

همانطور که می‌دانید در صورت عدم تعریف صریح layout در یک View، این تعریف از فایل Views_ViewStart.cshtml دریافت می‌گردد:

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
```

برای معرفی صریح فایل layout، تنها کافی است مسیر کامل فایل layout را در یک View مشخص کنیم:

```
@{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Index</h2>
```

حال ما می‌خواهیم یک فیلتر سفارشی را تعریف کنیم تا به راحتی امکان تعریف Layout در سطح کنترلر و هم در سطح اکشن متد به صورت Attribute را داشته باشد :

```
[SetLayoutAttribute("_MyLayout")]
public ActionResult Index()
{
    return View();
}
```

همچنین می‌توانیم این فیلتر سفارشی را در سطح کنترلر تعریف کنیم تا تمام اکشن متدهای داخل کنترلر از Layout مربوطه استفاده کنند:

```
[SetLayoutAttribute("_MyLayout")]
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    public ActionResult About()
    {
        ViewBag.Message = "Your application description page.";
        return View();
    }

    public ActionResult Contact()
    {
        ViewBag.Message = "Your contact page.";
        return View();
    }
}
```

برای تعریف چنین اکشن فیلتری کد زیر را می‌نویسیم :

```
public class SetLayoutAttribute : ActionFilterAttribute
```

```
{
    private readonly string _masterName;
    public SetLayoutAttribute(string masterName)
    {
        _masterName = masterName;
    }

    public override void OnActionExecuted(ActionExecutedContext filterContext)
    {
        base.OnActionExecuted(filterContext);
        var result = filterContext.Result as ViewResult;
        if (result != null)
        {
            result.MasterName = _masterName;
        }
    }
}
```

همانطور که می‌دانید برای تعریف یک اکشن فیلتر سفارشی می‌بایست از کلاس `ActionFilterAttribute` ارث بری کنیم، حالا برای کلاسی که تعریف کرده ایم یک خصوصیت `readonly` را تعریف و سپس برابر با یک پارامتر با نام `masterName` که از طریق سازنده کلاس دریافت می‌شود قرار داده ایم. در نهایت متد `OnActionExecuted` را بازنویسی کرده ایم به این صورت که مقدار دریافتی توسط سازنده کلاس را برابر با نام `Layout` موردنظرمان است را به خاصیت [MasterName](#) اختصاص می‌دهیم.