقدار یک فیلد از فرم استفاده می نماید (در این مثال کد پستی برای شهر و استان وارد شده توسط کاربر-- اما کاربر هرگز مجبور نیست که دکمه ای را کلیک کند! این همان احساس برنامه های desktop است که قبلاً درباره آن صحبت کرده بودم. پاسخدهی، احساس دینامیک بودن و بسیاری دیگر همه با یک کد کوچک Ajax.

خوانندگان هشیار باید توجه داشته باشند که فیلد ZipCode یک فیلد Text نرمال است. زمانیکه سرور کدپستی را برگشت داد و متد updatePage() مقدار آن فیلد را با کدپستی شهر/استان ست کرد، کاربران می توانند مقدار را بازنویسی کنند. این کار به دو دلیل عمدی است: برای اینکه همه چیز در این مثال ساده تر باشد و به شما نشان دهیم که بعضی اوقات شما می خواهید که کاربران قادر باشند که

آنچه را که سرور می گوید بازنویسی نمایند. هر دو اینها را در ذهن داشته باشید؛ این موارد در یک طراحی خوب UI مهم است.

گرفتاری در وب فرم

خوب چه چیزی باقیمانده است؟ عملاً هیچ چیز. شما یک متد جاوا اسکریپت دارید که اطلاعات وارد شده در فرم توسط کاربر را جمع آوری کرده، آنرا به سرور می فرستد، متد جاوا اسکریپت دیگری را به منظور گوش دادن و کار کردن با پاسخ، ارائه می دهد. و حتی هنگامی که پاسخ رسید، مقدار یک فیلد را ست می کند. آنچه که واقعاً باقیمانده، فراخوانی آن متد جاوا اسکریپت اولی و شروع کل پروسه است. بدیهی است که شما می توانید یک دکمه را فرم HTML تان اضافه کنید، اما آن زیبایی سال 2001 است، اینطور فکر نمی کنید؟ از مزایای تکنولوژی جاوا اسکریپت شبیه لیست 7 استفاده کنید.

لیست 7. شروع یک پروسه Ajax

```
<form>
<p>City: <input type="text" name="city" id="city" size="25"
onchange="callServer();" /></p>
<p>State: <input type="text" name="state" id="state" size="25"
onchange="callServer();" /></p>
<p>Zip Code: <input type="text" name="zipCode" id="zipCode" size="5" /></p>
</form>
```

اگر احساس می کنید که این کد شبیه کدهای معمولی است، حق با شماست. زمانیکه یک کاربر مقداری جدید برای هم فیلد شهر و هم استان وارد می کند، متد `callServer()` فراخوانی شده و بازی Ajax شروع می شود. احساس می کنید که بر روی همه چیز کنترل دارید؟ خوب است؛ ایده همین است!

در خاتمه

در این مرحله احتمالاً آماده نیستید که بروید و اولین برنامه Ajax خودتان را بنویسید -- حداقل تا وقتی که کمی در بخش منابع عمیق نشده اید. اما شما می توانید ایده اصلی درباره اینکه چگونه این برنامه ها کار می کنند و یک شناخت پایه از شیئی XMLHttpRequest را شروع کنید. در مقالات بعدی شما یاد می گیرید که در این شیئی استاد شوید، چگونه ارتباطات جاوا اسکریپت به سرور را انجام دهید، چگونه با فرم های HTML کار کنید، و حتی کنترلی بر روی DOM داشته باشید.

الآن کمی وقت بگذارید و فقط درباره اینکه برنامه های Ajax چقدر قوی هستند، فکر کنید. فرم ویی را تصور کنید که نه تنها موقع کلیک دکمه به شما پاسخ می دهد، بلکه هنگامیکه شما در یک فیلد تایپ می کنید، وقتی که یک آیتم از Combo box انتخاب می کنید... حتی زمانیکه موس را در صفحه Drag می کنید، نیز پاسخگو باشد. درباره اینکه غیر همزمانی (آسنکرون) دقیقاً چه معنایی دارد؛ درباره اینکه کد جاوا اسکریپت در حال اجراست و منتظر پاسخ سرور به درخواستش نمی ماند، فکر کنید. با چه مشکلاتی درگیر می شوید؟ چه زمینه هایی است که شما باید مراقب باشید؟ و چگونه طراحی فرم های شما به منظور توجیه این روش جدید در برنامه نویسی، تغییر خواهد کرد؟

اگر شما بر روی این موارد وقت بگذارید، خیلی بهتر از اینست که مقداری داشته باشید که بتوانید در برنامه ای که واقعاً نمی دانید چه می کند، کپی و Paste نمایید. در مقاله بعدی، این ایده ها را در عمل به کار می برید و جزئیاتی را درباره کدی که نیاز دارید تا برنامه ای این چنینی کار کند، به شما خواهم گفت. بنابراین تا آن موقع از امکانات Ajax لذت ببرید.

منابع آموزش

- is one of the companies on the leading edge of user interface design; you [Adaptive Path](#) can learn a ton about Ajax by perusing their pages.

- [Jesse James Garrett](#) If you're curious about where the term Ajax came from, check out [this one](#) and his excellent articles (like [this one](#)) on Ajax.
- You can get a head start on the next article in this series, focusing on the XMLHttpRequest object, by checking out this [excellent article on the XMLHttpRequest object](#).
- [Microsoft Developer Network's](#) If you use Internet Explorer, you can get the scoop at the [XML Developer Center](#).
- (developerWorks, September 2005) introduces a groundbreaking approach to creating dynamic Web application experiences that solve the page-reload dilemma. [Ajax for Java developers: Build dynamic Java applications](#)
- (developerWorks, October 2005) shows you five ways to serialize data in Ajax applications. [Ajax for Java developers: Java object serialization for Ajax](#)
- (developerWorks, October 2005) is a tutorial for those interested in developing rich Web applications that dynamically update content using Ajax and PHP. [Using Ajax with PHP and Sajax](#)
- (developerWorks, October 2005) shows how to implement a Web browser-based SOAP Web services client using the Ajax design pattern. [Call SOAP Web services with AJAX, Part 1: Build the Web services client](#)
- (developerWorks, May 2005) details the Document Object Model as a method to build dynamic Web applications. [XML Matters: Beyond the DOM](#)
- (developerWorks, November 2005) demonstrates how to construct real-time-validation-enabled Web applications with AJAX. [Build apps with Asynchronous JavaScript with XML, or AJAX](#)
- (developerWorks, November 2005) demonstrates how to automate the heavy-lifting of AJAX. [Ajax for Java developers: Ajax with Direct Web Remoting](#)
- [surveys AJAX/JavaScript libraries](#) The OSA Foundation has a wiki that details XMLHttpRequest (not to mention all kinds of other XML objects, as well as DOM, CSS, HTML, Web Service, and Windows and Navigation objects. [object reference section](#)
- detailing some of the basic Ajax principles. [strategy white paper](#) You can read a nice [Flickr.com](#) See one of the outstanding Ajax applications online at [Flickr.com](#)
- , from Google, is another great example of Ajax revolutionizing Web applications. [GMail](#)
- (O'Reilly Media, Inc., February 2006) takes the ideas outlined in this [Head Rush Ajax](#) article and series and brings them (and a lot more) to you in the innovative and award-winning Head First format. [Head Rush Ajax](#)
- , 4th Edition (O'Reilly Media, Inc., November 2001) is a [JavaScript: The Definitive Guide](#) great resource for the JavaScript language and working with dynamic Web pages. [JavaScript: The Definitive Guide](#)
- specializes in articles covering various Web-[Web Architecture zone](#) The developerWorks based solutions. [Web Architecture zone](#)

مفتحو



- [.Participate in the discussion forum](#)
- is a great blog for all things Ajax. [Ajax.NET Professional](#)
- [. developerWorks blogs](#)Get involved in the developerWorks community by participating in

تاثیر تکنولوژی جدید اجکس بر تجارت اینترنت

مقدمه

تجربه شان در دنیای وب و کار با وب سایت های مختلف، چک کاربران اینترنت افرادی هستند که نهایت گروه های اینترنتی است. اما امروزه آنها در دنیای اینترنت به دنبال کالاهای کردن ایمیل ها و یا عضویت در خرید کالایی موردنظرشان می باشند. آنها در وب سایت ها گشته و در یک فروشگاه اینترنتی تصمیم به اعتماد نموده و اغلب پس از خاص می گیرند. این کاربران به سختی به وب سایت های خرید و فروش آنلاین می شوند. بازگشت مجدد یک کاربر به وب آغاز پروسه خرید، آن را در نیمه رها کرده و از سایت خارج سایتی که تجربه ناخوشایند یک خرید اینترنتی را داشته است، بسیار بعید می نماید. در این مقاله، سعی در ادامه روش های نوین رفع این مشکلات بیان گردد شده است دلایل و

علت رها شدن پروسه خرید، پرداخت و ارسال سفارش چیست؟ در اکثر موارد کاربران از زمان طولانی که باید صرف پرکردن یک فرم و submit کردن آن کنند، گله داشته و به بیانی ساده تر، حوصله شان سر می رود. این امر در ایران بسیار ملموس تر می باشد زیرا سرعت اینترنت در کشور ما - بدون اغراق در اکثر شهرها - پایین می باشد و یا در نیمه راه، به طور کامل اتصال به اینترنت قطع می شود. فرض کنید در مرحله حساس ورود اطلاعات کارت اعباری خود در مدخل پرداخت یک بانک می باشید و پس از زدن دکمه "تایید" ناگهان با این پیغام روبرو می شوید:

The page cannot be displayed.

مطمئناً احساس بدی به شما دست داده و بر روی بازگشت در مرورگر خود کلیک می کنید ولی مسلماً کار بدتر می شود!

این فقط یک نمونه است و ممکن است به دلایل خاصی رخ دهد اما با فرض اینکه شما با موفقیت اطلاعات کارت اعباری خود را وارد کنید، باید مدتی را منتظر بمانید تا صفحه refresh شده و نتایج را مشاهده نمایید.

اجکس تکنولوژی جدیدی است که مخفف کلمه Asynchronous JavaScript And XML می باشد. در مورد این تکنولوژی مقالات متعددی وجود دارد و معرفی آن در این مقوله نمی گنجد. من تنها تاثیر این فن آوری جدید را بر افزایش فروش آنلاین یک سایت تجارت اینترنتی توضیح می دهم.

