

عنوان: نگاهی به گزینه‌های مختلف مپهای جهت میزبانی SignalR

نویسنده: وحید نصیری

تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۲:۴۸

آدرس: www.dotnettips.info

برچسب‌ها: ASP.Net, jQuery, SignalR

حداقل چهار گزینه برای Hosting سرویس‌های Hub برنامه‌های مبتنی بر SignalR وجود دارند که تا به اینجا، مورد دوم آن بیشتر بررسی گردید:

OWIN (1)

ASP.NET Hosting (2)

Self Hosting (3)

Cloud و ویندوز Azure (4)

OWIN (1)

اگر به اسمبلی‌های همراه با SignalR دقت کنید، یکی از آن‌ها Microsoft.AspNet.SignalR.Owin.dll نام دارد. OWIN مخفف Open Web Server Interface for .NET است و کار آن ایجاد لایه‌ای بین وب سرورها و برنامه‌های وب می‌باشد. یکی از اهداف مهم آن ترغیب دنیای سورس باز به تهیه ماژول‌های مختلف قابل استفاده در وب سرورهای دات نت است. نکته‌ی مهمی که در SignalR و کلیه میزبان‌های آن وجود دارد، بنا شدن تمامی آن‌ها بر فراز OWIN می‌باشد.

ASP.NET Hosting (2)

بدون شک، میزبانی ASP.NET از هاب‌های SignalR، مرسوم‌ترین روش استفاده از این فناوری می‌باشد. این نوع میزبانی نیز بر فراز [OWIN](#) بنا شده است. نصب آن توسط اجرای دستور پاور شل ذیل در یک پروژه وب صورت می‌گیرد:

```
PM> Install-Package Microsoft.AspNet.SignalR
```

3) خود میزبانی یا Self hosting

خود میزبانی نیز بر فراز OWIN تهیه شده است و برای پیاده سازی آن نیاز است وابستگی‌های مرتبط با آن، از طریق NuGet به کمک فرامین پاور شل ذیل دریافت شوند:

```
PM> Install-Package Microsoft.AspNet.SignalR.Owin
PM> Install-Package Microsoft.Owin.Hosting -Pre
PM> Install-Package Microsoft.Owin.Host.HttpListener -Pre
```

مواردی که با پارامتر pre مشخص شده‌اند، در زمان نگارش این مطلب هنوز در مرحله بتا قرار دارند اما برای دمو برنامه کفایت می‌کنند.

مراحل تهیه یک برنامه ثالث (برای مثال خارج از IIS یا یک وب سرور آزمایشی) به عنوان میزبان Hubs مورد نیاز به این شرح هستند:

الف) کلاس آغازین میزبان باید با پیاده سازی اینترفیسی به نام IAppBuilder تهیه شود.

ب) مسیریابی‌های مورد نیاز تعریف گردند.

ج) وب سرور HTTP یا HTTPS توکار برای سرویس دهی آغاز گردد.

باید توجه داشت که در این حالت برخلاف روش ASP.NET Hosting، سایر اسمبلی‌های برنامه جهت یافتن Hub‌های تعریف شده، اسکن نمی‌شوند. همچنین هنگام کار با jQuery مباحث عنوان شده در مورد تنظیم دسترسی‌های Cross domain نیز باید در اینجا اعمال گردند. به علاوه اجرای وب سرور توکار آن به دلایل امنیتی، نیاز به دسترسی مدیریتی دارد.

برای پیاده سازی یک نمونه، به برنامه‌ای که تاکنون تهیه کرده‌ایم، یک پروژه کنسول دیگر را به نام ConsoleHost اضافه کنید. البته باید در نظر داشت در دنیای واقعی این نوع برنامه‌ها را عموماً از نوع سرویس‌های ویندوز NT تهیه می‌کنند.

در ادامه سه فرمان پاور شل یاد شده را برای افزودن وابستگی‌های مورد نیاز فراخوانی نمائید. همچنین باید دقت داشت که این

دستور بر روی پروژه جدید اضافه شده باید اجرا گردد.

```
using System;
using Microsoft.AspNet.SignalR;
using Microsoft.AspNet.SignalR.Hubs;
using Microsoft.Owin.Hosting;
using Owin;

namespace SignalR02.ConsoleHost
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            app.MapHubs(new HubConfiguration { EnableCrossDomain = true });
        }
    }

    [HubName("chat")]
    public class ChatHub : Hub
    {
        public void SendMessage(string message)
        {
            var msg = string.Format("{0}:{1}", Context.ConnectionId, message);
            Clients.All.hello(msg);
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            using (WebApplication.Start<Startup>("http://localhost:1073/"))
            {
                Console.WriteLine("Press a key to terminate the server...");
                Console.Read();
            }
        }
    }
}
```

سپس یک کلاس Startup را با امضایی که مشاهده می‌کنید تهیه نمایید. در اینجا مسیریابی و تنظیمات دسترسی از سایر دومین‌ها مشخص شده‌اند. در ادامه یک Hub نمونه، تعریف و نهایتاً توسط WebApplication.Start، این وب سرور راه اندازی می‌شود. اکنون اگر برنامه را اجرا کرده و به مسیر <http://localhost:1073/signalr/hubs> مراجعه کنید، فایل پروکسی تعاریف متادیتای مرتبط با سرور قابل مشاهده خواهد بود. سمت کلاینت استفاده از آن هیچ تفاوتی نمی‌کند و با جزئیات آن پیشتر آشنا شده‌اید؛ برای مثال در کلاینت جی‌کوئری خاصیت connection.hub.url باید به مسیر جدید سرور هاب تنظیم گردد تا اتصالات به درستی برقرار شوند.

دریافت پروژه کامل مرتبط با این 4 قسمت (البته بدون فایل‌های باینری آن، جهت کاهش حجم 32 مگابایتی)

[SignalRSamples.zip](#)