

برای ری استارت کردن یک برنامه‌ی ASP.NET حتماً نیازی نیست تا IIS را متوقف و سپس راه اندازی کرد یا تنظیمات App pool برنامه را در IIS تغییر داد. روش‌های دیگری نیز وجود دارند که عدم آگاهی از آن‌ها می‌تواند سبب بروز مشکلات عدیده‌ای گردد و گاهی خطایابی آن‌ها بسیار مشکل است؛ زیرا ری استارت شدن برنامه = از دست رفتن آنی تمام سشن‌های InProc تمام کاربران سایت؛ پاک شدن کش برنامه در IIS؛ از دست رفتن تمام متغیرهای استاتیک، Application State و مواردی از این دست؛ - نوشتن در پوشه bin برنامه (ایجاد فایل یا ایجاد پوشه یا تغییر نام پوشه‌ها و مواردی از این دست). (بنابراین قرار دادن بانک اطلاعاتی برنامه در این پوشه، اشتباه بوده و به همین جهت پوشه‌ی استاندارد App_Data طراحی شده است) - نوشتن در فایل web.config برنامه (به صورت دستی، حتی در حد اضافه کردن یک فاصله یا توسط خود برنامه و یا استفاده از متد File.SetLastWriteTime). در این حالت ASP.NET FileSystemWatcher بلافاصله وارد عمل شده و برنامه را ری استارت می‌کند. (هدف اصلی وجودی این فایل، برخورد فقط خواندنی با آن است نه ذخیره سازی تنظیمات پویای برنامه در آن. برای ذخیره سازی تنظیمات پویا، از بانک اطلاعاتی و یا یک فایل XML مجزای از وب کانفیگ استفاده کنید یا مواردی مشابه) - هر گونه تغییری در فایل‌ها و یا پوشه‌های ، App_Code ، Global.asax ، machine.config ، App_WebReferences ، App_GlobalResources و App_LocalResources نیز سبب ری استارت برنامه خواهند شد. - با ایجاد، حذف یا تغییر نام یکی از ساب دایرکتوری‌های واقع شده در ریشه برنامه. بنابراین اگر برنامه‌ی شما به صورت پویا پوشه‌هایی را ایجاد یا حذف می‌کند باید منتظر ری استارت‌های پی در پی باشید (البته این مورد با از کار انداختن FileChangesMonitor مربوط به HttpRuntime قابل حل می‌باشد (+))، ولی همانطور که عنوان شد به صورت پیش فرض همواره فعال است) - فراخوانی متد System.Web.HttpRuntime.UnloadAppDomain شبیه به همان Application.Exit در برنامه‌های دسکتاپ است و بلافاصله سبب خاتمه‌ی برنامه می‌شود. قرار دادن فایل App_Offline.htm در پوشه اصلی برنامه نیز چنین رفتاری را سبب خواهد شد. علاوه بر آن تگ httpRuntime در وب کانفیگ نیز دارای گزینه‌ی enable است و تنظیم آن به false ، سبب خاتمه‌ی سرویس دهی برنامه خواهد شد. - رسیدن به عدد numRecompilesBeforeAppRestart تعریف شده در فایل machine.config که عموماً به عدد 15 تنظیم شده است. اگر تغییرات زیادی را در فایل‌های (مرتبط با ASP.NET مانند aspx ، asmx و غیره) برنامه داده باشید (بیشتر از 15 مورد) و نیازی به ری کامپایل اساسی وجود داشته باشد، ASP.NET FileSystemWatcher به صورت خودکار برنامه را ری استارت خواهد کرد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۱ ۱۰:۹

چند نکته‌ی تکمیلی

- اگر هاست شما تمام سایت‌ها را با یک [Application pool](#) مدیریت می‌کند، کرش یکی از چند ده سایت دیگر می‌تواند سبب ری استارت شدن سایت شما هم بشود؛ چون برنامه‌ها از همه ایزوله نشده‌اند. راه حل آن ایجاد یک [Application pool](#) مجزا به ازای هر سایت هست (توسط هاست‌دار).
- زمانیکه تمام سایت‌ها با یک Application pool واحد مدیریت می‌شوند، تمام آن‌ها توسط یک وهرله از w3wp.exe اجرا خواهند شد. با تعریف Application pool های مجزا، هر سایت، یک وهرله‌ی مجزا از w3wp.exe را به خود اختصاص خواهد داد. یعنی اگر Task manager سرور را بررسی کنید، به ازای هر سایت، یک w3wp.exe با pid مجزا قابل مشاهده است. به این ترتیب اگر pid=1234 کرش کرد، تاثیری روی pid=4321 نخواهد داشت.
- یک برنامه‌ی ASP.NET پس از مدتی بیکاری (قابل تنظیم در Application pool برنامه)، به صورت خودکار توسط IIS از حافظه خارج می‌شود. با درخواست بعدی که به آن برنامه می‌رسد (مثلا گشودن یک صفحه‌ی آن توسط یک کاربر)، مجدداً از صفر اجرا خواهد شد. این مورد نیز به معنای ری استارت کامل برنامه است.
- در تنظیمات Application pool موارد زیادی را می‌توان تنظیم کرد که سبب ری استارت شدن برنامه می‌شوند. برای مثال اگر مصرف CPU و یا حافظه‌ی برنامه به حد مشخصی رسید، برنامه ری استارت شود و امثال آن.