نکته: استفاده از تکنولوژی های جدید ممکن است تعداد کاربران وب سایت ما را کاهش دهد.

1. آیا در تمام وبسایت ها و platform ها قابل اجراست؟
2. آیا این قابلیت بر تجربیات کاربر خواهد افزود و وی راحت تر می تواند از سایت استفاده کند؟
3. آیا این قابلیت کاربران را گیج خواهد نمود؟

و موارد دیگری از این دست .

در استفاده از تکنولوژی اجکس می توان برای کاربران مطالبی همراه با تصاویر تحت عنوان "راهنما" - "آموزش کار با تکنولوژی جدید" و غیره فراهم آورد تا کاربران بی تجربه ما گیج نشوند.

به عنوان مثال ، با استفاده از تکنولوژی اجکس کاربر می تواند نتایج جستجوی خود را در بلادرنگ مشاهده کرده و منتظر refresh شدن صفحه ننماید. اجکس کار با وب سایت ها را تسهیل کرده و سرعت را افزایش داده است. می توان در این مورد - استفاده از اجکس در یک وب سایت خرید و فروش اینترنتی - به موارد زیر اشاره کرد:

1. مشاهده بلادرنگ نتایج جستجو
2. drag and drop محصولات به سبد خرید - کاربر بسیار سریع تر می تواند سبد خرید خود را مدیریت کند.
3. زیرمجموعه های منوهای سایت بدون refresh شدن صفحه باز شده و اطلاعات لازم را به کاربر می دهند.
4. اگر مدیر سایت برای بخش پشتیبانی از برنامه های Live Chat استفاده می کند، سریع تر مشتری را به پاسخ سوالات رهنمون خواهد شد.

این موارد نمونه ای از کمک تکنولوژی اجکس به تجارت اینترنتی می باشد. شما می توانید با توسعه سایت خود و استفاده از این فن آوری به راحتی کاربران و وبگردهایی که به سایت شما می آیند به مشتری تبدیل کرده و فروش خود را بالا ببرید. مطمئناً کاربری که مایل به خرید اینترنتی است، در چنین سایتی تجربه خوبی از ecommerce خواهد داشت و هرگز رقباى شما را ترجیح نخواهد داد.

AjaxTagLib یک farmework ساده و موثر برای توسعه دهندگان J2EE

Ajax Tag Library مجموعه ای از تگ های JSP است که استفاده از روش Ajax را در صفحات JSP آسان کرده است. از آنجایی که روش Ajax یک روش سمت کلاینت و تمرکز آن بر روی متدهای JavaScript است و اینکه اکثر برنامه نویسان سمت سرور معمولاً دانش کمی درباره برنامه نویسی سمت کلاینت دارند. این فریم ورک به J2EE developer ها کمک میکند تا با استفاده از این مجموعه تگ ها، به راحتی و بدون نوشتن هیچ کدی در سمت کلاینت، به نتیجه مطلوب خود برسند.

برای استفاده از این مجموعه تگ ها احتیاج به JDK 1.4+ و یک Servlet Container نظیر tomcat دارید . مرورگرهای Internet Explorer 5.0+ و Firefox 1.0+ از این framework پشتیبانی می کنند .

- [مشاهده مثالها به صورت آن لاین](#)
- [دانلود مثالها از SourceForge](#)
- [دانلود Framework از SourceForge](#)
- [دانلود Demo از SourceForge](#)

این مجموعه کتابخانه ای شامل تگ های زیر می باشد:

Autocomplete

نمایش درمی آورد که با توجه به متنی که درون آن نوشته می شود، را به TextBox این تگ پر کاربرد یک را که با متن ورودی مشابه باشد به عنوان موارد پیشنهادی ارائه می دهد تا مجموعه ای از مقادیر Tag این بنویسد، به طور دقیق به خاطر نداشته باشد، TextBox کاربر آیتمی را که باید در این درصورتی که به او کمک می کند تا از مجموعه پیشنهادی، آیتم مورد نظر خود را انتخاب نماید.

Enter Car Model

Available values start with letters: 'A', 'C', 'E', 'F', 'M', 'R', 'T'

Name:

E

Escape

Expedition

Explorer

Element

Callout

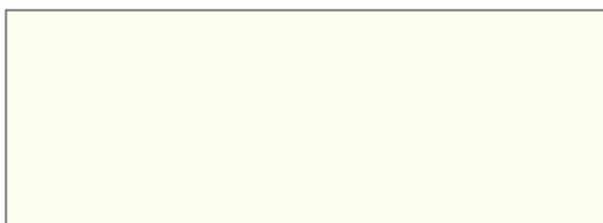
این تگ یک کاربرد ساده ولی موثر دارد. به این صورت که با وصل شدن به هر تگ HTML باعث می شود که با کلیک کردن روی تگ HTML مورد نظر یک بالون popup با متن دلخواه نمایش داده شود.

HTML Content Replace

این Tag محتوای یک Content Area همانند تگ DIV یا غیره را با کلیک کردن یک link یا image یا هر تگ html دیگری تغییر میدهد.

قبل از کلیک

!Error



بعد از کلیک

!Error

FORD
Models

- Escape
- Expedition
- Explorer
- Focus
- Mustang
- Thunderbird

Last Updated: Wed Sep 14 11:03:33 EDT 2005

HtmlContent in Action

Select by ANCHOR link.

- [Ford](#)
- [Honda](#)
- [Mazda](#)

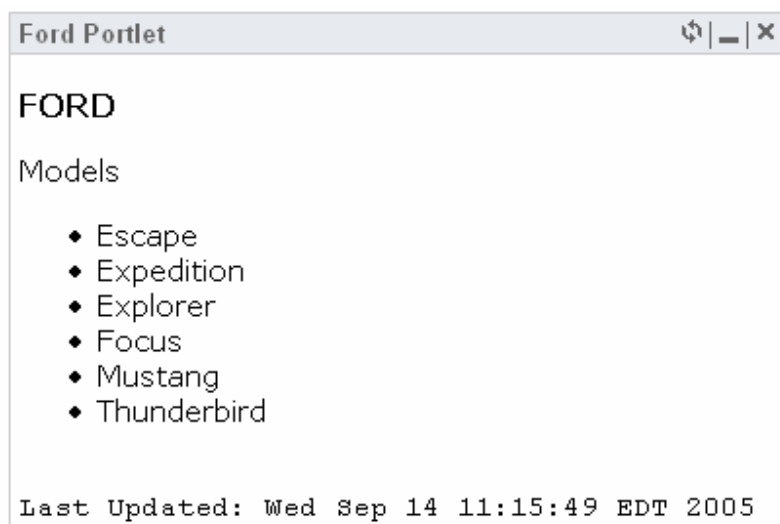
HtmlContent in Action

Select by ANCHOR link.

- [Ford](#)
- [Honda](#)
- [Mazda](#)

Portlet

خب این هم یک Tag برای Portal developer ها. این Tag یک پرتلت آجاکسی را به نمایش در می آورد که خودش هر چند ثانیه به طور خودکار refresh می شود.



Select/Dropdown

با استفاده از این تگ قادرید فیلد های یک dropdown را با توجه به انتخاب فیلد های dropdown دیگر تغییر بدهید.

قبل از انتخاب



Select/Dropdown

با استفاده از این تگ قادرید فیلد های یک dropdown را با توجه به انتخاب فیلد های dropdown دیگر تغییر بدهید.

قبل از انتخاب


Choose Your Car

Make:

Model:

بعد از انتخاب

Choose Your Car



Make:

Model:

Tab Panel

این هم تگی برای نمایش Tab Panel های آجاکسی

!Error

Ford

Honda

Mazda

FORD
Models

- ◆ Escape
- ◆ Expedition
- ◆ Explorer
- ◆ Focus
- ◆ Mustang
- ◆ Thunderbird

Last Updated: Wed Sep 14 11:16:45 EDT 2005

Toggle

این Tag برای نمایش و تغییر حالت True و False بکار برده می شود

قبل از کلیک

☆ Toggle Me

Response

بعد از کلیک

★ Toggle Me

Response

This is a **test** of the emergency broadcast system.

Update Field

این Tag مقادیر یک یا چند فیلد را با توجه به مقدار وارد شده در یه فیلد دیگر تغییر می دهد.

قبل از وارد کردن مقدار

Velocity Conversion

Enter miles per hour and click Calculate

Miles/Hour (mph)

Calculate

Kilometers/Hour (kph)

Meters/Second (m/s)

بعد از وارد کردن مقدار و کلیک کردن دکمه Calculate

Velocity Conversion

Enter miles per hour and click Calculate

Miles/Hour (mph)

Kilometers/Hour (kph)

Meters/Second (m/s)

راهنمای استفاده از AjaxTagLib

1. برای شروع ajaxtags.jar را در مسیر WEB_INF/lib کپی نمایید
2. فایل تعریف taglib یا همان ajaxtags.tld را به صورت زیر در web.xml معرفی کنید. این کار برای کاربرهای JSP 2.0 الزامی نیست.

```
<taglib>
<uri>http://ajaxtags.org/tags/ajax</uri>
<location>/WEB_INF/ajaxtags.tld</location>
</taglib>
```

1. فایل های JavaScript موردنیاز که همراه با این framework دانلود می شود را به web application خود اضافه کنید.

- **Prototype 1.4.0** : Prototype 1.4.0.js
- **Scriptaculous 1.5.1**: scriptaculous.js, builder.js, controls.js, dragdrop.js, effects.js, slider.js
- **OverLIBMWS** (Optional, for ajax: callout only): overlibmws.js

این مجموعه از جاوا اسکریپت ها از موارد مطرح در زمینه استفاده از تکنیک Ajax است که توسط شرکت های مختلف توسعه یافته و در فریم ورک های مختلفی از آنها استفاده شده است.

توصیه من به Ajax developer ها اینست که حتما نگاهی بر source این فایل ها بیندازند. مطمئنا می تواند مفید و الهام بخش باشد.

خوب حال به سمت سرور می پردازیم

برای ایجاد Response به فرمت های HTML, XML و text ، کلاس های کمکی یا همان Helper Class هایی در این framework در نظر گرفته شده است.

