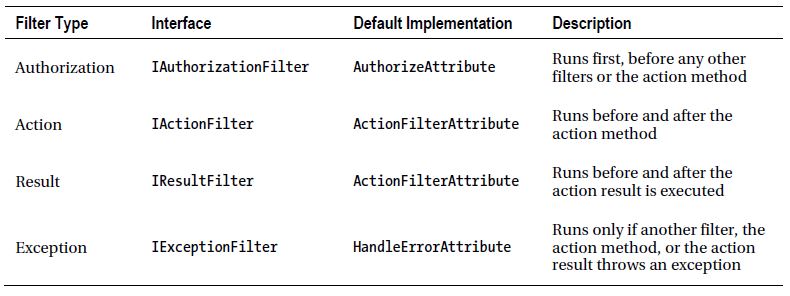
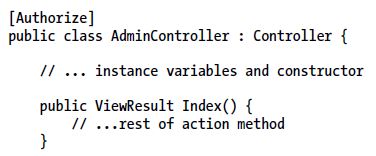
1. **Using Filters**
   1. 內建四種基本的Filter種類



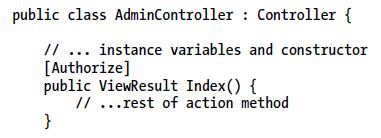
* 1. 可套用範圍
     1. 整個Controller (等於裡面的Action全部套用)

\* 如果Controller是繼承自己寫的某個class，若這個class有套用filter，那繼承的子類

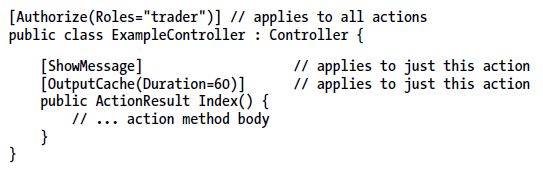
別也會一併套用



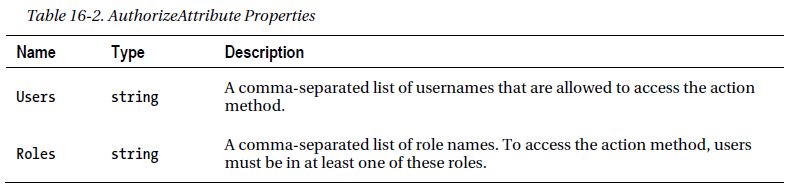
* + 1. 單一個Action Method

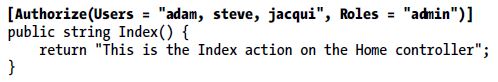


* + 1. 混合使用 (Controller與Action都使用，甚至不同的Filter也可以一起套用)

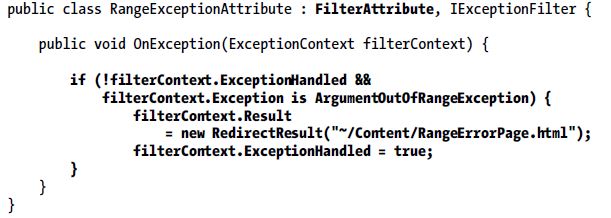


* 1. 種類
     1. **Authorization**：AuthorizeAttribute (可以設定哪個角色或使用者可以存取action，如下圖)



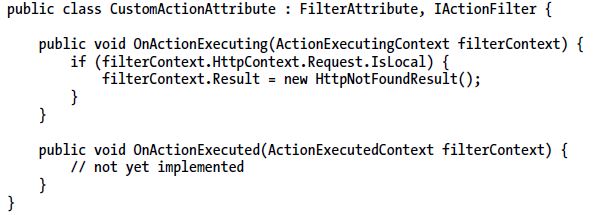


* + 1. **Exception**：HandleErrorAttribute (或是實作IExceptionFilter介面以及FilterAttribute類別)

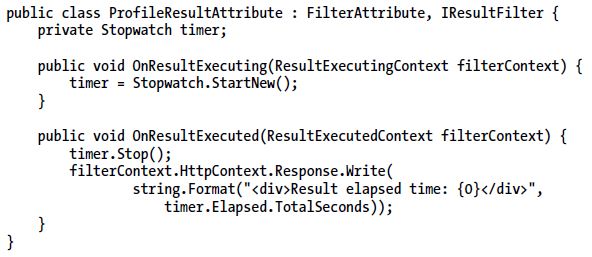


* + 1. **Action**：ActionFilterAttribute (或是實作IActionFilter介面以及FilterAttribute類別)

