عنوان: **افزودن خودکار کلاس های WebAPI و SignalR Hub به برنامه در حالت SelfHost**
نویسنده: ابوالفضل رجب پور
تاریخ: ۱۳۹۳/۰۸/۰۵ ۱۳:۵۵
آدرس: www.dotnettips.info
گروه‌ها: SignalR, ASP.NET Web API, Self-Hosting, ASP.NET Web API 2

یکی از گزینه‌های میزبانی WebAPI و SignalR حالت SelfHost می‌باشد که روش آن قبلا در مطلب « نگاهی به گزینه‌های مختلف مهیای جهت میزبانی SignalR » توضیح داده شده است.

ابتدا نگاه کوچکی به یک مثال داشته باشیم:
هاب زیر را در نظر بگیرید.

```
public class MessageHub : Hub
    {
        public void NotifyAllClients()
        {
            Clients.All.Notify();
        }
}
```

برای selfHsot کردن از یک برنامه‌ی کنسول استفاده می‌کنیم:

```
static void Main(string[] args)
        {
            const string baseAddress = "http://localhost:9000/"; // "http://*:9000/";

            using (var webapp = WebApp.Start<Startup>(baseAddress))
            {
                Console.WriteLine("Start app...");

                var hubConnection = new HubConnection(baseAddress);
                IHubProxy messageHubProxy = hubConnection.CreateHubProxy("messageHub");

                messageHubProxy.On("notify", () =>
                {
                    Console.WriteLine();
                    Console.WriteLine("Notified!");
                });

                hubConnection.Start().Wait();

                Console.WriteLine("Start signalr...");

                bool dontExit = true;
                while (dontExit)
                {
                    var key = Console.ReadKey();
                    if (key.Key == ConsoleKey.Escape) dontExit = false;

                    messageHubProxy.Invoke("NotifyAllClients");
                }

            }
        }
```

با کلاس start-up ذیل:

```
public partial class Startup
    {
        public void Configuration(IAppBuilder appBuilder)
        {
            var hubConfiguration = new HubConfiguration()
            {
                EnableDetailedErrors = true
            };

            appBuilder.MapSignalR(hubConfiguration);

            appBuilder.UseCors(CorsOptions.AllowAll);
```

```
        }
    }
```

اکنون اگر برنامه را اجرا کنیم، با زدن هر کلید در کنسول، یک پیغام چاپ می‌شود که نشان دهنده صحت کارکرد هاب پیام می‌باشد.

خوب؛ تا الان همه چیز درست کار میکند.

صورت مساله:

معمولا برای منظم کردن و مدیریت بهتر کدهای نرم افزار، آن‌ها را در پروژه‌های مجزا یا در واقع همان class libraryهای مجزا نگاه داری میکنیم.

اکنون در برنامه‌ی فوق ، اگر کلاس messageHub را به یک class library دیگر منتقل کنیم و آن را به برنامه‌ی کنسول ارجاع دهیم و برنامه را مجدد اجرا کنیم، با خطای زیر مواجه می‌شویم:

```
{"StatusCode: 500, ReasonPhrase: 'Internal Server Error', Version: 1.1, Content:
System.Net.Http.StreamContent, Headers:\r\n{\r\n  Date: Mon, 27 Oct 2014 09:36:48 GMT\r\n  Server:
Microsoft-HTTPAPI/2.0\r\n  Content-Length: 0\r\n}"}
```

مشکل چیست؟

همانطور که در مطلب « نگاهی به گزینه‌های مختلف مهیای جهت میزبانی SignalR » عنوان شده‌است، «در حالت SelfHost بر خلاف روش asp.net hosting ، اسمبلی‌های ارجاعی برنامه اسکن نمی‌شوند» و طبیعتا مشکل رخ داده شده در بالا از اینجا ناشی می‌شود.

راه حل:

- این کار باید به صورت دستی انجام پذیرد. با افزودن کد زیر به ابتدای برنامه (قبل از شروع هر کدی) اسمبلی‌های مورد نظر افزوده می‌شوند:

```
AppDomain.CurrentDomain.Load(typeof(MessageHub).Assembly.FullName);
```

طبیعتا افزودن دستی هر اسمبلی مشکل و در خیلی مواقع ممکن است با خطای انسانی فراموش کردن مواجه شود!

کد خودکار زیر، میتواند تکمیل کننده‌ی راه حل بالا باشد:

```
class LoadAssemblyHelper
    {
        public static void Load(string searchPattern)
        {
            var path = Assembly.GetExecutingAssembly().Location;
            var entityAssemblies = Directory.GetFiles(Path.GetDirectoryName(path), searchPattern:
searchPattern);
            var assemblyNames = entityAssemblies.Select(e => AssemblyName.GetAssemblyName(e)).ToList();
            assemblyNames.ToList().ForEach(e => AppDomain.CurrentDomain.Load(e));
        }
    }
```

و برای فراخوانی آن در ابتدای برنامه می‌نویسیم:

```
static void Main(string[] args)
        {
            //AppDomain.CurrentDomain.Load(typeof(MessageHub).Assembly.FullName);
            //AppDomain.CurrentDomain.Load(typeof(MessageController).Assembly.FullName);

            LoadAssemblyHelper.Load("myFramework.*.dll");

            const string baseAddress = "http://*:9000/";
            using (var webapp = WebApp.Start<Startup>(baseAddress))
            {
                ...
            }
        }
```

نکته‌ی مهم

این خطا و راه حل آن، در مورد hubهای signalr و هم controllerهای webapi صادق می‌باشد.