

Ajax : تحولی بزرگ در عرصه وب (بخش ششم)

آنچه تاکنون گفته شده است :

[بخش اول](#) تاثیر متقابل وب و نرم افزار بر یکدیگر

[بخش دوم](#) Ajax و فناوری های مرتبط با آن

[بخش سوم](#) بررسی نمونه برنامه های مبتنی بر Ajax

[بخش چهارم](#) برنامه نویسی غیرهمزمان در برنامه های وب

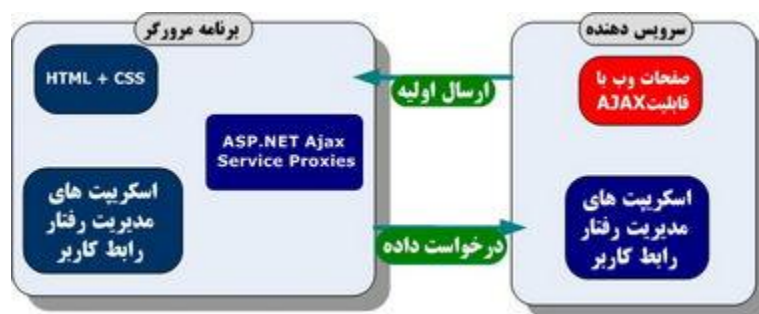
[بخش پنجم](#) بررسی معماری ASP. NET Ajax

در [بخش پنجم](#) ضمن اشاره به ضرورت استفاده از يك فریمورک در برنامه های وب مبتنی بر Ajax ، با عناصر موجود در سمت سرویس دهنده و سرویس گیرنده فریمورک ASP.NET AJAX آشنا شدیم . در این بخش با نحوه تعامل این عناصر در جهت تامین خواسته پیاده کنندگان برنامه های وب آشنا خواهیم شد . بدین منظور به بررسی دو سناریوی مختلف خواهیم پرداخت : مدل پیاده سازی با محوریت سرویس گیرنده و مدل پیاده سازی با محوریت سرویس دهنده .

طراحی انعطاف پذیر معماری ASP. NET Ajax ، دو رویکرد و یا مدل پیاده سازی مختلف را ارائه می نماید . به اولین سناریو که در سمت سرویس گیرنده پیاده سازی می گردد ، "مدل پیاده سازی با محوریت سرویس گیرنده" و به دومین رویکرد که مبتنی بر سرویس دهنده است ، "مدل پیاده سازی با محوریت سرویس دهنده" گفته می شود . قبل از هر چیز لازم است با نحوه عملکرد هر یک از مدل های فوق بیشتر آشنا شویم تا بتوانیم از هر یک در زمان مناسب استفاده کنیم .

مدل پیاده سازی با محوریت سرویس گیرنده

در این مدل ، لایه Presentation متاثر از اسکریپت های سمت سرویس گیرنده با بکارگیری DHTML و جاوا اسکریپت است. این بدان معنی است که یک برنامه با هوشمندی و تعامل بیشتر ، از طریق سرویس دهنده برای سرویس گیرنده ارسال می گردد (در زمان استقرار صفحه در حافظه برای مرتبه اول) . پس از آن ، تعامل بین برنامه مرورگر و سرویس دهنده محدود به بازیابی داده مورد نیاز جهت بهنگام سازی صفحه است . در این مدل کاربران با برنامه تعامل زیادی خواهند داشت (برنامه ای که در سمت سرویس گیرنده و در مرورگر کاربر اجرا شده است) . شکل 1 ، مدل پیاده سازی با محوریت سرویس گیرنده را نشان می دهد .



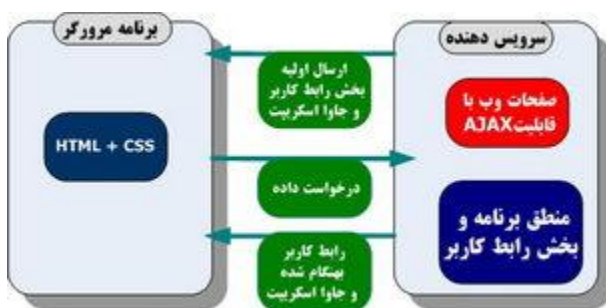
شکل 1 : مدل پیاده سازی با محوریت سرویس گیرنده

مدل فوق برای برنامه هائی که اشتیاق زیادی به استفاده همه جانبه از ویژگی های DHTML دارند مناسب تر می باشد (نظیر برنامه های mashup). برنامه های mashup ، برنامه های وی بی می باشند که محتویات خود را از بیش از یک منبع خارجی دریافت و آنها را با یک مکانیزم مطلوب در اختیار کاربران قرار می دهند . سایت Pageflakes.com یک نمونه در این رابطه است . این نوع سایت ها در تعامل مستمر با کاربر می باشند . با توجه به این که لازم است از یک طرف صفحه سبک و با کارائی مطلوب باشد و از طرف دیگر از منابع سمت سیستم بطور هوشمندانه استفاده گردد ، پیاده سازی این نوع برنامه ها با محوریت سرویس گیرنده ، یک گزینه مطلوب می باشد .

