عنوان: **تزریق خودکار وابستگی‌ها در برنامه‌های ASP.NET Web forms**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۲۷ ۱۷:۴
آدرس: www.dotnettips.info
گروه‌ها: Design patterns, Dependency Injection, IoC

همانطور که در قسمت‌های قبل عنوان شد، دو نوع متداول تزریق وابستگی‌ها وجود دارند:
الف) تزریق وابستگی‌ها در سازنده کلاس
ب) تزریق وابستگی‌ها در خواص عمومی کلاس‌ها یا Setters injection

حالت الف متداول‌ترین است و بیشتر زمانی کاربرد دارد که کار وهله سازی یک کلاس را می‌توان راسا انجام داد. اما در فرم‌ها یا یوزرکنترل‌های ASP.NET Web forms به صورت پیش فرض کار وهله سازی فرم‌ها و یوزرکنترل‌ها توسط موتور ASP.NET انجام می‌شود و در این حالت اگر بخواهیم از تزریق وابستگی‌ها استفاده کنیم، مدام به همان روش معروف Service locator و استفاده از container.Resolve در تمام قسمت‌های برنامه می‌رسیم که آنچنان روش مطلوبی نیست.
اما ... در ASP.NET Web forms می‌توان وهله سازی فرم‌ها را نیز تحت کنترل قرار داد، که برای آن دو روش زیر وجود دارند:
الف) یک کلاس مشتق شده را از کلاس پایه PageHandlerFactory تهیه کنیم. این کلاس را پیاده سازی کرده و نهایتا بجای وهله ساز پیش فرض فرم‌های موتور داخلی ASP.NET، در فایل وب کانفیگ برنامه استفاده کنیم. یک نمونه از پیاده سازی آن‌را در اینجا می‌توانید مشاهده کنید.
مشکلی که این روش دارد سازگاری آن با حالت Full trust است. یعنی برنامه شما در یک هاست Medium trust (اغلب هاست‌های خوب) اجرا نخواهد شد.
ب) روش دوم، استفاده از یک Http Module است برای اعمال Setter injectionها، به صورت خودکار. اکنون که حالت الف را همه جا نمی‌توان بکار برد یا به عبارتی نمی‌توان وهله سازی فرم‌ها را راسا در دست گرفت، حداقل می‌توان خواص عمومی اشیاء صفحه تولید شده را مقدار دهی کرد که در ادامه، این روش را بررسی می‌کنیم.

تهیه ماژول انجام Setters injection به صورت خودکار در برنامه‌های ASP.NET Web forms به کمک StructureMap

پیشنیاز این بحث، مطلب «استفاده از StructureMap به عنوان یک IoC Container» می‌باشد که پیشتر مطالعه کردید (در حد نحوه نصب StructureMap و آشنایی با تنظیمات اولیه آن)

```
using System.Collections;
using System.Web;
using System.Web.UI;
using StructureMap;

namespace DI05.Core
{
    /// <summary>
    /// تسهیل در کار تزریق خودکار وابستگی‌ها در سطح فرم‌ها و یوزرکنترل‌ها
    /// </summary>
    public class StructureMapModule : IHttpModule
    {
        public void Dispose()
        { }

        public void Init(HttpApplication app)
        {
            app.PreRequestHandlerExecute += (sender, e) =>
            {
                var page = HttpContext.Current.Handler as Page; // The Page handler
                if (page == null)
                    return;

                WireUpThePage(page);
                WireUpAllUserControls(page);
            };
        }

        private static void WireUpAllUserControls(Page page)
        {
            // در اینجا هم کار سیم کشی یوزر کنترل‌ها انجام می‌شود
            page.InitComplete += (initSender, evt) =>
            {
```

```
                var thisPage = (Page)initSender;
                foreach (Control ctrl in getControlTree(thisPage))
                {
                    // فقط یوزر کنترل‌ها بررسی شدند
                    // اگر نیاز است سایر کنترل‌های قرار گرفته روی فرم هم بررسی شوند شرط را حذف کنید
                    if (ctrl is UserControl)
                    {
                        ObjectFactory.BuildUp(ctrl);
                    }
                }
            };
        }

        private static void WireUpThePage(Page page)
        {
            ObjectFactory.BuildUp(page); // برقراری خودکار سیم کشی‌ها در سطح صفحات
        }

        private static IEnumerable getControlTree(Control root)
        {
            foreach (Control child in root.Controls)
            {
                yield return child;
                foreach (Control ctrl in getControlTree(child))
                {
                    yield return ctrl;
                }
            }
        }
    }
}
```