فایل های helper با نام های AjaxHtmlHelper و AjaxXmlBuilder در این package واقع شده اند .

org.ajaxtags.helpers

نمونه استفاده از AjaxXmlBuilder

```
// Get maker from your service bean
CarService service = new CarService();
List list = service.getModel sByMake(make);

return new AjaxXml Builder(). addItems(list, "model ", make"). toString();
```

خروجی تولید شده به صورت زیر می باشد.

```
<?xml version="1.0" encoding="UTF-8"?>
<ajax-response>
  <response>
    <item>
      <name>Expedition</name>
      <value>Ford</value>
    </item>
    <item>
      <name>Focus</name>
      <value>Ford</value>
    </item>
    ...
  </response>
</ajax-response>
```

نمونه ای از Response به فرمت XML

```
<?xml version="1.0" encoding="UTF-8"?>
<ajax-response>
  <response>
    <item>
      <name>Record 1</name>
      <value>1</value>
      <item>
        <name>Record 2</name>
        <value>2</value>
      </item>
    </item>
    <item>
      <name>Record 3</name>
      <value>3</value>
    </item>
  </response>
</ajax-response>

</item>
```

نمونه ای از Response به فرمت text

Record 1,1
Record 2,2
Record 3,3

آماده کردن فایل JSP

1. معرفی کردن taglib در صفحه
2. اضافه کردن فایل های جاوا اسکریپت مورد نیاز به صفحه
3. اضافه کردن فایل های style مورد نیاز به صفحه
4. نوشتن کد های مورد نیاز صفحه
5. اضافه کردن آجاکس تگ های مورد نیاز به صفحه

یک صفحه نمونه JSP

```
<%@ taglib uri="http://ajaxtags.org/tags/ajax" prefix="ajax" %>
<html>
```

```

<head>
<title>AJAX JSP Tag Library</title>

<script type="text/javascript" xsrc="prototype.1.4.0.js"></script>
<script type="text/javascript" xsrc="scriptaculous.js"></script>
<script type="text/javascript" xsrc="overlibmws.js"></script>
<script type="text/javascript" xsrc="ajaxtags-1.2.js"></script>

<link type="text/css" rel="stylesheet" xhref="ajax.css" />
</head>
<body>

<form>
Make :
<select id="make" name="make">
<option value="">Select make</option>

<c:forEach items="${makes}" var="make">

<option value="${make}">${make}</option>
</c:forEach>
</select>

Model :
<select id="model" name="model">
<option value="">Select model</option>
</select> </form>

<ajax:select
baseUrl="${pageContext.request.contextPath}/GetCarModel.view"
source="make"
target="model"
parameters="make={make}" />

</body>
</html>

```

خوب، حالا می‌رسیم به فایل اصلی برنامه که وظیفه رسیدگی و پاسخ دادن به Request ها را بر عهده دارد. بدین منظور کاربران باید از یک کلاس انتزاعی (abstract) که در این فریم ورک گنجانده شده به نام **BaseAjaxServlet** و کاربران Strust از **BaseAjaxAction** ارث بری کرده و متد **getXmlContent** (HttpServletRequest, HttpServletResponse) آن را پیاده سازی کنند.

برای کسانی که به طور کامل با جاوا اسکریپت قطع رابطه نکردند

یکی از جدیدترین امکاناتی که به این framework اضافه شده است، جدا کردن متدهای پردازش پاسخ یا Response Parser می باشد؛ به این مفهوم که کاربر را قادر می سازد تا در صورت نیاز با توجه به فرمت خروجی یا همان فرمت Response، نوع parser خود را تغییر دهد. در این فریم ورک، چند parser پیش فرض در نظر گرفته شده است که در صورت نیاز توسط کاربر قابل جایگزینی می باشد.

پیدا خواهید کرد. framework آشنایی بهتر و دقیق تری با این AjaxTagLib مطمئناً با مراجعه به مستندات این مجموعه، ساختار به نسبت ساده آن می باشد که برای کاربران امکان آ یکی از مهمترین خصوصیات توسعه آن را فراهم ساخته است.

این روزا اگه بخوای تو مباحث **web application** کلاس بذاری باید کلمه **Ajax** رو هم ورد زیونمون کنیم تا اطرافیانمون حداقل بدونن که یه چیزائی حالیمونه.

هدف از بیان این مطلب معرفی **AjaxAc** بود که این روزا مٹ خیلی تکنولوژی های دیگه طرافداران زیادی رو جذب خودش کرده.

حالا این **AjaxAc** چي هست؟

AjaxAc يك فریم ورك باز متن هستش که با **php** نوشته شده و برای ایجاد و توسعه **Ajax Applications** بکار می رود.

حتما در ياهو دیده اید که با تایپ حروف اول نام دوستانتان نام چند نفر از آنها از دیتابیس لود شده و به شما نمایش داده می شود. اینکار بدون اینکه صفحه بارگذاری مجدد شود انجام می شود. به زبان ساده تر آژاکس تکنولوژی ای است که در آن با لود شدن یک صفحه، امکان فراخوانی برنامه های سمت سرور بدون بارگذاری و لود مجدد صفحه وجود دارد. برنامه های آژاکس راهی برای ساخت برنامه های اینتراکتیو تر در وب فراهم کرده اند. برای اینکار کافیسیت یک عنصر فعال که قابل فراخوانی در جاوااسکریپت باشد بتواند تحت کنترل جاوااسکریپت با سرور تماس گرفته و اطلاعات درخواست کند. بعنوان مثال این عنصر فعال می تواند یک برنامه **PHP** یا **ASP** یا ... را فراخوانی کرده و نتایج آن را از طریق جاوااسکریپت در صفحه جاری تاثیر دهد. عنصر فعال قبل از معرفی آژاکس می توانست یک **ActiveX** یا جاوا اپلت ساده باشد اما اکنون این عنصر آبجکت **XMLHttpRequest** است که در مجموعه ابزارهای **XML** میکروسافت قرار دارد.

این فریم ورك می تونید از اینجا [دانلود](#) نمائید
برای کسب اطلاعات بیشتر سری به [اینجا](#) بزنید

منبع : <http://iranphp.net/>

اشاره :

اگر برنامه نویس وب هستید، لابد خبر دارید که این روزها همه درباره **Ajax** صحبت می‌کنند. بعضی معتقدند Ajax یکی از ارکان فنی موج جدیدی است که به Web 2.0 معروف شده است و انجمن‌ها و تربون‌های آنلاین مملو از مقالات و اظهارنظرهای مربوط به Ajax است

اگر برنامه نویس وب هستید، لابد خبر دارید که این روزها همه درباره **Ajax** صحبت می‌کنند. بعضی معتقدند Ajax یکی از ارکان فنی موج جدیدی است که به Web 2.0 معروف شده است. بعضی از سایت‌های بزرگ مانند یاهو و گوگل هم‌اکنون Ajax را به خدمت گرفته‌اند، انجمن‌ها و تربون‌های آنلاین مملو از مقالات و اظهارنظرهای مربوط به Ajax است و همزمان تلاش‌های وسیعی در جریان است تا این فناوری جدید همچنان به صورت این سورس باقی بماند و انحصاری نشود.

Ajax چیست؟

واژه **Ajax** با تلفظ <ای جکس> یا <ای ژاکس> سرنام عبارت Asynchronous JavaScript and XML و به معنی <ترکیب نامتقارن جاوا اسکریپت و XML> است. ماهیت صفحات وب و پروتکل HTTP به گونه‌ای است که به طور معمول وقتی در حال وب‌گردی هستیم، به ازای هر کنش و واکنش میان ما و سایتی که در حال کار با آن هستیم، کل یک صفحه وب از نو بارگذاری و تازه‌سازی (refresh) می‌شود.

ای‌جکس فناوری جدیدی است که تغییر محسوس را در این سناریو به وجود می‌آورد؛ به این ترتیب که به جای بارگذاری مجدد کل صفحه، فقط قسمتی تغییر می‌کند که قرار است اطلاعات جدید را به نمایش درآورد و کلیه عملیات ارسال اطلاعات و دریافت نتایج در پشت صحنه انجام می‌شود. در نتیجه هیچ‌گاه صفحه سفید و خالی وب در فواصل کنش و واکنش‌های هنگام کار با مرورگر دیده نمی‌شود و احساسی مشابه تجربه کار با یک نرم‌افزار دسکتاپ به کاربر دست می‌دهد.

جادوی ای‌جکس چیزی نیست جز یک فکر بکر و آن هم ترکیب کردن جاوا اسکریپت و XML در قالب یک موجود نرم‌افزاری جدید.

اصطلاح Ajax را اولین بار کارشناسی از شرکت Adaptive Path به نام جسی جیمزگرت در مقاله‌ای با عنوان <ای‌جکس؛ رهیافت جدیدی در برنامه‌های تحت وب> مطرح کرد و خیلی سریع مورد استقبال گسترده برنامه نویسان وب در سراسر جهان قرار گرفت. اعتقاد عمومی این است که تاریخچه به‌کارگیری تکنیک مذکور به پیدایش نرم‌افزار Outlook WebAccess و استفاده از فرمان XMLHttpRequest که مایکروسافت ابداع کرده و در نرم‌افزار اینترنت اکسپلور به‌کار رفته است، برمی‌گردد. اما امروزه اغلب مرورگرهای مهم و شناخته‌شده (از جمله فایرفاکس) از آن پشتیبانی می‌کنند و دیگر یک فناوری محدود به اینترنت اکسپلورر نیست.

چرا Ajax این قدر اهمیت دارد؟

فناوری ای‌جکس جدید است و شگفت‌انگیز به نظر می‌رسد. ولی در حقیقت کل این شعبده بر اساس فناوری‌هایی بنا شده است که هم‌اکنون موجودند: جاوا اسکریپت و XML هر دوی این فناوری‌ها تا حد زیادی باز هستند و منحصر به شرکت خاصی نیستند. به همین دلیل این روزها تمام محافل دنیای برنامه نویسی مملو از مقالات و تحلیل‌هایی درباره **Ajax** است. به عنوان نمونه می‌توانید صفحه نخست سایت موسوم به <کانال شبکه برنامه نویسان سان> را باز کنید.

