

اگر با برنامه‌های وب و StructureMap کار کرده باشید، حتما از متد جدید `HttpContextLifecycle.DisposeAndClearAll` و متد قدیمی `ObjectFactory.ReleaseAndDisposeAllHttpScopedObjects` آن نیز برای `Dispose` خودکار کلیه اشیاء `IDisposable` در `Application_EndRequest` استفاده کرده‌اید. البته شرط استفاده از متدهای یاد شده نیز این است که طول عمر اشیاء `IDisposable` به صورت `Http Scoped` تعریف شده باشند:

```
x.For<IUnitOfWork>().HybridHttpOrThreadLocalScoped().Use<MyContext>();
```

سؤال: برای سایر حالات چگونه؟ در یک برنامه‌ی ویندوزی کنسول یا سرویس ویندوز که `Http Scoped` در آن معنا ندارد چکار باید کرد؟

پاسخ: در اینجا حداقل دو راه حل وجود دارد:

الف) استفاده از `nested containers`

```
using (var container = ObjectFactory.Container.GetNestedContainer())
{
    var uow = container.GetInstance<IUnitOfWork>();
}
```

قابلیتی از نگارش 2.6 استراکچرکمپ به آن اضافه شده‌است به نام `nested containers` که هدف از آن `Dispose` خودکار کلیه اشیاء `Transient` از نوع `IDisposable` است. در اینجا منظور از `Transient` این است که طول عمر شیء مدنظر به صورت `Singleton`، `HttpContext scoped` و یا `ThreadLocal scoped` تعریف نشده باشد (هیچ نوع `caching` خاصی به طول عمر آن اعمال نشده باشد). در مثال فوق، پس از پایان کار قطعه‌ی `using` نوشته شده، به صورت خودکار کلیه اشیاء `IDisposable` یافت شده و `Dispose` می‌شوند.

ب) نگاهی به پشت صحنه‌ی متد `DisposeAndClearAll`

اگر اشیاء `IDisposable` شما با طول عمر `HybridHttpOrThreadLocalScoped` معرفی شده باشند (و `Transient` نباشند)، [یا دستور ذیل](#) چه در برنامه‌های ویندوزی و چه در برنامه‌های وب، کلیه‌ی آن‌ها یافت شده و به صورت خودکار `Dispose` می‌شوند:

```
new HybridLifecycle().FindCache(null).DisposeAndClear();
```

متد `HttpContextLifecycle.DisposeAndClearAll` فقط مختص است به برنامه‌های وب. اگر نیاز به متدی دارید که در هر دو حالت برنامه‌های وب و ویندوزی کار کند، از روش `HybridLifecycle` فوق استفاده نمایید.

بنابراین به صورت خلاصه

اگر طول عمر شیء `IDisposable` مدنظر به صورت هیبرید تعریف شده‌است، از متد `DisposeAndClear` موجود در `HybridLifecycle` می‌توان استفاده کرد. اگر طول عمر شیء `IDisposable` مورد استفاده، معمولی است و هیچ نوع `caching` خاصی برای آن در نظر گرفته نشده‌است، می‌توان از روش `nested containers` برای رها سازی خودکار منابع آن کمک گرفت.