مدل پیاده سازی با محوریت سرویس دهنده

در این مدل ، منطق برنامه و اکثر عناصر بخش رابط کاربر بر روی سرویس دهنده باقی می ماند . در ادامه ، تغییرات مورد نیاز رابط کاربر برای برنامه مرورگر ارسال می گردد ، در مقابل این که تغییرات از طریق اجرای اسکریپت های سمت سرویس گیرنده ایجاد شوند . رویکرد فوق ما را به یاد مدل سنتی صفحات ASP.NET می اندازد . مدلی که بر اساس آن ، سرویس دهنده بخش رابط کاربر را در هر مرتبه postback ایجاد و آن را برای مرورگر و به منظور تفسیر و نمایش یک صفحه جدید ارسال می نماید .

تفاوت مدل فوق با مدل سنتی صفحات ASP.NET در این است که صرفاً قسمت هائی از بخش رابط کاربر که می بایست تفسیر گردند برای برنامه مرورگر ارسال می شوند (در مقابل تمام صفحه) . مهمترین دستاورد رویکرد فوق ، بهبود محسوس میزان تعامل و تاخیر در برنامه های وب است . شکل 2 ، ماهیت مدل پیاده سازی با محوریت سرویس دهنده را نشان می دهد .



شکل 2 : مدل پیاده سازی با محوریت سرویس دهنده

رویکرد فوق برای بسیاری از پیاده کنندگان ASP.NET جالب است . چراکه در این مدل به پیاده کنندگان امکان نگهداری هسته رابط کاربر به همراه منطق برنامه بر روی سرویس دهنده داده می شود . عدم تاثیر پذیری این نوع برنامه ها از عملیاتی نظیر غیرفعال کردن جاوا اسکریپت در مرورگر ، باعث شده است که ادامه حیات و سرویس دهی آنها مستقل از پارامترهای تاثیرگذار در سمت سرویس گیرنده باشد . در زمان کار با کنترل هائی نظیر GridView و Repeater در ASP.NET ، مدل فوق ساده ترین و مطمئن ترین روش را ارائه می نماید .

اهداف ASP.NET Ajax

پس از آشنائی اولیه با معماری و برخی از ویژگی های ASP.NET Ajax ، بد نیست به اهداف و دستاوردهای این فریمورک برای پیاده کنندگان برنامه های وب نیز اشاره ای داشته باشیم .

- **استفاده ساده ، فریمورکی با کارآئی بالا** : پیاده کنندگان برنامه های وب تمایل دارند که به سادگی و با صرف وقت اندک بتوانند از پتانسیل های Ajax در برنامه های خود استفاده نمایند . فریمورک ASP.NET Ajax با ارائه یک کتابخانه قدرتمند در سمت سرویس گیرنده و یک مجموعه از کنترل های سرویس دهنده این امکان را در اختیار پیاده کنندگان برنامه های وب قرار می دهد که بتوانند به سادگی از امکانات فوق در جهت تامین خواسته های خود (برنامه های گذشته و برنامه های جدید) استفاده نمایند .
- **یکپارچگی مدل برنامه نویسی سرویس دهنده** : کنترل های سرویس دهنده ارائه شده به همراه فریمورک ASP.NET Ajax برای پیاده کنندگان برنامه های وب یک الگوی کاملاً شناخته شده می باشد . چراکه پیش از این پیاده کنندگان از کنترل های سرویس دهند متعددی در برنامه های وب ASP.NET استفاده کرده اند .
- **عناصر و ابزارهایی با کلاس جهانی** : عناصر و ابزارهایی که بر روی فریمورک ایجاد شده اند ، نه تنها قابلیت و توانمندیهای فریمورک را توسعه داده اند ، بلکه مجموعه ای از ابزارهای قدرتمند (نظیر اشکال زدائی ، tracing و profiling) را در اختیار جامعه بزرگ پیاده کنندگان قرار می دهد .
- **حمایت از پلت فرم های متعدد** : حمایت در IE ، فایرفاکس ، Safari و سایر مرورگرها این اطمینان را ایجاد می نماید که در زمان کار با مرورگرهای مختلف درگیر مشکلات مربوطه نخواهیم شد .

سادگی ، قابلیت توسعه ، ابزارهای قدرتمند و وجود هزاران پیاده کننده از مشخصه های مثبت فریمورک ASP.NET Ajax محسوب می گردد.

خلاصه

تاکنون و در طی شش مقاله با مواردی نظیر شی XMLHttpRequest ، برخی الگوهای Ajax استفاده شده در زمان پیاده سازی برنامه های وب قدرتمند با توانائی ایجاد تعامل بیشتر ، معماری ASP.NET AJAX و سناریوهای مختلف پیاده سازی آشنا شدیم .

در بخش بعد با استناد به موارد فوق با نحوه استفاده از این فریمورک آشنا خواهیم شد و صفحاتی را ایجاد خواهیم کرد که در آنها از Ajax استفاده شده باشد.