بیشتر این صفحه (در زمان نگارش این یادداشت) به مقالات و مطالب متنوعی درباره **Ajax** اختصاص یافته است. مجلات برنامه‌نویسی نیز مقالات متعددی در این زمینه منتشر کرده‌اند و سایت‌های معروفی مانند O'Reilly و xml.com نیز در این باره مطلب دارند.

بنابراین ظرفیت بالایی برای تبدیل ای‌جکس به یک استاندارد جهانی وجود دارد. از این رو قرار است در تاریخ سیزدهم مارس سمینار مهمی در زمینه **Ajax** برگزار شود. اگر همین الان به سایت ajaxseminar.com مراجعه کنید، متوجه می‌شوید که علاوه بر طراح این تکنیک، چندین برنامه‌نویس شاخص از شرکت‌های بزرگی همچون یاهو در آن شرکت خواهند داشت.

این فناوری از یک جنبه دیگر نیز اهمیت دارد. به دلیل عدم نیاز به بارگذاری مجدد کل یک صفحه وب، مقدار داده‌هایی که لازم است برای تکمیل یک Interaction میان کاربر و سایت مبادله شود، به شدت کاهش می‌یابد و این به معنی افزایش محسوس سرعت نرم‌افزارهای تحت وب، سهولت به‌کارگیری اینترنتی‌های

میتنی بر ای جکس و کاربر پسند تر شدن آن ها می شود. به همین دلیل این روزها اکثر پورتال های بزرگ (مانند یاهو) مشغول طراحی اینترفیس های میتنی بر Ajax هستند .

با این همه مزایای ای جکس به همین موارد ختم نمی شود. تصور کنید وقتی در طول شبانه روز میلیارد ها بار فرایند بارگذاری مجدد صفحات وب تعاملی در شبکه اینترنت تکرار می شود، استفاده از ای جکس چگونه موجب آزاد شدن پهنای باند اینترنت و در نتیجه میلیاردها دلار صرفه جویی اقتصادی در این زمینه می شود و این به نوبه خود انقلابی در عرصه وب و فضای سایبر به شمار می رود. این تحول در شرایطی اتفاق می افتد که همزمان فناوری های اینترنت پرسرعت همچون ADSL به شدت در حال رشد هستند .

حال تصور کنید اگر فناوری ای جکس در مقیاس گسترده ای مورد استفاده نرم افزارهای تحت وب قرار گیرد. آنگاه افزایش سرعت کار با اینترنت چند برابر خواهد شد. چنین تحولی می تواند اینترنت را به کامپیوتر دوم کاربران تبدیل کند. به گونه ای که برای آن ها اجرای يك نرم افزار از روی وب تفاوت محسوسی با اجرای آن از روی کامپیوتر دسکتاپ نداشته باشد. در این صورت ممکن است واقعا بخش مهمی از توان پردازشی نرم افزارها، چه از نوع دسکتاپ و چه از نوع تحت وب به سیستم های موسوم به Web Service واگذار شود. از این رو کسانی که مبتکر و مشوق ایده وب سرویس بوده اند، این روزها انگیزه تازه ای برای سرعت بخشیدن به این فناوری پیدا کرده اند و تلاش می کنند دو فناوری ای جکس و وب سرویس را به یکدیگر پیوند دهند و راهکارهای تازه ای بیافرینند.

کدام سایت ها از ای جکس استفاده کرده اند؟

دنیای برنامه نویسی وب هنوز درگیر هیجانات مربوط به آشنایی با این پدیده است و در حال بررسی مشکلات تکنیکی و هضم و جذب آن در بافت برنامه های تحت وب می باشد. به همین دلیل میزان استفاده عملی از آن چندان گسترده نیست. اما چون کارایی ای جکس دیگر برای همه ثابت شده است، حرکت های بزرگی در سراسر اینترنت به سمت استفاده از این فناوری در جریان است. از جمله اینترفیس نسخه دوم Yahoo Mail که به نسخه بتا معروف است و فعلا در دسترس کاربران سرویس غیر رایگان یاهو قرار دارد، از همین فناوری استفاده می کند که به زودی اینترفیس جدید در دسترس عموم قرار می گیرد. سرویس Gmail و Google Map نیز از این فناوری استفاده می کنند .

بهروز نوعی پور
ماهانامه شبکه - بهمن ۱۳۸۴

کتابخانه ASP.NET Atlas

مقدمه

ASP.NET Atlas کتابخانه ای پربار، هم در سمت سرور و هم در سمت Client برای پیاده سازی برنامه های بر مبنای AJAX است. این مقاله یک دید عمومی نسبت به امکانات اطلس (Atlas) به شما خواهد داد. چون اطلس یک کتابخانه بسیار گسترده است این مقاله دو امکان بسیار مهم اطلس را معرفی میکند:

1. امکان صدا کردن یک web service از طریق یک اسکریپت در سمت Client
2. سهولت استفاده از JavaScriptهایی که در تمام مرورگرهای وب اجرا میشوند.

پیش زمینه

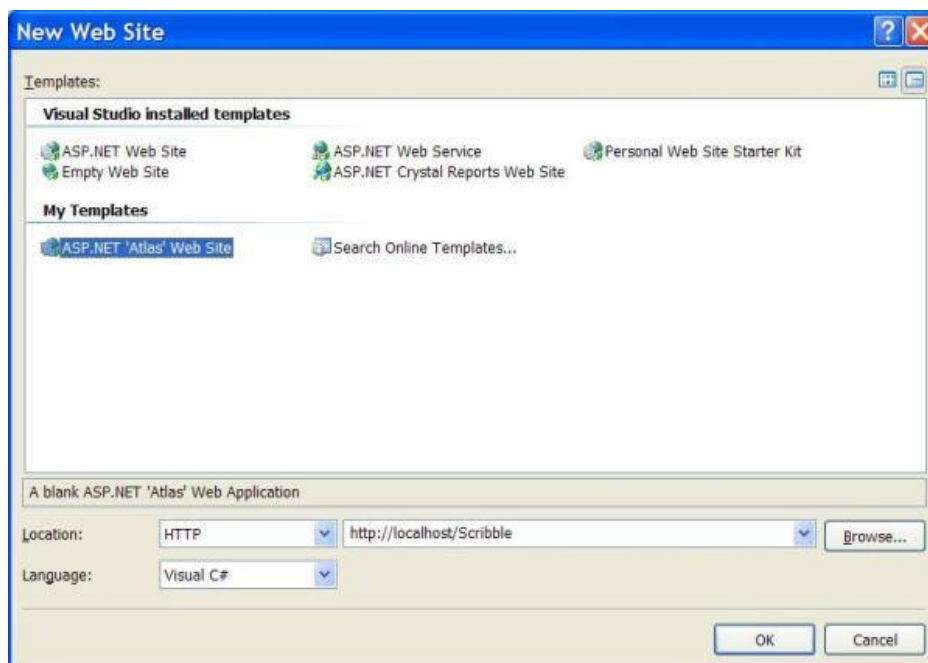
نقاشی از طریق وب همواره برای کاربران، از جمله من جالب بوده است. این برنامه به کاربر اجازه میدهد که با کمک موس روی صفحه نقاشی کند البته در مرورگرهای وب. نمونه این برنامه در سایت [JavaScript Draw](#) موجود است ولی تنها بروی Mozilla Firefox قابل اجرا است. ولی کد ذیل با تکنولوژی Atlas محدودیتی روی هیچ مرورگری ندارد.

نصب

هم اکنون (زمان نوشتن این مقاله) شما میتوانید نسخه ماه دسامبر Atlas را از [اینجا](#) بردارید. اگر این لینک وجود نداشت، میتوانید از وب سایت Atlas لینک درست را بردارید. کتابخانه Atlas فایل با پسوند .vsi (تمپلت Visual Studio 2005) موجود است. کافی است که روی فایل در دستگاه خود (که Visual Studio 2005 را روی آن نصب کرده اید) دبل کلیک کنید تا براحتی نصب شود.

ساخت یک پروژه اطلس

به محض اینکه اطلس را نصب کردید میتوانید یک پروژه اطلس را با رفتن به منوی File > New > Web Site . بعد از اینکار صفحه زیر پدیدار میشود:



در قسمت Location دو حالت File System یا HTTP قابل انتخاب است. با انتخاب HTTP، میتوانید وب سایت خود را روی IIS Sever بسازید. و با انتخاب File System وب سایت شما روی فایل سیستم محلی ساخته میشود (بدون نیاز به IIS). شما هر کدام از انتخابها را میتوانید استفاده کنید. ولی به نظر میرسد که برنامه با IE روی IIS بهتر کار میکند.

ساختار پروژه اطلس

وب سایت اطلس تازه ساخته شده، چنین ساختاری دارد:

- App_Data
یک فولدر خالی برای گذاشتن فایل‌های داده
- Bin
فولدری که dllهای اطلس در آن است. در واقع قسمت سروری کتابخانه اطلس در این فولدر است.
- ScriptLibrary
محلی که میتوانید فایل‌های JavaScript خود را در آن بگذارید.
- Atlas
 - اسکریپت‌های سمت کلاینت (Client) در این فولدر و در دو زیردایرکتوری قرار میگیرند.
 - § Debug
نسخه قابل دیباگ از اسکریپت‌های JavaScript در این فولدر قرار میگیرند
 - § Release
نسخه قابل عرضه در اینجا قرار میگیرند. این اسکریپت‌ها خیلی فشرده تر نوشته شده اند و بعضی از کدهای مربوط به دیباگ حذف شده اند.

اسکریپت‌های کلاینت اطلس

نسخه ماه دسامبر اطلس شامل اسکریپت‌های زیر است:

- Atlas.js
هسته اصلی اسکریپت‌های اطلس که شامل فانکشن‌های کاربردی اولیه و کنترل‌ها و کامپوننت‌های سمت کلاینت است.
- AtlasCompaat.js و AtlasCompat2.js
لایه سازگار کننده انواع مرورگرهای وب با اطلس در این دو فایل هستند.
- AtlasRuntime.js
نسخه خلاصه Atlas.js که شامل کنترل‌ها و کامپوننت‌های سمت کلاینت نیست.
- AtlasUIDragDrop.js
ابزارهایی برای پیاده کردن Drag & Drop در صفحات وب درون این فایل است.
- AtlasUIGlits.js
ابزارهایی برای پیاده کردن جلوه های ویژه و انیمیشن در وب سایت درون این فایل است.
- AtlasUIMap
ابزارهایی برای استفاده از Virtual Earth

فایل‌های دیگر

پروژه اطلس فایل‌های ذیل را نیز به ریشه پروژه اضافه میکند:

- Default.aspx و Default.aspx.cs
صفحه ویی است که شامل کنترل Script Manager است که مسوولیت پردازش بلاک‌های اسکریپت مربوط به سمت کلاینت میباشد. یک اسکریپت سمت کلاینت که از نوع test/xml-script است به صفحه اضافه میشود. این اسکریپت برای نوشتن اسکریپت‌ها با استفاده از XML استفاده میشود.
- Eula.rtf
- Readme.txt
- Web.Config

این فایل برای اجرای پروژه اطلس لازم است. این فایل شامل تنظیماتی مخصوص اطلس است.

