

LEVEL 2 – Interceptor Seçiciler

Kütüphaneler

- Autofac
- Autofac.Extras.DynamicProxy

Seviye 2’de interceptorları interceptor seçicilerle kullanmayı göreceğiz.Öncelikle her zamanki gibi interceptorlarımızı oluşturuyoruz.Ben örnek olarak cache ve validasyon işlemini simule edecek olan CacheAspect ve ValidationAspect isminde interceptorlar oluşturacağım.

```
public class CacheAspect : IInterceptor
{
    public void Intercept(IInvocation invocation)
    {
        Console.WriteLine("CacheAspect is run");
        invocation.Proceed();
    }
}
public class ValidationAspect : IInterceptor
{
    public void Intercept(IInvocation invocation)
    {
        Console.WriteLine("ValidationAspect is run");
        invocation.Proceed();
    }
}
```

Şimdi de interceptor seçicimizi oluşturalım.Bunun için seçme işlemini yaptırmak istediğimiz nesneye IInterceptorSelector interface’ini implement ediyoruz.Karşımıza IInterceptor referans dizisini dönen SelectInterceptors metodu çıkıyor.

```
public class InterceptorSelector : IInterceptorSelector
{
    public IInterceptor[] SelectInterceptors(Type type, MethodInfo method,
        IInterceptor[] interceptors)
    {
        interceptors = new IInterceptor[] { new CacheAspect() };
        return interceptors;
    }
}
```

Interceptorlarımızı oluşturduktan sonra bellekte referanslarını çalışma zamanında oluşturmak için ContainerBuilder nesnesine register edeceğim.Aynı şekilde servislerimizide register ediyoruz.

```
static void Main(string[] args)
{
    var container = new ContainerBuilder();
    container.RegisterType<CacheAspect>();
    container.RegisterType<ValidationAspect>();
    container.RegisterType<ProductManager>()
        .EnableClassInterceptors(new ProxyGenerationOptions
        {
            Selector=new InterceptorSelector()
        });

    var build = container.Build().BeginLifetimeScope();
    var item = build.Resolve<ProductManager>();
    item.Add();
}
```

Fakat burada interceptor kullanmak istediğimiz servisi register ederken EnableClassInterceptors() metoduna ProxyGenerationOptions nesnesini parametre vererek interceptor seçicimizi belirtiyoruz.Eğer servisimizde interceptor seçici kullanırsak Intercept attribute'ü ile tanımladığımız interceptorlar çalışmaz sadece SelectInterceptors metodunun döndürdüğü interceptorlar çalışacaktır.Yani CacheAspect interceptor nesnesi çalışacaktır

```
[Intercept(typeof(ValidationAspect))]
public class ProductManager : IProductService
{
    public virtual void Add()
    {
        Console.WriteLine("Product is added");
    }
}
```

ProductManager nesnemize interceptor seçici belirttiğimiz için nesnenin üstünde belirttiğimiz interceptor çalışmayacaktır.

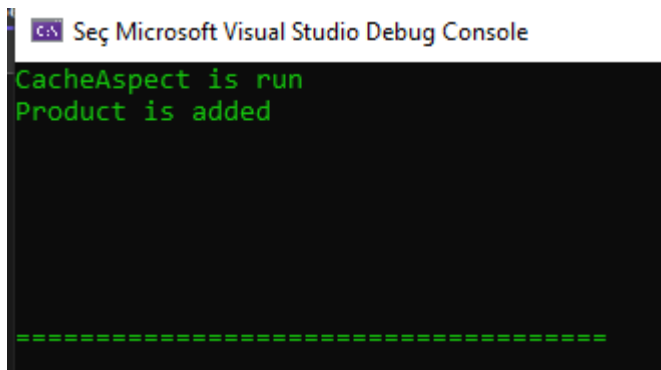
Buradaki önemli bir husustan daha bahsedelim.Interceptor seçici kullandığımız zaman seçicinin bize döndürdüğü interceptorları ayrıca ContainerBuilder nesnesine register etmemize gerek yoktur.Bunun nedeni zaten interceptor seçici bize interceptor nesnelerinin referans dizisini yani referansı oluşmuş halini döndürecek olmasıdır.

Seviye 1 de gördüğümüz manuel şekilde(Intercept attribute ile) interceptor tanımlarken ContainerBuilder nesnesine register etmemiz gerekiyordu

Yani register işlemlerinin son hali bu şekilde olacaktır

```
static void Main(string[] args)
{
    var container = new ContainerBuilder();
    container.RegisterType<ProductManager>()
        .EnableClassInterceptors(new ProxyGenerationOptions
        {
            Selector=new InterceptorSelector()
        });

    var build = container.Build().BeginLifetimeScope();
    var item = build.Resolve<ProductManager>();
    item.Add();
}
```



Seç Microsoft Visual Studio Debug Console

CacheAspect is run
Product is added

=====

Uygulamanın çıktısı yandaki gibi olacaktır.

NOT: Interceptor seçicilerinin çalışabilmesi için servisimizdeki (ProductManager) interceptor kullanılmak istenilen metodun virtual olması gereklidir.Virtual olmayan metoda interceptor uygulanmaz.Bunu ileriki seviyelerde aşmayı yani bir metodun virtual olmadığı halde interceptor yapısını nasıl çalıştırabiliriz bunu göstereceğiz.

Sonraki seviyelerde görüşmek üzere...