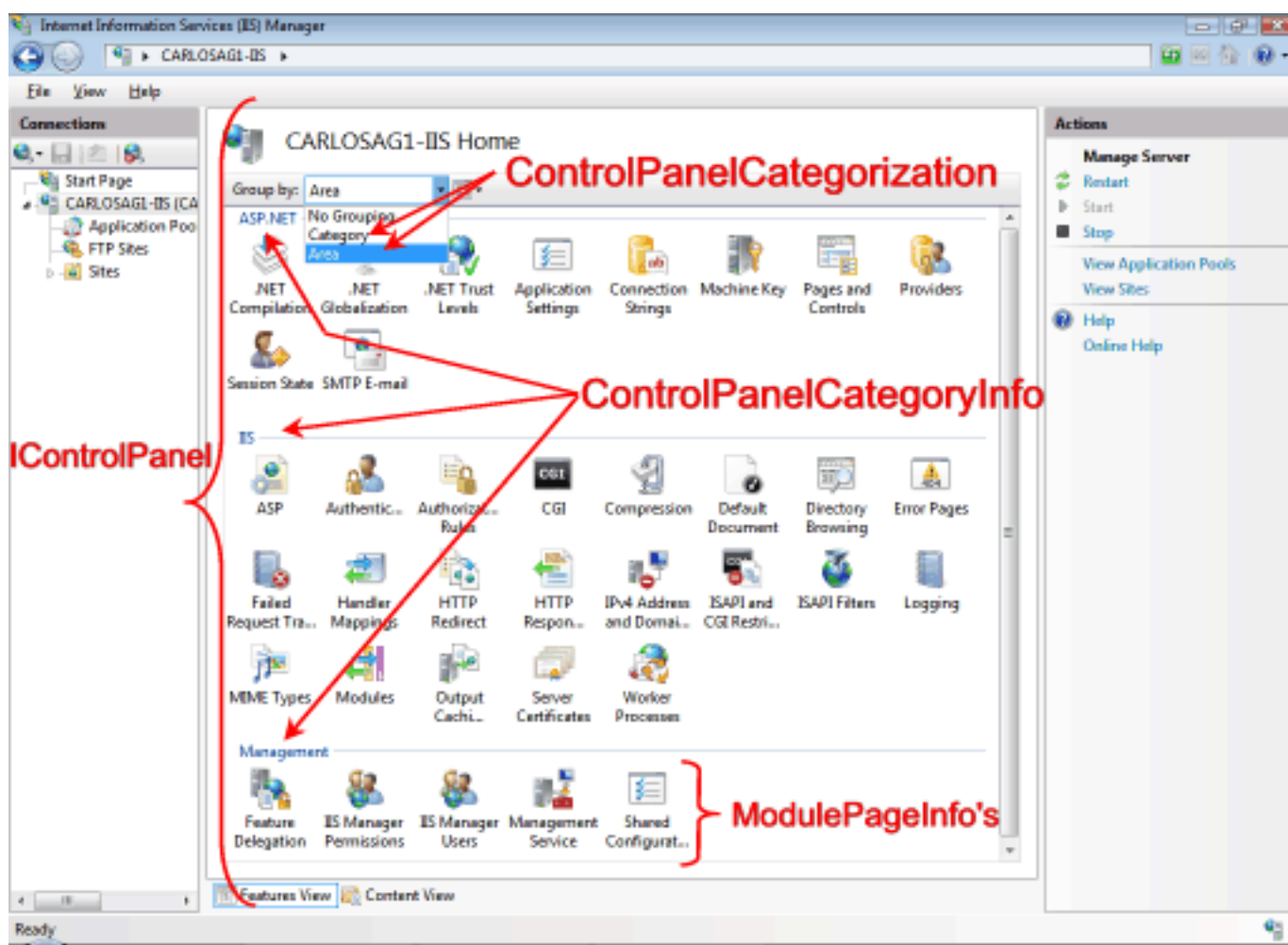


در دو مقاله پیشین ([۱](#) و [۲](#)) در مورد اینکه چگونه یک httpmodule یا httpHandler را به عنوان یک ماژول جدید به IIS اضافه کنیم صحبت کردیم و الان قصد داریم که در این بخش این مبحث را ببندیم. آخرین بار توانستیم که یک UI را به IIS نسبت داده و از آن استفاده کنیم و الان قصد داریم آن را تکمیل‌تر کرده و کمی آن را شکل‌تر کنیم. اولین نکته‌ای که توجه ما را جلب میکند این است که ماژول ما یک آیکن پیش فرض (چرخ دنده) دارد که بهتر است به آیکن دلخواه ما تغییر کند. به این منظور در پروژه، یک فایل Resource ایجاد کنید و یک تصویر را به این فایل Resource اضافه کنید و در کلاس imageCopyrightUI کد را به صورت زیر تغییر دهید:

```
internal class imageCopyrightUI : Module
{
    protected override void Initialize(IServiceProvider serviceProvider, ModuleInfo moduleInfo)
    {
        base.Initialize(serviceProvider, moduleInfo);