توضیح کد

با این برنامه کاربر قادر است روی صفحه با فشردن کلید چپ موس، خط بکشد. زمانی که کاربر کلید موس را رها میکند یا موس را به بیرون از صفحه میبرد، کشیدن خط به انتها میرسد. راه دیگری برای کشیدن خط با استفاده از JavaScript و VML وجود دارد ولی ما از VML استفاده نمیکنیم. صفحه Default یک عکس (از نوع عکس HTML – تگ IMG) خواهد داشت. رویدادهای موس بر روی عکس، توسط JavaScript گرفته میشود. JavaScript یک سری نقطه (نقطه های که در مسیر حرکت موس هستند) را به web service میفرستد. Web service عکس را با کشیدن خطهایی بین تمام نقطه هایی که کلاینت فرستاده است، بروزرسانی میکند و آنرا در Session قرار میدهد. در نهایت، کلاینت (Client) یک درخواست برای عکس بروز شده به سرور میفرستد. منبع عکس، یک HTTP handler است که عکس را بصورت یک جریان (stream) از session گرفته و به کلاینت میفرستد. کامپونتهای اصلی نرم افزار ما به شرح ذیل هستند:

Default.aspx

صفحه ای که شامل عکس پویا و کنترل مدیریت اطلس (Atlas Script Manager) است. (control)

ScribbleImage.ashx

یک HTTP handler که عکس ذخیره شده در Session را بصورت جریان (stream) در می آورد.

ScribbleService.asmx

Web service ی که تمام درخواستها به آن داده میشود. این web service عکس را تغییر میدهد.

Scribble.js

کدهای JavaScript نرم افزار که درون یک فایل قرار داده شده اند.

Global.asax

رویدادهای Session_Start و Session_End در این فایل کنترل میشوند. Session_Start یک متغیر Session ایجاد میکند و Session_End عکس ذخیره شده در Session را از حافظه خارج میکند (Dispose)
Global.asax

ما کد نویسی خود را از فایل Global.asax شروع میکنیم.

1. در منوی Website روی Add New Item کلیک کنید یا کلیدهای Ctrl + Shift + A را بفشارید.
2. در پنجره Add New Item، Global Application Class را انتخاب و ok را کلیک کنید. شما باید فایل Global.asax را ببینید که ایجاد شده است.
3. حال باید System.Drawing را به این فایل اضافه کنید. خط زیر را در ابتدای فایل بگذارید:

```
<%@ Import Namespace="System.Drawing" %>
```

کد زیر را به Session_Start اضافه کنید:

```
void Session_Start(object sender, EventArgs e)
{
    Bitmap bmp = new Bitmap(200, 200);
    using (Graphics g = Graphics.FromImage(bmp))
```

```

{
    g.FillRectangle(new SolidBrush(Color.White),
        new Rectangle(0, 0, bmp.Width, bmp.Height));
    g.Flush();
}
Session["Image"] = bmp;
}

```

4. این کد یک عکس (bitmap) 200 در 200 پیکسل برنگ سفید ایجاد میکند و آنرا درون متغیری به نام Image در Session میگذارد.

5. فانکشن Session_End این عکس را آزاد میکند.

```

Bitmap bmp = (Bitmap)Session["Image"];

bmp.Dispose();

```

6. از منوی Website ، Add Reference را انتخاب کنید.

7. در Add Reference ، System.Drawing را انتخاب و Ok کنید.

8. در نهایت از منوی Build ، Build Web Site را انتخاب کنید و یا Ctrl + Shift + B بفشارید تا مطمئن شوید اشکالی در کد وجود ندارد.

ScribbleImage.ashx

وظیفه این web handler این سات که عکس ذخیره شده در session را به صورت جریان (Stream) به کلاینت برگرداند.

1. در منوی Website روی Add New Item کلیک کنید یا کلیدهای Ctrl + Shift + A را فشار دهید.

2. در پنجره Generic Handler Add New Itemvg را انتخاب و اسم آنرا ScribbleImage.ashx بگذارید.

3. برای اینکه یک web handler بتواند از متغیر session استفاده کند باید Interface IRequiresSessionState را پیاده کند. نیازی برای پیاده سازی (override) هیچ متدی از این اینترفیس نیست و فقط کافی است که معرفی شود. خطی که کلاس را تعریف میکند به این شکل تغییر دهید:

```

public class ScribbleImage : IHttpHandler,
    System.Web.SessionState.IRequiresSessionState

```

4. حال این کد را به ProcessRequest اضافه کنید:

```

public void ProcessRequest (HttpContext context)
{
    context.Response.ContentType = "image/png";

    context.Response.Cache.SetNoStore();
}

```

```

context.Response.Cache.SetCacheability(HttpCacheability.NoCache);

context.Response.Cache.SetExpires(DateTime.Now);

context.Response.Cache.SetValidUntilExpires(false);


System.Drawing.Bitmap bmp =

(System.Drawing.Bitmap)context.Session["Image"];

lock(bmp)

{

    using (MemoryStream ms = new MemoryStream())

    {

        bmp.Save(ms, ImageFormat.Png);

        ms.Flush();

        context.Response.BinaryWrite(ms.GetBuffer());

    }

}

}

}

```

- خط نخست، ContentType را برابر image/png قرار میدهد. این کار باعث میشود که مرورگر وب جواب دریافتی از سرور را بصورت png بشناسد (بجای HTML).
- چهار خط بعدی به مرورگر وب تذکر میدهد که response را نباید Cache کند. هر چهار خط باید نوشته شود تا مطمئن شویم این کد روی انواع مرورگرها عمل میکند.
- در نهایت عکس داخل Session در داخل یک حافظه (memory stream) قرار میگیرد و محتوای حافظه روی response نوشته میشود. متد BinaryWrite را برای اینکار استفاده میکنیم چرا که عکس، یک داده باینری است.

ScribbleService.aspx

حال ما ابزاری برای ایجاد اولیه عکس در session و فرستادن آن به شکل response داریم. نیاز بعدی ما متدی است که خود عکس را عوض کند. ما از کلاینت انتظار صدا کردن web service، ScribbleService.asmx را برای کشیدن خطها داریم.

1. در منوی Website روی Add New Item کلیک کنید یا کلیدهای Ctrl + Shift + A را بفشارید.

2. در پنجره Add New Item، Web Service را انتخاب و پس از مشخص کردن نام ScribbleService.asmx کلیک ok را بفشارید. مطمئن شوید که گزینه Place Code in a Separate File را غیر فعال است.

3. خط زیر را به کد اضافه کنید:

```
Using System.Drawing;
```

4. در قدم بعدی باید یک کلاس برای point تعریف کنیم. توجه کنید که نمیتوانیم از System.Drawing.Point استفاده کنیم زیرا این کلاس XML serializable نیست. کد زیر را درست قبل از قسمت تعریف کلاس ScribbleService بگذارید.

```
public class Point
{
    public int X;
    public int Y;
};
```

5. در نهایت، باید متدی بنویسیم که با استفاده از مجموعه ای از نقاط، نقاشی کند. وب متد Draw به همین منظور به web service اضافه میکنیم.

```
[WebMethod(EnableSession = true)]
public void Draw(Point[] points)
{
    Image scribbleImage = (Image)Session["Image"];

    lock(scribbleImage)
    {
```



```

using (Graphics g = Graphics.FromImage(scribbleImage))

using(Pen p = new Pen(Color.Black, 2))

{
    if (points.Length > 1)
    {
        int startX = points[0].X;

        int startY = points[0].Y;

        for (long i = 1; i < points.Length; i++)
        {
            g.DrawLine(p, startX, startY,

                points[i].X, points[i].Y);

            startX = points[i].X;

            startY = points[i].Y;

        }
    }
}

```

- WebMethod(EnableSession = true) امکان دسترسی به متغیرهای session را به web service میدهد.
- عکس به حالت قفل درمی آید تا مشکلات استفاده همزمان پیش نیاید.
- خود عمل نقاشی، کاملاً ساده است. تنها کار لازم وصل کردن نقاط به همدیگر است.

Scribble.js

تمام کدهایی که برای سمت سرور است، نوشتیم. حال نوبت به اسکریپتی در سمت client داریم که نقاط را از رویداد موس گرفته و به web service تحویل دهد.

1. در solution explorer، فولدر ScriptLibrary را انتخاب کنید.
2. در منوی WebSite روی Add New Item کلیک کنید و یا Ctrl + Shift + A را بفشارید.



3. در پنجره Add New Item، Jscript File را انتخاب و نام را Scribble.js تعیین کرده و Ok را بفشارید. با اینکار فایل Scribble.js را درون فولدر ScriptLibrary قرار دادید.

4. در قدم بعدی باید چند متغیر عمومی تعریف کنیم.

```
//The HTML image element that is to be drawn

var image;

//The source of the image

var originalSrc;

//The number of iteration

var iter = 0;

//The array of points

var points = null;
```

توضیحات بالای هر تعریف متغیر، هدف هر کدام را توضیح میدهد. متغیر iter برای تغییر منبع عکس، بعد از هر بار درخواست از سرور است. برای مرورگر IE، image.src = image.src عکس را بروز می‌رساند ولی همین کد در Firefox کار نمیکند. برای حل این مشکل ما متغیر iter را تعریف و بعد از هر درخواست کلاینت به سرور، یکی به مقدار آن اضافه میکنیم. با اضافه کردن این عدد به متغیر originalSrc، مرورگر فکر میکند که هر بار باید اطلاعات جدید را، بجای عکس Cache شده از سرور بگیرد.