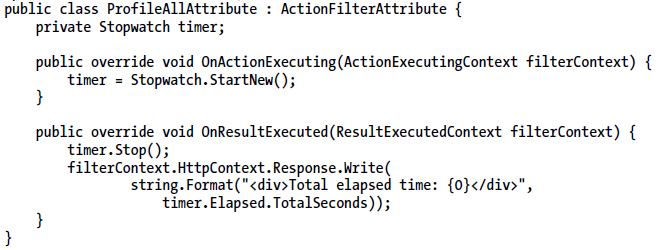
\* 最主要就是ActionExecuting以及ActionExecuted



* + 1. **Result**：ActionFilterAttribute (或是實作IResultFilter介面以及FilterAttribute類別)

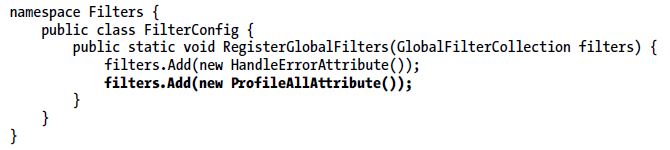


下面這個是直接繼承ActionFilterAttribute (兩種做法都可以達到目的)

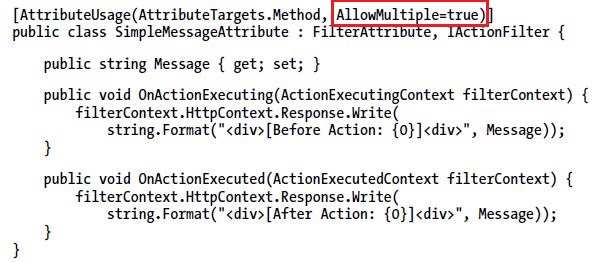


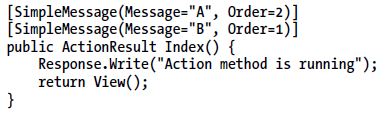
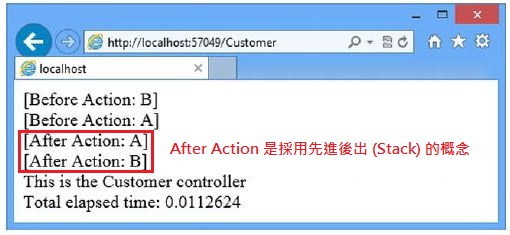
* + 1. **1**

1. **Filter其他特色**
   1. 不使用Attribute方式的Filter
      1. 直接透過Controller類別就可以做到上述的事情，因為Controller類別已經實作開頭提到的四個介面IAuthorizationFilter、IExceptionFilter、IActionFilter、IResultFilter
   2. 全域性的Filter
      1. 在FilterConfig裡面加入需要的Filter類別

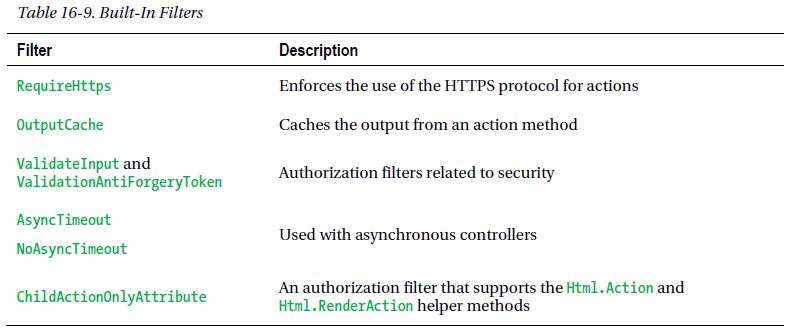


* 1. 控制Filter的執行順序
     1. 大類的Filter順序是Athorization→Action→Result，Exception在任一順序都有可能發生
     2. 單一種類的Filter可以設定執行的優先順序 (下面的範例是同一個Filter套用2次)



1. **MVC內建的Filter**

****