        IControlPanel controlPanel = (IControlPanel)GetService(typeof(IControlPanel));
        ModulePageInfo modulePageInfo = new ModulePageInfo(this, typeof(imageCopyrightUIPage),
        "Image Copyright", "Image Copyright", Resource1.Visual_Studio_2012, Resource1.Visual_Studio_2012);
        controlPanel.RegisterPage(modulePageInfo);
    }
}
```

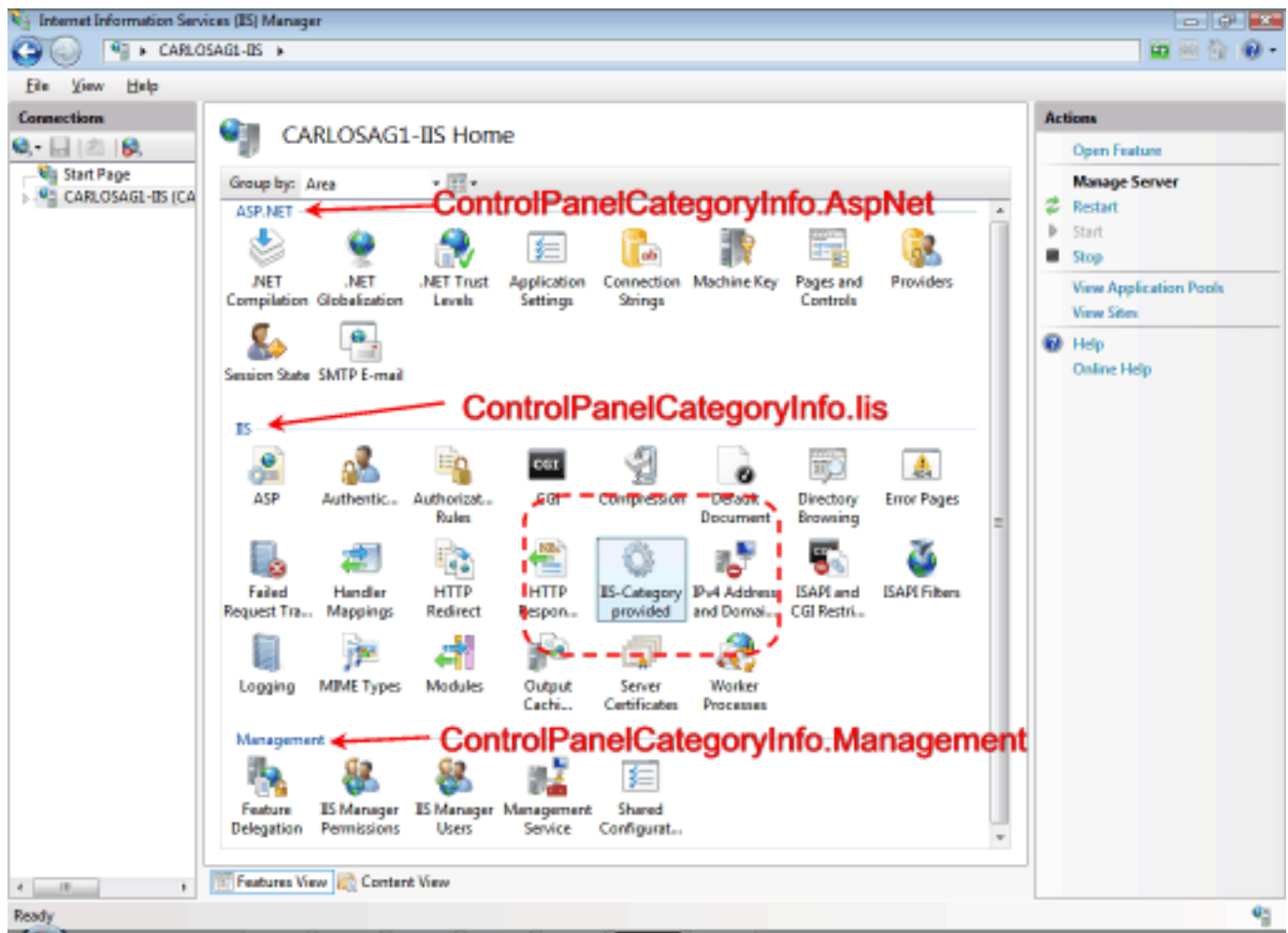
حال پروژه را Rebuild کنید و IIS را مجدداً باز کنید تا ببینید که کامپوننت جدید شما آیکن جدید را دارد. دومین نکته‌ای که به چشم می‌آید این است که اگر دقت کنید سایر ماژول‌ها یا کامپوننت‌ها هر کدام در گروهی جداگانه جای گرفته‌اند و افزونه‌ی شما در یک گروه به اسم others، که زیاد جالب نیست و شاید دوست داشته باشید که ماژول شما هم در یکی از همین دسته‌ها جای بگیرد یا حتی اینکه خودتان یک گروه جدید بسازید. برای اینکه خوب قسمت‌ها و نام کلاس‌های آن‌ها را یاد بگیرید به شکلی که در قسمت قبلی هم گذاشته بودیم خوب دقت کنید.