5. یک فانکشن به نام startStroke تعریف میکنیم تا در زمان رویداد mousedown اجرا شود.

```
function startStroke()

{

    points = new Array();

    window.event.returnValue = false;

}
```

وقتی کاربر شروع به نقاشی میکند. ما یک مجموعه از نقاط می‌سازیم (خط اول). خط دوم عمل پیش فرض برای یک عکس، شروع عمل mousedown رویداد را باطل میکند. این کار لازم است زیرا پیش فرض رویداد و جلوگیری از پیش آمدن رویدادهای دیگر است. Drag & Drop

6. وقتی عمل نقاشی با صدا شدن رویداد mouseup یا رویداد mouseout تمام میشود، میباید متد web service را صدا کنیم. این عملیات با فانکشن endStroke انجام میشود.

```

function endStroke()
{
    if (!points || points.length < 2)
        return true;

    //Send the points to the webservice
    ScribbleService.Draw(
        points,onWebMethodComplete,onWebMethodTimeout,onWebMethodError);

    points = null;

    window.event.returnValue = false;
}

```

قسمت جالب این کد خط

```

ScribbleService.Draw(
points, onWebMethodComplete, onWebMethodTimeout, onWebMethodError);

```

است که متد Draw از webservice را بصورت غیرهمزمان صدا میزند. این فانکشن بصورت اتوماتیک توسط محیط اطللس قابل دسترس است.

7. onWebMethodError فانکشنی است که در زمان بوجود آمدن مشکل در web service صدا زده میشود. و متد onWebMethodTimeout زمانی فعال میشود که صدا زدن متد web service از زمان مشخص قابل تنظیمی بیشتر طول بکشد (این زمان در فریم ورک اطللس قابل تنظیم است).

```

function onWebMethodError(fault)
{
    alert("Error occured:\n" + fault.get_message());
}

```

```

function onWebMethodTimeout()
{
    alert("Timeout occured");
}

```

8. متد onWebMehtodComplete زمانی صدا زده میشود که صدا زدن web method با موفقیت انجام پذیرد. در این مرحله عکس باید دوباره load شود.

```
function onWebMethodComplete(result, response, context)
{
    //We need to refresh the image

    var shimImage = new Image(200, 200);

    shimImage.src = originalSrc + "?" + iter++;

    shimImage.onload = function()
    {
        image.src = shimImage.src;
    }
}
```

ما یک متغیر shimImage از نوع Image میسازیم و منبع آن را منبع اصلی عکسی را که کشیدیم قرار میدهیم. وقتی شی عکس بارگذاری میشود، و منبع (source) عکس HTML واقعی را، برابر یک شی image موقت قرار میدهیم. این کار برای جلوگیری از پرش عکس انجام میشود.

9. لازم است که زمان رویداد mousemove آرایه نقاط را پر کنیم. این کار بار فانکشن addPoints انجام میگردد.

```
function addPoints()
{
    if (points)
    {
        var point = { X : window.event.offsetX,
                       Y : window.event.offsetY};

        points.push(point);

        if (points.length == 3)
        {
            endStroke();
        }
    }
}
```

```

        points = new Array();

        points.push(point);

    }

    window.event.returnValue = false;

}

}

```

- 0 نقطه جدید با استفاده از دو مشخصه `offsetX` و `offsetY` از شی رویداد ساخته شده و سپس به آرایه `points` اضافه میشود. دو مشخصه `offsetX` و `offsetY` موقعیت نسبی موس را در المنت `HTML` را میدهد.
- 0 وقتی طول آرایه به 3 رسید، ما بصورت انوماتیک درخواست را به سرور میفرستیم تا نقاشی انجام شود و آرایه را خالی میکنیم. این کار برای اینکه کاربر در حین حرکت موس، نقاشی را ببینند انجام میدهیم.

10. در انتها فانکشنهای نوشته شده باید به رویدادها بچسبند.

```

function pageLoad()
{
    var surface = document.getElementById("drawingSurface");

    image = surface.getElementsByTagName("IMG")[0];

    originalSrc = image.src;

    surface.attachEvent("onmousedown", startStroke);

    surface.attachEvent("onmouseup", endStroke);

    surface.attachEvent("onmouseout", endStroke);

    surface.attachEvent("onmousemove", addPoints);
}

```

- o فانکشن `pageLoad` یک فانکشن مخصوص است که بعد از بارگذاری کامل فریم ورک اطلس اجرا میشود.
- o ما از این فانکشن بجای رویدادهای معمول `window` یا `body` استفاده میکنیم تا مطمئن شویم که فریم ورک بارگذاری شده است.
- o المنت عکس درون `HTML` درون یک تگ `div` قرار گرفته که نام آن `drawingSurface` است. اندازه آن دقیقاً برابر عکس ما است (200 در 200) بنابراین با خیال راحت میتوانیم رویدادها را به `div` بچسبانیم.

Default.aspx

کامپوننت‌های ما باید درون صفحه Default.aspx قرار بگیرند. کد این صفحه به شرح ذیل است:

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

<!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 runat="server">

    <title>Atlas Scribble Sample</title>

</head>

<body>

    <form id="form1" runat="server">

        <Atlas:ScriptManager ID="AtlasScriptManager" runat="server"
            EnableScriptComponents="False" >

            <Services>

                <Atlas:ServiceReference Path="ScribbleService.asmx" />

            </Services>

            <Scripts>

                <Atlas:ScriptReference Path="ScriptLibrary/Scribble.js" />

            </Scripts>

        </Atlas:ScriptManager>

        <div id="drawingSurface"
            style="border:solid 1px black;height:200px;width:200px">

