

عنوان: ایجاد ServiceLocator با استفاده از Ninject

نویسنده: مسعود پاکدل

تاریخ: ۲۰:۵ ۱۳۹۱/۱۲/۰۴

آدرس: www.dotnettips.info

برچسب‌ها: Dependency Injection, ServiceLocator, Ninject

در [پست قبلی](#) روش استفاده از ServiceLocator رو با استفاده از Microsoft Unity بررسی کردیم. در این پست قصد داریم همون مثال رو با استفاده از Ninject پیاده سازی کنیم. Ninject ابزاری برای پیاده سازی Dependency Injection در پروژه‌های دات نت است که کار کردن با اون واقعا راحت. برای شروع کلاس‌های Book و BookRepository و BookService و اینترفیس IBookRepository از این [پست](#) دریافت کنید.

حالا با استفاده از NuGet باید ServiceLocator رو برای Ninject دریافت کنید. برای این کار در Package Manager Console دستور زیر رو وارد کنید.

```
PM> Install-Package CommonServiceLocator.NinjectAdapter
```

بعد از دانلود و نصب Reference‌های زیر به پروژه اضافه می‌شوند.

Ninject

NinjectAdapter

Microsoft.Practices.ServiceLocation

اگر دقت کنید برای ایجاد ServiceLocator داریم از Enterprise Library:ServiceLocator استفاده می‌کنیم. ولی برای این کار به جای استفاده از UnityServiceLocator باید از NinjectServiceLocator استفاده کنیم.

ابتدا

برای پیاده سازی مثال قبل در کلاس Program کدهای زیر رو وارد کنید.

```
using System;
using Ninject;
using NinjectAdapter;
using Microsoft.Practices.ServiceLocation;

namespace ServiceLocatorPattern
{
    class Program
    {
        static void Main( string[] args )
        {
            IKernel kernel = new StandardKernel();

            kernel.Bind<IBookRepository>().To<BookRepository>();

            ServiceLocator.SetLocatorProvider( () => new NinjectServiceLocator( kernel ) );

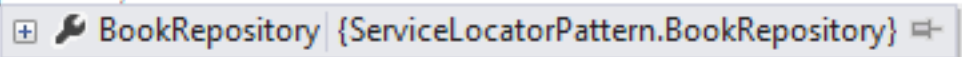
            BookService service = new BookService();

            service.PrintAllBooks();

            Console.ReadLine();
        }
    }
}
```

همان طور که می‌بینید روال انجام کار دقیقا مثل قبل هست فقط Syntax کمی متفاوت شده. برای مثال به جای استفاده از IUnityContainer از IKernel استفاده کردم و به جای دستور RegisterType از دستور Bind استفاده شده. در نهایت هم ServiceLocator به یک NinjectServiceLocator ای که IKernel رو دریافت کرده ست شد. به تصویر زیر دقت کنید.

```
public BookService()
{
    BookRepository = ServiceLocator.Current.GetInstance<IBookRepository>();
}
```

A tooltip is shown for the `BookRepository` property access. It contains a plus icon, a wrench icon, the text `BookRepository`, and a list of items in curly braces: `{ServiceLocatorPattern.BookRepository}`.

همان طور که می‌بینید در هر جای پروژه که نیاز به یک Instance از یک کلاس داشته باشید می‌تونید با استفاده از ServiceLocator این کار خیلی راحت انجام بدید.

بعد از اجرای پروژه خروجی دقیقا مانند مثال قبل خواهد بود.