نکته ای که باید توجه داشته باشید این است که گروه بندی در IIS به سه شیوه صورت می‌گیرد: یکی به شیوه ناحیه **Area** مثل **ASP.Net** و **management** و **IIS**، دسته بندی **Category** مثل **Application Development**، **ServerFeatures** و بدون گروه بندی که می‌توانید از طریق لیست **Group By** در بالای پنجره IIS یکی از این سه حالت را انتخاب نمایید برای اینکه ماژول در یک ناحیه یا دسته‌ای مشخص قرار بگیرد کد زیر را بنویسید:

```
internal class imageCopyrightUI : Module
{
    protected override void Initialize(IServiceProvider serviceProvider, ModuleInfo moduleInfo)
    {
        base.Initialize(serviceProvider, moduleInfo);
        IControlPanel controlPanel = (IControlPanel)GetService(typeof(IControlPanel));
        ModulePageInfo modulePageInfo = new ModulePageInfo(this, typeof(imageCopyrightUIPage),
        "Image Copyright", "Image Copyright", Resource1.Visual_Studio_2012, Resource1.Visual_Studio_2012);
        controlPanel.RegisterPage(ControlPanelCategoryInfo.AspNet, modulePageInfo);
    }
}
```

اگر خوب دقت کنید می‌بینید که تنها تغییر کدها در متد **regiterpage** بوده است که با استفاده از کلاس **ControlPanelCategoryInfo** ناحیه مورد نظر را مشخص کردیم و اگر حالا پروژه را **Rebuild** کنید می‌بینید که در ناحیه **ASP.Net** قرار گرفته است. گروه‌های مختلفی دیگری چون **Application Development**، **Server**، **Performance** و ... هم در این کلاس وجود دارند که جمعا 9 گروه می‌شوند.