        </div>

    </form>

</body>

</html>
```

```
</form>

</body>

</html>
```

مهمترین هدف این صفحه کنترل سروری atlas:ScriptManager است. کنترل سروری ScriptManager مسئول ایجاد کدهای سمت کلاینتی برای اطلس و هر گونه کد مربوط به web service است. با هم استفاده از این کنترل را در Default.aspx بررسی میکنیم:

1. مشخصه EnableScriptComponents برابر با false میشود. این کار باعث میشود که بجای Atlas.js از AtlasRuntime.js استفاده شود. ما نسخه جمع و جور تر فریم ورک را ترجیح میدهم زیرا استفاده ای از کامپوننتها یا کنترلهای اطلس نکرده ایم.
 2. ما یک serviceReference به web service خودمان، ScribbleService.asmx داده ایم. این کار URL دسترسی به این web service را درون اسکریپتهای کلاینتی میگذارد.
 3. همچنین Scribble.js را نیز معرفی کرده ایم.
- این صفحه تمام کارهای ما را یک جا میکند. حال خود باید نتیجه کار را با اجرا کردن این کدها ببینید.

جادوی اطلس

کارهایی که اطلس برای ما انجام میدهد:

1. اطلس به ما اجازه میدهد نرم افزارهای وبی را بدون نگرانی از نوع مرورگرها بنویسیم. صدا زدن web service و استفاده از رویدادهای سمت کلاینت روی هر دو مرورگر IE و Firefox کار میکند. اطلس به صورت اتوماتیک تعاریفات لازم را روی شی های Firefox انجام میدهد تا آنها شبیه شی های IE شوند. برای دیدن چگونگی این کار فایلی AtlasCompat.js را ببینید.

2. اطلس به طور اتوماتیک یک کد JavaScript برای web service ما میسازد.

کجا هستیم

ما دیدیم که چگونه میتوانیم با کمک اطلس یک web service را از سمت کلاینت صدا بزنیم بدون نگرانی از نوع مرورگر.

دانلود و اجرای کد

1. Atlas Web Site را دانلود و نصب کنید.
2. بعد از اینکار یک web site اطلس را بسازید (از منوی File->New و انتخاب ASP.NET 'Atlas' Web Site)
3. فایل کد ما را بعد از دانلود، باز کرده و در یک فولدر جدید بریزید.
4. در منوی Website گزینه Add Existing Item را انتخاب و فایلهای ScribbleService.asmx و ScribbleImage.ashx را به ریشه پروژه اضافه کنید. و فایلهای Scribble.js و ScriptLibrary را به فولدر ScriptLibrary اضافه کنید.
5. Web site را اجرا کنید

یکی از ویژگی های مهم برنامه های وب ، تبعیت آنان از معماری "سرویس گیرنده - سرویس دهنده" است . این بدان معنی است که پیاده کنندگان برنامه های وب می توانند به منظور تحقق پردازش های سمت سرویس دهنده و سرویس گیرنده از فن آوری های متعددی استفاده نمایند . یکی از نکات مهم در خصوص انجام پردازش های سمت سرویس گیرنده ، میزان وابستگی آنان به اطلاعات موجود در سمت سرویس دهنده است . به عبارت دیگر ، اجرای يك event handler در سرویس گیرنده تا چه میزان وابسته به کد سمت سرویس دهنده است و به منظور انجام آن چه میزان داده می بایست بین سرویس گیرنده و سرویس دهنده مبادله گردد ؟ صرفنظر از این که به سوال فوق چه پاسخی داده می شود ، واقعیت این است که به منظور مدیریت رویدادهای محقق شده در سمت سرویس گیرنده ، می بایست ملزومات مورد نیاز ایجاد تا پیاده کنندگان بتوانند با استفاده از آنان پردازش های سمت سرویس گیرنده را مدیریت نمایند.

پردازش های سمت سرویس گیرنده از گذشته تاکنون

با توجه به این که پردازش های سمت سرویس گیرنده در برنامه های وب می بایست مستقل از نوع پلت فرم باشند ، بدیهی است که تمامی تلاش های انجام شده در این عرصه ، می بایست متمرکز بر روی برنامه های مرورگر باشد تا با ایجاد پتانسیل هایی در آنها ، امکان انجام پردازش های سمت سرویس گیرنده فراهم گردد . ظهور زبان های اسکریپت نویسی نظیر جاوااسکریپت و تجهیز مرورگرها به برنامه های مفسر مربوطه از جمله اقدامات عملی در این عرصه است . استفاده از زبان جاوااسکریپت به منظور کدینگ پردازش های سمت سرویس گیرنده دارای قدمتی چندین ساله است . در ادامه ، قابلیت های جدیدی به مرورگرها اضافه گردید تا پیاده کنندگان بتوانند به کمک آنان برنامه های وب سمت سرویس گیرنده را ایجاد نمایند . هم اینک ، تمامی مرورگرهای متداول از يك مدل شی گراء موسوم به (DOM برگرفته از document object model) استفاده می نمایند و تعداد اندکی از آنها از يك ویژگی جدید با نام XMLHttpRequest استفاده می نمایند که به کمک آن ، سرویس گیرندگان و سرویس دهندگان می توانند بدون نیاز به انجام يك postback کامل و round trip با یکدیگر و به صورت مستقیم ارتباط برقرار نمایند .

XMLHTTP (شامل مجموعه ای (API رابط برنامه نویسی) است که امکان ارسال و یا دریافت داده به صورت باینری ، HTML و XML را از سرویس دهندگان وب بر روی اینترنت و به کمک پروتکل HTTP فراهم می نماید . در مواردی که نیاز به داده موجود در سمت سرویس دهنده می باشد ، XMLHttpRequest به صورت پیوسته اقدام به ارسال درخواست خود برای سرویس دهنده می نماید تا آخرین اطلاعات را بدون نیاز به refresh کردن مدام مرورگرها ، بازیابی نماید . در واقع ، به کمک فن آوری فوق ، سرویس گیرندگان قادر به مبادله غیرهمزمان با سرویس دهنده بوده و می توانند اقدام به ارسال و یا دریافت داده XML بدون نیاز به انجام يك round trip کامل که باعث تولید مجدد يك صفحه می گردد ، نمایند .

ماحصل این تحولات ، ظهور نسل جدیدی از برنامه های وب نظیر [Microsoft VirtualEarth](#) و [Microsoft Windows Live](#) است . ایجاد چنین برنامه های وبی کار ساده ای نخواهد بود و پیاده کنندگان می بایست شناخت مناسبی نسبت به جاوااسکریپت و مدل DOM داشته باشند که ممکن است در هر مرورگر متفاوت باشد . علاوه بر این ، جاوااسکریپت تمامی ویژگی های يك زبان شی گراء را ارائه نمی نماید و بسیاری از ملزومات مورد نیاز پیاده کنندگان برنامه های وب در فریمورک دات نت را تامین نمی نماید (نظیر type-safe

ایجاد برنامه های وب با تاکید بر انجام پردازش سمت سرویس گیرنده ، نیازمند ملزوماتی نظیر يك زبان برنامه نویسی و پلت فرم پیاده سازی جدید است .

AJAX گرفته از Asynchronous JavaScript and XML

پیاده سازی برنامه های وب با استفاده از فن آوری های اشاره شده (اسکریپت نویسی سمت سرور گیرنده و مبادله غیرهمزمان با سرورس دهنده) ، **AJAX** نامیده می شود **AJAX** . ، پیاده کنندگان را قادر به تولید صفحاتی می نماید که از توان پاسخگویی بسیار مطلوبی در سمت سرورس گیرنده متناسب با رویداد ایجاد شده ، برخوردار می باشند . چراکه آنها از اسکریپت های سمت سرورس دهنده برای دستیابی و مدیریت عناصر بخش رابط کاربر استفاده می نمایند . علاوه بر این ، با توجه به مبادله غیرهمزمان داده به منظور ارسال و دریافت داده ، امکان انجام عملیات مورد نظر بر روی داده بدون وقفه و ازدست دادن state وجود خواهد داشت . [Microsoft Virtual Earth](#) و برنامه سرورس گیرنده نامه های الکترونیکی [OutlookWeb Access](#) ، دو نمونه از برنامه های سبک **AJAX** ، می باشند .

Atlas تلاشی در جهت ایجاد يك پلت فرم پیاده سازی جامع :

فن آوری جدید ASP.NET با نام Atlas ، مجموعه ای از فن آوری های میکروسافت را شامل می شود که با تمرکز بر روی اصول **AJAX** ، سعی در توسعه و بهبود آن را دارد .

Atlas ، يك فن آوری جدید در عرصه پیاده سازی برنامه های وب ASP.NET است که کتابخانه های اسکریپت سرورس گیرنده را با فریمورک پیاده سازی مبتنی بر سرورس دهنده ASP.NET 2.0 ترکیب می نماید . در واقع ، Atlas به پیاده کنندگان برنامه های وب يك پلت فرم مناسب به منظور پیاده سازی صفحات وب مبتنی بر سرورس گیرنده را ارائه می نماید که قبلاً مشابه آن در خصوص صفحات وب مبتنی بر سرورس دهنده توسط ASP.NET ارائه شده بود . با توجه به این که Atlas ، به عنوان يك پتانسیل اضافه در کنار ASP.NET مطرح می باشد ، بدیهی است که کاملاً سازگار با سرورس های مبتنی بر سرورس دهنده باشد .

با استفاده از Atlas ، می توان بخش قابل توجهی از پردازش های مورد نیاز يك برنامه را به سمت سرورس گیرنده انتقال داد . (fat-client) در چنین مواردی ، امکان ارتباط سرورس گیرنده با سرورس دهنده در background فراهم می گردد . محصل این فن آوری ، ایجاد برنامه های وبی است که علاوه بر ارائه امکانات مناسب در لایه رابط کاربر (UI) ، دارای توان پاسخگویی بالایی می باشند و به سادگی می توانند با سرورس دهنده ارتباط برقرار نمایند .

مفاهیم فنی اولیه AJAX

در **AJAX** ، پیاده کنندگان می توانند سرورس های وب را بطور غیرهمزمان و از طریق اسکریپت های سمت سرورس گیرنده و با استفاده از شی XMLHttpRequest فراخوانده تا اطلاعات به عنوان داده XML بسته بندی و در ادامه آنها را بر روی شبکه ارسال نمایند . بدین منظور شی XMLHttpRequest يك شی پراکسی را ارائه می نماید تا به کمک آن امکان فراخوانی از راه دور و مبادله داده فراهم گردد . در فن آوری **AJAX** از زبان های اسکریپت سمت سرورس گیرنده نظیر جاوااسکریپت بطرز گسترده ای استفاده می گردد .

از جاوااسکریپت در مواردی نظیر فراخوانی از راه دور روتین ها ، انجام پردازش های مورد نیاز يك برنامه در سمت سرورس گیرنده و ایجاد ویژگی های پیشرفته ای در سطح لایه رابط کاربر استفاده می شود . **AJAX** علاوه بر جاوااسکریپت از چندین فن آوری دیگر در سمت سرورس گیرنده نیز استفاده می نماید :

• **DOM** : به کمک فن آوری فوق ، عناصر موجود در يك صفحه HTML به عنوان مجموعه ای از اشیاء استاندارد (نظیر document و windows در نظر گرفته می شوند . بدین ترتیب ، امکان دستیابی و انجام عملیات دلخواه بر روی آنان از طریق کد فراهم می گردد .

(**DHTML** • برگرفته از : (DynamicHTML فن آوری فوق ، توانمندی HTML را به منظور واکنش مناسب در خصوص عملیاتی که توسط کاربران انجام می شود (نظیر درج داده) با استفاده از اسکریپت های سمت سرورس گیرنده و بدون نیاز به يك roundtrip افزایش می دهد .

• **رفتارها** ، شامل روشی مناسب به منظور برخورد سیستماتیک با عملیاتی نظیر drag and drop در

سطح لایه رابط کاربر و مرتبط با عناصر موجود بر روی يك صفحه می باشد .

• **عناصر** : اشیاء سفارشی شده جاوااسکریپت می باشند که پتانسیل های توسعه یافته ای را در سمت سرویس گیرنده ایجاد می نمایند .

چالش های فن آوری AJAX

برنامه نویسی صفحات به سبك **AJAX** دارای چالش های متعددی است :