در این ماژول، کار با HttpContext.Current.Handler شروع می‌شود که دقیقا معادل با وهله‌ای از یک صفحه یا فرم می‌باشد. اکنون که این وهله را داریم، فقط کافی است متد ObjectFactory.BuildUp مربوط به StructureMap را روی آن فراخوانی کنیم تا کار Setter injection را انجام دهد. مرحله بعد یافتن یوزر کنترل‌های احتمالی قرار گرفته بر روی صفحه و همچنین فراخوانی متد ObjectFactory.BuildUp، بر روی آن‌ها می‌باشد.
پس از تهیه ماژول فوق، باید آن‌را در فایل وب کانفیگ برنامه معرفی کرد:

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />

    <httpModules>
      <add name="StructureMapModule" type="DI05.Core.StructureMapModule"/>
    </httpModules>
  </system.web>

  <system.webServer>
    <modules runAllManagedModulesForAllRequests="true">
      <add name="StructureMapModule" type="DI05.Core.StructureMapModule"/>
    </modules>
    <validation validateIntegratedModeConfiguration="false" />
  </system.webServer>
</configuration>
```

مثالی از نحوه استفاده از StructureMapModule تهیه شده

فرض کنید لایه سرویس برنامه دارای اینترفیس‌ها و کلاس‌های زیر است:

```
namespace DI05.Services
{
    public interface IUsersService
    {
        string GetUserEmail(int id);
    }
}


namespace DI05.Services
{
```

```
    public class UsersService: IUsersService
    {
        public string GetUserEmail(int id)
        {
            //فقط جهت بررسی تزریق وابستگی‌ها
            return "test@test.com";
        }
    }
}
```

کار تنظیمات اولیه آن‌ها را در فایل global.asax.cs برنامه انجام خواهیم داد:

```
using System;
using StructureMap;
using DI05.Services;

namespace DI05
{
    public class Global : System.Web.HttpApplication
    {
        private static void initStructureMap()
        {
            ObjectFactory.Initialize(x =>
            {
                x.For<IUsersService>().Use<UsersService>();

                x.SetAllProperties(y =>
                {
                    y.OfType<IUsersService>();
                });
            });
        }

        protected void Application_Start(object sender, EventArgs e)
        {
            initStructureMap();
        }

        void Application_EndRequest(object sender, EventArgs e)
        {
            ObjectFactory.ReleaseAndDisposeAllHttpScopedObjects();
        }
    }
}
```

در اینجا فقط باید دقت داشت که ذکر SetAllProperties الزامی است. از این جهت که از روش Setter injection در حال استفاده هستیم.

مرحله آخر هم استفاده از سرویس‌های برنامه به شکل زیر است:

```
using System;
using DI05.Services;

namespace DI05
{
    public partial class Default : System.Web.UI.Page
    {
        public IUsersService UsersService { set; get; }

        protected void Page_Load(object sender, EventArgs e)
        {
            lblEmail1.Text = string.Format("From Default Page: {0}", UsersService.GetUserEmail(1));
        }
    }
}
```

همانطور که ملاحظه می‌کنید در این فرم، هیچ خبری از وجود IoC Container مورد استفاده نیست و کار وهله سازی و مقدار دهی سرویس مورد استفاده به صورت خودکار توسط Http Module تهیه شده انجام می‌شود.

دریافت مثال کامل قسمت جاری

DI05.zip

یک نکته‌ی تکمیلی

برای ارتقاء نکات مطلب جاری به نگارش سوم StructureMap نیاز است موارد ذیل را لحاظ کنید:

الف) نصب بسته‌ی وب آن