حال این سوال پیش می‌آید اگر بخواهیم گروه اختصاصی ایجاد کنیم، چه کاری باید انجام شود. پس کد را به شکل زیر تغییر دهید:

```
internal class imageCopyrightUI : Module
{
    protected override void Initialize(IServiceProvider serviceProvider, ModuleInfo moduleInfo)
    {
        base.Initialize(serviceProvider, moduleInfo);
        IControlPanel controlPanel = (IControlPanel)GetService(typeof(IControlPanel));
        ModulePageInfo modulePageInfo = new ModulePageInfo(this, typeof(imageCopyrightUIPage),
        "Image Copyright", "Image Copyright", Resource1.Visual_Studio_2012, Resource1.Visual_Studio_2012);

        ControlPanelCategorization areaCategorization = null;
        foreach (ControlPanelCategorization categorization in controlPanel.Categorizations) {
            if (categorization.Key == ControlPanelCategorization.AreaCategorization) {
                areaCategorization = categorization;
                break;
            }
        }
        ControlPanelCategoryInfo cat=new ControlPanelCategoryInfo("dotnettips","Dot Net Tips","This
is a Tutorial",areaCategorization);
        controlPanel.RegisterCategory(cat);
        controlPanel.RegisterPage(cat.Name,modulePageInfo);
    }
}
```

در خطوط foreach ما به دنبال نوع گروهی که قرار است ماژول ما در آن قرار بگیرد می‌گردیم و علاقمندیم که گروه بندی ما در نوع Area باشد. برای همین این نوع را یافته و در متغیری از نوع ControlPanelCategorization قرار می‌دهیم. سپس توسط کلاس ControlPanelCategoryInfo یک نوع گروه بندی جدید ایجاد کرده ایم که به ترتیب نام، عنوان، توضیح و نهایتاً نوع گروه بندی را در آن لحاظ می‌کنیم و سپس با دستور controlPanel.RegisterCategory گروه جدید خود را که در area قرار دارد، در IIS ثبت می‌کنیم و موقع افزودن صفحه مستقیماً نام گروه را در آن ذکر می‌کنیم. پروژه را Rebuild و IIS را مجدداً اجرا کنید. اگر Group by

شما روی Area باشد که به طور پیش فرض چنین است شما باید گروه Dot Net Tips را ببینید؛ حال از طریق لیست Group By گزینه‌ی Category را انتخاب نمایید. همانطور که مشاهده می‌کنید دوباره ماژول شما در دسته‌ی Others قرار می‌گیرد؛ چرا که ما برای Area گروه بندی را لحاظ کرده بودیم. برای اینکه بتوانیم در دو حالت، دسته بندی داشته باشیم، کد را به شکل زیر تغییر می‌دهیم:

```
ControlPanelCategorization areaCategorization = null;
ControlPanelCategorization CategoryCategorization = null;
foreach (ControlPanelCategorization categorization in controlPanel.Categorizations) {
    if (categorization.Key == ControlPanelCategorization.AreaCategorization) {
        areaCategorization = categorization;
    }
    if (categorization.Key == ControlPanelCategorization.CategoryCategorization)
    {
        CategoryCategorization = categorization;
    }
}
ControlPanelCategoryInfo cat1=new ControlPanelCategoryInfo("dotnettipsarea","Dot Net Tips
Area","This is a Tutorial",areaCategorization);
controlPanel.RegisterCategory(cat1);
controlPanel.RegisterPage(cat1.Name,modulePageInfo);

ControlPanelCategoryInfo cat2 = new ControlPanelCategoryInfo("dotnettipsat", "Dot Net Tips
Category", "This is a Tutorial", CategoryCategorization);
controlPanel.RegisterCategory(cat2);
controlPanel.RegisterPage(cat2.Name, modulePageInfo);
```

فکر کنم اتفاق بالا با توجه به مواردی که قبلا یاد گرفتید با وضوح کامل مشخص باشد که اینبار دو حالت گروه بندی را ذخیره و هر کدام را جداگانه در کنترل پنل IIS ثبت کردیم.

بین سایر گزینه‌های کنترل پنل گزینه controlpanel.regiterhomepage هم به چشم می‌خورد که اگر رابط کاربری را با این گزینه رجیستر کنیم صفحه اصلی IIS پریده و رابط کاربری ما جایگزینش می‌شود که البته این قسمت را باید با احتیاط با آن برخورد کرد وگرنه اگر بخواهیم همین رابط کاربری را به عنوان صفحه‌ی خانگی رجیستر کنیم، دسترسی ما به دیگر ماژول‌ها قطع خواهد شد.

[سورس پروژه](#)

تا به اینجا این مبحث از سری آموزشی ما بسته می‌شود. در مقالات آینده موارد دیگری از IIS را مورد بررسی قرار خواهیم داد.