• عناصر موجود در صفحات وب می بایست متناسب با شرایط هر مرورگر برنامه نویسی گردند ، چراکه هر مرورگر يك نسخه متفاوت از DOM و DHTML را ارائه می نمایند(هر چند این تفاوت ها اندك باشد .)

• برنامه نویسی سمت سرویس گیرنده صرفاً" با استفاده از جاوااسکریپت انجام می شود . پیاده سازی برخی از پتانسیل های **AJAX** می تواند برای پیاده کنندگان بسیار پیچیده باشد و نیازمند دانش بالائی در خصوص استفاده از جاوااسکریپت است .

• جاوا اسکریپت ، ویژگی ها و امکانات مورد نیاز پیاده کنندگان برنامه های دات نت را تامین نمی نماید (نظیر يك رویکرد شی گراء کامل) . علاوه بر این ، در این فن آوری از کتابخانه ای نظیر آنچه در پلت فرم دات نت ارائه شده است ، استفاده نمی گردد و برنامه نویسان می بایست تمامی برنامه را از ابتدا کد نمایند .

• جاوااسکریپت و پیاده سازی سمت سرویس گیرنده ، عموماً" بخوبی در IDEs حمایت نمی گردند .

فن آوری Atlas ، مسائل اشاره شده را با ارائه يك فریمورك كامل برای ایجاد برنامه های وب مبتنی بر سرویس گیرنده برطرف می نماید.

فن آوری Atlas ،

دارای عناصر سرویس گیرنده و سرویس دهنده ای است که آن را به خوبی با ASP.NET یکپارچه و مرتبط می نماید

مفاهیم فنی اولیه معماری Atlas

در ساده ترین حالت ، يك برنامه Atlas ، از کتابخانه های سمت سرویس گیرنده به منظور مدیریت رابط کاربر و فراخوانی عناصر مبتنی بر سرویس دهنده استفاده می نماید Atlas . ، همچنین عناصر سمت سرویس دهنده ای را ارائه می نماید که با سرویس گیرنده مرتبط و می توانند اسکریپت سمت سرویس گیرنده از قبل تعریف شده ای را تولید نمایند .

عناصر سمت سرویس گیرنده Atlas

فن آوری Atlas برای پیاده سازی برنامه های سمت سرویس گیرنده ، از مجموعه ای کتابخانه سمت سرویس گیرنده (فایل هائی با انشعاب (. JS استفاده می نماید که پیامد آن تعریف يك رويکرد لایه ای برای ایجاد برنامه های مبتنی بر سرویس گیرنده است . این لایه ها عبارتند از :

• **لایه مختص مرورگرها :** با استفاده از پتانسیل های ارائه شده توسط این لایه ، اسکریپت های Atlas در اکثر مرورگر سازگار بوده و ضرورتی به نوشتن اسکریپت های مختص يك مرورگر وجود ندارد .

• **سرویس های هسته :** atlas شامل ضمانتی به جاوااسکریپت نظیر کلاس ها ، namespace ، eventhandler، توارث ، نوع های داده و تسلسل اشیاء است . ویژگی های فوق يك مدل برنامه نویسی شی گراء را در اختیار پیاده کنندگان قرار می دهد تا به کمک بتوان علاوه بر سرعت در ایجاد برنامه ها از کد تولید شده نیز بدفعات استفاده نمود .

• **کتابخانه کلاس پایه :** Atlas شامل عناصری نظیر دیباگرها ، Timers ، ردیابی و string buliders است .

• **لایه شبکه :** این لایه با سرویس های مبتنی بر وب و برنامه ها ارتباط برقرار می نماید و مدیریت فراخوانی متدها از راه دور و غیرهمزمان را برعهده می گیرد . لایه فوق ، مدیریت فراخوانی غیرهمزمان بروی XMLHTTP را برعهده گرفته و باعث می گردد که پیاده کنندگان بدون این که درگیر پیچیدگی های موجود شوند با نوشتن چندین خط کد ، متدهای مورد نظر را فراخوانند .

• **لایه : UI** در این لایه قابلیت های سرویس گیرنده Atlas نظیر رفتارها ، گرامر تعریفی Atlas ، عناصر UI و نسبت دهی داده ارائه می گردد .

• **لایه کنترل ها :** این لایه کنترل های مختص atlas را برای پیاده سازی سمت سرویس گیرنده ایجاد می نماید . علاوه بر این که می توان از طریق اسکریپت به این کنترل ها دستیابی داشت ، امکان انجام عملیات متفاوتی نظیر نسبت دهی داده نیز وجود دارد. کنترل های Navigation و data-bound listview نمونه هائی در این زمینه می باشند .

• **يك مدل برنامه نویسی تعریفی** که پیاده کنندگان را قادر می سازد عناصر atlas را با روشی مشابه کنترل های سرویس دهنده ASP.NET ایجاد نمایند .

فن آوری Atlas را می توان به عنوان کتابخانه های اسکریپت سرویس گیرنده تصور نمود که زیرمجموعه ای از معماری ASP.NET بر روی سرویس دهنده می باشند

برای پیاده سازی برنامه های وب سمت سرویس گیرنده ، فن آوری Atlas ویژگی های متعددی را ارائه می نماید . ارائه يك مجموعه API برای پیاده سازی در جاوااسکریپت ، قابلیت سازگاری اتوماتيك با مرورگرها و يك مدل تعریفی برای پیاده سازی سمت سرویس گیرنده ، نمونه هائی در این زمینه می باشند .

عناصر سمت سرویس دهنده Atlas

فن آوری Atlas ، صرفاً در ارتباط با اسکریپت های سمت سرویس گیرنده نمی باشد و از عناصر سمت سرویس دهنده ، سرویس ها و کنترل هائی استفاده می نماید که می توانند با اسکریپت های Atlas سمت سرویس گیرنده مرتبط گردند :

• **سرویس های وب** که ویژگی های ASP.NET نظیر سرویس های پروفایل ، roles ، membership ، globalization و personalization را ارائه می نمایند .

• **کنترل های سرویس دهنده Atlas** که کنترل های سرویس دهنده ASP.NET را reassemble می نمایند ولی اسکریپت های سمت سرویس گیرنده Atlas را منتشر می نمایند . این نوع کنترل ها ارتباط بسیار نزدیکی با کنترل های سرویس دهنده ASP.NET نظیر دکمه ها ، Label و ... دارند .

• **کنترل های سرویس دهنده Atlas** که باعث می گردند جاوااسکریپت رفتارهای سمت سرویس گیرنده

را تولید نماید . کنترل های HoverBehavior ، ClickBehavior ، Popup و AutoBehavior نمونه هائی در این زمینه می باشند .

تمامی کنترل های سرویس دهنده atlas در ویژوال استودیو نیز قابل استفاده می باشند. بنابراین می توان از آنان در زمان طراحی استفاده نمود (همانند کنترل های سرویس دهنده . ASP.NET)

فن آوری Atlas ، اسکریپت نویسی سمت سرویس گیرنده را با پیاده سازی سمت سرویس دهنده ASP.NET یکپارچه می نماید و پیاده کنندگان می توانند از پتانسیل های ASP.NET در سمت سرویس دهنده برای برنامه های Atlas نیز استفاده نمایند .

هدف اولیه فن آوری Atlas ،

ترکیب ویژگی اسکریپت های سمت سرویس گیرنده با ویژگی هائی از ASP.NET بر روی سرویس دهنده است تا به کمک آن یک پلت فرم پیاده سازی جامع و فراگیر ایجاد گردد.

تهیه شده در شرکت سخاروش

و ابزارهای برنامه **Ajax** پیاده سازی های مختلف این راهنما حاوی لینک های سودمندی درباره مرتبط با آن است نویسی

شرح:

و ابزارهای برنامه نویسی **Ajax** پیاده سازی های مختلف در این صفحه می توانید لینک های سودمندی درباره را مطالعه **Ajax** بديده انقلابي > می توانید یادداشت **Ajax** مرتبط با آن پیدا کنید. برای آشنایی اولیه با آمده است < Ajax همه چیز درباره > کنید. شرح کامل این تکنیک در مقاله

کتابخانه های جاوا اسکریپت

دارد Google Suggest اسکریپتی که کارکردی مشابه سیستم کلاس های جاوا - [Suggest Framework](#) تکنیک های مختلف دسترسی به دیتا به کمک جاوا و جاوا اسکریپت. حاوی یک - **AJAX** مثالهای **AJAX** درباره مشکلات برنامه نویسی با PowerPoint بر روی کارکرد یک برای شبیه سازی تاثیر کندي و قطع شدن ارتباط شبکه HTTP Proxy یک - **AJAX Proxy** برنامه ای جکس را می گیرد و به هر تابع جاوا اسکریپتی مورد نظر HTML اطلاعات یک فرم مناسب که Toolkit یک - **AJForm** می کند ارسال را آسان می کند Perl با **AJAX** برنامه های کار نوشتن Toolkit این - [CGI::Ajax](#) پشتیبانی نیز **AJAX** که از (OOP) اسکریپتی این سورس و شیء گرا یک فریم ورک جاوا - [Prototype](#) می کند **AJAX** جالب با پشتیبانی کامل از یک کتابخانه جاوا اسکریپتی - [Rico](#) ساده Toolkit یک - [Sajax](#) Prototype اسکریپتی مرتبط با فریم ورک یک کتابخانه جاوا - [Script.aculo.us](#) وابسته و عناصر XMLHttpRequest بدون AJAX - [Alternative AJAX techniques](#) XMLHttpRequest مدیریت شیء - [XHRConnection](#)

ASP.NET

در ASP.NET 2.0 **AJAX** پیاده سازی کامل - [Atlas](#) نیز هست روی پلتفرم دات نت که رایگان AJAX اولین پیاده سازی - [Ajax.NET Professional](#) های مناسب برای Web Control شامل **AJAX** یک کتابخانه دات نت رایگان برای - [BorgWorX Ajax.NET](#) ASP.NET ASP.NET کنترل های ای جکس برای استفاده در ویژوال استودیو و - [PowerWEB LiveControls for ASP.NET](#) روی وب سرویس متمرکز شده است که ASP.NET یک موتور ای جکس رایگان برای - [Aspects of Ajax](#) xmlhttp روی ASP.NET فرم های در Post Back ارسال - [OutPost](#) ASP.NET برای غنی تر کردن ساختار صفحات **AJAX** بر یک فریم ورک مبتنی - [ComfortASP.NET](#) کاربرد آسان یک کتابخانه ای جکسی با - [MonoRail](#) ASP.NET در محیط **AJAX** کلاینت/سروری برای پیاده سازی رهیافت ترکیبی - [Bitkraft](#)

PHP

در PHP **AJAX** برای پیاده سازی Toolkit یک - [XAJAX](#) از طریق جاوا اسکریپت سمت کلاینت در PHP Server-side کلاس های کتابخانه ای برای فراخواندن - [NAJAX](#) **AJAX** ساده برای Toolkit یک - [SAJAX **AJAX** برای پیاده سازی Toolkit یک - \[PAJAX\]\(#\) PHP پیاده سازی ای جکس در - \[HTML Ajax\]\(#\)](#)

Java

که از مدل سه لایه J2EE و AJAX نویسی سریع (برای برنامه) RAD یک محیط - [ThinkCAP JX](#) برای کاستن از میزان کدنویسی جاوا اسکریپت استفاده می کند client/server event model برای جاوا یک کتابخانه ای جکسی رایگان - [Echo2](#)

نوشته شده JSP برای ارتقاء نرم افزارهای کنونی تحت وب که با یک پروژه این سورس - [AjaxAnywhere](#) باشند

که کار با ایجکس در این پلاتفرم را آسان می کند JSP های رایگان Tag یک سری - [AjaxTags](#)

امکان HTML را بدون نیاز به نوشتن جاوا اسکریپت در AJAX تکنیک یک فریم ورک برنامه نویسی که - [Guise](#) پذیر می کند

جاوا یک کتابخانه مجانی برای - [DWR](#)

سمت کلاینت و سمت (synchronize) برای هماهنگ کردن AJAX یک فریم ورک این سورس که از - [jWic](#) سرور استفاده می کند

نوشتن برنامه های ایجکسی برای JSF یک فریم ورک - [ICEfaces](#)

پلاتفرم مالتی

ASP و PHP نویسی برای پیاده سازی ایجکس در یک کتابخانه کامل برنامه - [CPAINT](#)

ASP.NET با استفاده از جاوا یا تولید برنامه های ایجکسی - [GeneXus](#)

AJAX برای برای برنامه های مبتنی بر پلاتفرم جاوایی و دات نت - [WebORB](#)

Enterprise ایجکس برای محیط های - [JackBe](#)

1.



2.

<http://www.codeproject.com/books/AJAXFutureWeb.asp>



3.

<http://www.ajaxpro.info/Examples/DataType/default.aspx>