```
PM> Install-Package structuremap.web
```

ب) ReleaseAndDisposeAllHttpScopedObjects حذف شده را به متد جدید HttpContextLifecycle.DisposeAndClearAll() تغییر دهید.

ج) x.SetAllProperties را به x.Policies.SetAllProperties ویرایش کنید.

نظرات خوانندگان

نویسنده: فرهاد یزدان پناه
تاریخ: ۱۳۹۲/۰۱/۲۷ ۲۲:۴۱

وقت بخیر مهندس نصیری. من شخصا بیشتر کلاس‌های اصلی مربوط به ASP.NET رو سفارشی کردم (Page، UserControl، HttpHandler و ...) و در سازنده عملیات‌های مربوط به سیم کشی! رو انجام دادم. (قبلا در مباحث مربوط به Entity Framework این روش رو توضیح داده بودید) ولی به نظرم این روش HttpModule خیلی منظمتره.
مسئله ای که به نظر می‌تونه کمک کنه اینه که کاش فقط Pageها رو سیم کشی نمی‌کردید (همه چیز حتی HttpHandlerها هم میشه همینجا کلکشون کنده بشه و همچنین WebControlها)
در هر حال دستتون درد نکنه.

نویسنده: فرهاد یزدان پناه
تاریخ: ۱۳۹۲/۰۲/۰۸ ۲۱:۲۷

وقت بخیر در حالاتی که کنترل هایی دارای کالکشن هایی از کنترل‌های دیگر باشند (مثل GridView که دارای ستون هایی است - هرچند که ستون کنترل نیست) این موارد سیمکشی نمی‌شوند چون جزو درخت Page نیستند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۸ ۲۱:۵۹

شما داخل کنترل‌های قرار گرفته داخل GridView نیاز به تزریق وابستگی‌ها دارید؟
مثلا یک ستون آن دارای سلول‌هایی از جنس یوزر کنترل است که داخل این نیاز هست تزریق صورت گیرد؟

نویسنده: فرهاد یزدان پناه
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۸:۵۹

تقریبا!
من یک کنترل آکاردیون دارم که دارای ساختاری شبه زیر است:

```
<Common:HomeAccordion runat="server">
    <Common:HomeAccordionPane runat="server" HeaderText="Pane #1">
        <Common:HomeAccordionItem runat="server" Text="Item #1"/>
        <Common:HomeAccordionItem runat="server" Text="Item #1"/>
    </Common:HomeAccordionPane>
    <Common:HomeAccordionPane runat="server" HeaderText="Pane #2">
        <Common:HomeAccordionItem runat="server" Text="Item #1"/>
        <Common:HomeAccordionItem runat="server" Text="Item #1"/>
    </Common:HomeAccordionPane>
</Common:HomeAccordion>
```

حال گاهی من از کنترل اصلی AccordionPane ارث برده و کنترل هایی را ایجاد می‌کنم که به صورت خودکار از سرویس‌های امنیتی استفاده کرده و آیتم‌های لازم (که کاربر جاری به آنها دسترسی دارد) را اضافه می‌کنم.

```
<Common:HomeAccordion runat="server">
    <DF:HomeAccordionPaneBasic runat="server" />
    <DF:HomeAccordionPaneSystem runat="server" />
    <DF:HomeAccordionPaneMyAccount runat="server" />
</Common:HomeAccordion>
```

گویا این اقلام (که معمولا می‌توانند کنترل هم نباشند - چون توسط والد رندر می‌شوند) جزو درخت صفحه نیستند - استفاده از ParseChildren در کنترل والد اجازه افزودن مجموعه کنترل هایی را می‌دهد)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۹:۱۹

- نیاز به مثال کامل برای دیباگ هست.
- در کل اگر از روش مطرح شده در مطلب جاری جواب نگرفتید یا کمبود داشت، در سازنده کلاس، متد زیر را فراخوانی کنید:

```
ObjectFactory.BuildUp(this);
```

نویسنده: فرهاد یزدان پناه
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۹:۲۴

ممنون.
طبق چیزی که قبلا خودتون فرموده بودید (در دوره EF) من هم از همین روش استفاده می‌کنم.
فقط کلاس‌های Page، UserContrl، و چند کلاس دیگر را (که به عنوان کلاس‌های من قرار دارند) در سازنده انجام دادم.
فقط برای HttpModule (چون همانند شی Application دارای یک نمونه است) در متد Init کارهای لازم رو انجام دادم.

نویسنده: فرهاد یزدان پناه
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۹:۲۶

سعی می‌کنم یک مثال که موارد لازم در اون دخیل باشه رو براتون ارسال کنم (در همین پست به عنوان پیوست)

نویسنده: vici
تاریخ: ۱۳۹۲/۱۱/۰۷ ۲۲:۲۸

سلام؛ تا قبل از این قسمت رو خوب متوجه شدم ولی توی این قسمت کلاس StructureMapModule اصلا متوجه نشدم چیه؟ یه مقدار راهنمایی می‌کنید؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۷ ۲۲:۴۹

در مطلب « بایدها و نبایدهای استفاده از IoC Containers » عنوان شد که تا حد ممکن نباید کدهای مرتبط با یک IoC Container داخل کدهای متداول ما ظاهر شوند. کار کلاس StructureMapModule تمیز کردن فایل‌های code behind از وجود IoC Container مورد نظر است.

نویسنده: vici
تاریخ: ۱۳۹۲/۱۱/۰۷ ۲۳:۱

ممنون ، یعنی این کلاس ثابت هست؟ در صورتی که نیاز به ویژگی یا تنظیم دیگری نداشته باشیم همین کلاس کفایت می‌کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۷ ۲۳:۵

کار این کلاس، در مثالی که زده شد (انتهای مطلب) این است که UsersService درخواستی را به صورت خودکار وهله سازی و قابل استفاده می‌کند؛ بدون اینکه جایی اثری از StructureMap در این فایل code behind دیده شود.

نویسنده: فواد کریمی
تاریخ: ۱۳۹۲/۱۲/۲۱ ۱۸:۳۸

با سلام؛ من در ویندوز اپلیکیشن ازاین ساختار استفاده میکنم و از فرم‌های Devexpress استفاده میکنم .
در کلاس BasePage روی دستور ObjectFactory this خطای زیر رو میده

```
An unhandled exception of type 'StructureMap.StructureMapException' occurred in StructureMap.dll
Additional information: Error in the application.
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۲ ۰:۳۹

این خطای کلی بدون مشخص بودن کدهای شما قابل بررسی نیست. اطلاعات بیشتر
معمولا در VS.NET اگر بر روی جزئیات بیشتر استثنای رخداده کلیک کنید، مقادیر تو در توی آن مانند inner exception حاصل، اطلاعات بیشتری را به همراه دارند.

نویسنده: ابوالفضل علیاری
تاریخ: ۱۳۹۳/۰۱/۰۶ ۱۲:۲۸

برای کار با ef و استفاده از UOW ،لایه سرویس باید از روشی که در EF#12 آموزش دادید نوشته بشود؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۶ ۱۲:۵۶

مطلب و دوره جاری، پیشنیازی است برای درک جزئیات مطلبی که به آن اشاره کردید .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۱۳ ۰:۵۳

به روز سانی
روش ارتقاء به نگارش سوم StructureMap به انتهای بحث اضافه شد.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۲/۲۵ ۲۲:۳۲

ممنون، ظاهراً با MVC5 سازگار نیست، ربطش رو نمی‌دونم ولی با MVC5 تست کردم مشکل داشت (از مقدار بازگشتی توسط متد GetControllerInstance اشکال می‌گرفت) ، با تعویض لایه وب به ورژن MVC4 مشکلم حل شد. مثالی تکمیلی مربوط به قسمت 12 سری EF شما هم برای ورژن MVC4 بود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۶ ۱:۴۳

با MVC5 هم تستش کردم. مشکلی نبود. در GetControllerInstance فقط باید بررسی کنید که آیا controllerType نال هست یا خیر. اگر نال بود، یعنی یک آدرس یافت نشد در برنامه دارید:

```
if (controllerType == null && requestContext.HttpContext.Request.Url != null)
                throw new InvalidOperationException(string.Format("Page not found: {0}",

requestContext.HttpContext.Request.Url.AbsoluteUri.ToString(CultureInfo.InvariantCulture)));
```