عنوان: **اشیاء context در ASP.NET**
نویسنده: کیانی مقدم علیرضا
تاریخ: ۱۳۹۴/۰۸/۰۸ ۹:۲۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC

در asp.net تعدادی اشیاء پایه، حاوی اطلاعات بسیار با ارزشی در خصوص درخواست جاری، application و پاسخی که ارسال می‌شود هستند و به صورت غیر مستقیم جهت دست‌یابی به قسمتهای مرکزی و هسته‌ای چهارچوب asp.net مانند , security stat data می‌توان این اشیاء را بکار گرفت.
بررسی این اشیاء از این جهت حائز اهمیت است که در کنترلرها و ویوها می‌توان پاسخهای ارسالی به کلاینت‌ها را بر حسب شرایط مختلفی مانند درخواست رسیده یا حالت خاص دیگری تغییر داد. ضمن اینکه از این اشیاء در ماژول‌ها و هندلرها نیز استفاده می‌شود. لذا قبل از پرداختن به مفاهیم مرتبط به ماژولها و هندلرها بهتر است این اشیاء بررسی شوند.

کلاس httpcontext هسته‌ی چهارچوب Asp.net است که خواص بسیار مهمی به شرح ذیل را فراهم می‌آورد:
Application : شیء HttpAplicationState را جهت مدیریت Application State Data بر می‌گرداند.
ApplicationInstance : شیء HttpApplication مرتبط با درخواست جاری را بر می‌گرداند.
Cache : شیء Cache را جهت کش داده‌ها بر می‌گرداند.
Current : خصوصیت استاتیکی که شیء HttpContext درخواست جاری را بر می‌گرداند.
CurrentHandler : وهله‌ای از IHttpHandler را که محتویات درخواست جاری را تولید خواهد کرد، بر می‌گرداند.
IsDebuggingEnabled : مقدار True یا False را بسته به اینکه Debug فعال باشد یا خیر بر می‌گرداند.
Items : یک Collection را که می‌تواند جهت انتقال State Data بین اجزاء مختلف فریم ورک که در پردازش یک درخواست همکاری می‌کنند، بر گرداند.
GetSection : قسمت خاصی از فایل Web.Config را بر می‌گرداند.
Request : شیء HttpRequest را که حاوی جزئیات درخواست پردازش شده است، بر می‌گرداند.
Response : شیء HttpResponse را بر می‌گرداند که حاوی جزئیات Response ای است که آماده ارسال به مرورگر است.
Session : برگرداننده‌ی شیء HttpSession است. این خاصیت قبل از وقوع رخداد PostAcquireRequestState مربوط به Application، مقدار null را خواهد داشت.
TimeStamp : شیء از نوع Datetime و معرف زمانی است که شیء HttpContext ایجاد شده‌است.
Trace : جهت ثبت اطلاعات تشخیصی به کار می‌رود.

در فایل global از طریق خاصیت Context به وهله‌ای از HttpContext دسترسی داریم. ولی این خاصیت فراگیر نبوده و در ویوها و کنترلرها از طریق خاصیت HttpContext در دسترس است که در کنترلر خاصیتی از کلاس پایه Controller و در ویو خاصیتی از کلاس پایه WebViewPage می‌باشد و در ماژول‌ها از طریق پارامتر ورودی متد Init در دسترس است.
در هندلرها متد ProcessRequest وهله‌ای از شیء HttpContext را دریافت می‌کند.
به طور سراسری در هر نقطه‌ای از برنامه از طریق خاصیت استاتیک HttpContext.Current به شیء HttpContext مرتبط با درخواست جاری دسترسی خواهیم داشت.
در کلاسهای مدل نیز از طریق خاصیت استاتیک یاد شده می‌توان به این شیء دسترسی داشت؛ ولی بهتر است اگر مدل نیاز به اطلاعاتی در خصوص یک Request داشت این اطلاعات از طریق اشیاء Context در کنترلر دریافت و به عنوان آرگومان به مدل ارسال شوند.
به صورت کلی، خلاصه‌ی مطالب فوق به صورت ذیل می‌باشد:

بخش مورد نظر	روش دسترسی به اشیاء context
Controller	از طریق خاصیت HttpContext
View	خاصیت Context
Global Application Class	خاصیت Context

بخش مورد نظر	روش دسترسی به اشیاء context
Modules	آرگومان متد Init
Handlers	آرگومان متد ProcessRequest
به صورت کلی	از طریق خاصیت استاتیک HttpContext.Current

خواص ذکر شده‌ی در جدول فوق، همگی نوع یکسانی ندارند. در فایل global و هندلرها و ماژولها، دقیقا به همان HtpContext ای دسترسی داریم که قبلا با آن‌ها آشنا شدیم. این نوع از اشیاء (منظور خواص کلاس HttpContext) قبل از اینکه فریم ورک MVC، دست به کار مدیریت یک درخواست شود، کار خود را شروع می‌کنند که این امر کار آزمایش واحد را مشکل می‌کند. برای رفع این مشکل خواص مهیا در کنترلر و ویو وهله‌هایی از کلاسهای متفاوتی را برمی گردانند که از کلاس Context مشتق شده‌اند و ضمنا آزمایش واحد را به سادگی پشتیبانی می‌کنند(+) . خاصیت HttpContext کلاس Controller وهله‌ای از کلاس HttpContextBase را بر می‌گردانند. همه‌ی اشیاء context موجود در این کلاس، دارای پسوند Base هستند؛ مانند : HttpRequestBase, HttpResponseBase و...

اگر نیاز داشتید که یک شیء از نوع پسوند دار Base را از یک شیء فاقد این پسوند ایجاد کنید، برای مثال یک شیء HttpRequest دارید، ولی متدی دارید که آرگومان آن وهله‌ای از کلاس HttpRequestBase است، برای این کار کلاسهایی با پسوند Wrapper مانند HttpContextWrapper ,... وجود دارند که از کلاسهایی با پسوند Base مشتق شده و در متد سازنده خود آرگومانی از کلاسهای اصلی فاقد پسوند Base را می‌گیرند.

ولی عکس آن امکان پذیر نیست و نمی‌توان کلاس با پسوند Base را به نوع اولیه برگرداند. ولی همان طور که گفتیم هر جایی که نیاز داشتیم، می‌توان از طریق خاصیت استاتیک System.Web.HTTPContext.Current به اشیاء Context اصلی دسترسی داشت.

بسیاری از کلاسهای فریم ورک Asp.Net دارای خاصیت‌هایی هستند که به خواص کلاس HttpContext نگاشت شده‌اند. یکی از این نمونه‌ها کلاس HttpApplication است که کلاس پایه‌ی فایل global نیز هست. همان طور که در ذیل نیز می‌بینید، بسیاری از خواص و متدهای کلاس HttpApplication به نوعی مرتبط با اشیاء context هستند.

Application : به خاصیت HttpContext.Application نگاشت شده‌است که دسترسی به state data application را مهیا می‌سازد.

CompleteRequest : به چرخه حیات جاری خاتمه خاتمه داده و آن را مستقیما به رخداد LogRequest منتقل می‌کند.

Context : شیء HttpContext مرتبط با درخواست جاری را بر می‌گرداند.

Init : هر گاه متد Init یکی از ماژولهای ثبت شده در برنامه فراخوانی شود، این متد صدا زده خواهد شد.

Modules : یک شیء HttpModuleCollection را که حاوی جزئیات ماژولهای برنامه است، بر می‌گرداند.

RegisterModule(type : متد استاتیکی است که یک ماژول جدید را ثبت می‌کند.

Request : مقدار HttpContext.Request را برگشت می‌دهد و در صورتی که این مقدار null باشد، استثنایی رخ خواهد داد.

Response : مقدار HttpContext.Response را برگشت داده و در صورت null بودن این مقدار، استثنایی رخ می‌دهد.

Server : به خاصیت HttpContext.Server نگاشت شده است.

Session : مقدار HttpContext.Session را برگشت می‌دهد که در صورت null بودن این مقدار، استثنایی رخ خواهد داد.

در این مقاله با خواص کلاس HttpContext که اشیائی مهم و پرکاربرد بوده و حاوی اطلاعات بسیار سودمندی هستند، آشنا شدیم. ضمنا همان طور که در ابتدای مقاله گفته شد، از این اشیاء در ماژول‌ها و هندلرها استفاده‌ی زیادی می‌شود.