Part 77 - Custom action filters in asp.net mvc

Suggested Videos
Part 74 - CacheProfiles
Part 75 - RequireHttps attribute
Part 76 - ValidateInput attribute

In this video, we will discuss creating custom action filters in asp.net mvc.

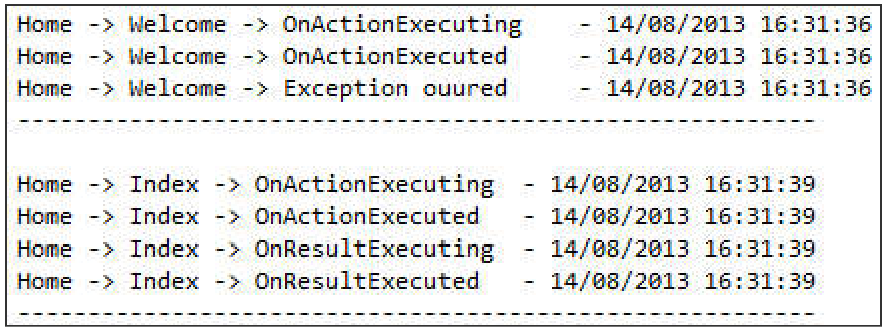
Actions are public methods in a controller. Action filters are attributes, that can be applied either on a controller or on a controller action method, which allow us to add pre and post processing logic to the action methods.

So, in simple terms an action filter allow us to execute some custom code, either, just before an action method is executed or immediately after an action method completes execution. We have discussed some of the built-in action filters in the previous sessions of this video series.
Part 70 - Authorize attribute
Part 72 - HandleError attribute
Part 73 - OutputCache attribute
Part 75 - RequireHttps attribute

Part 76 - ValidateInput attribute

Now let's discuss, creating a custom action filter. The custom action filter that we are going to build, should log the following information to a text file.
1. The name of the controller
2. The name of the action method
3. Execution time
4. If there is an exception, log the exception message and the time of the exception.

The output of the text file should be as shown below.



```
Home -> Welcome -> OnActionExecuting    - 14/08/2013 16:31:36
Home -> Welcome -> OnActionExecuted     - 14/08/2013 16:31:36
Home -> Welcome -> Exception ouured     - 14/08/2013 16:31:36
---------------------------------------------------------

Home -> Index -> OnActionExecuting  - 14/08/2013 16:31:39
Home -> Index -> OnActionExecuted   - 14/08/2013 16:31:39
Home -> Index -> OnResultExecuting  - 14/08/2013 16:31:39
Home -> Index -> OnResultExecuted   - 14/08/2013 16:31:39
---------------------------------------------------------
```

There are 4 types of filters in asp.net mvc.
1. Authorization filters - Implements IAuthorizationFilter. Examples include AuthorizeAttribute and RequireHttpsAttribute. These filters run before any other filter.
2. Action filters - Implement IActionFilter
3. Result filters - Implement IResultFilter. Examples include OutputCacheAttribute.
4. Exception filters - Implement IExceptionFilter. Examples include HandleErrorAttribute.

For detailed explanation of these attributes, please refer to the following MSDN link
http://msdn.microsoft.com/en-us/library/gg416513(v=vs.98).aspx

Step 1: Create an asp.net mvc 4 application using "Empty" template

Step 2: Right click on the project name in solution explorer and add "Data" folder. Add a text file to this folder and name it Data.txt

How to become .NET developer

Resources available to help you

Dot Net Video Tutorials

ASP.NET Core Tutorial

Angular 6 Tutorial

Angular CRUD Tutorial

Angular CLI Tutorial

Angular 2 Tutorial

Design Patterns

SOLID Principles

ASP.NET Web API

Bootstrap

AngularJS Tutorial

jQuery Tutorial

JavaScript with ASP.NET Tutorial

JavaScript Tutorial

Charts Tutorial

LINQ

LINQ to SQL

LINQ to XML

Entity Framework

Step 3: Right click on the project name in solution explorer and add "Common" folder. Add a class file to this folder and name it "TrackExecutionTime.cs". Copy and paste the following code. Notice that our custom filter "TrackExecutionTime" inherits from ActionFilterAttribute and IExceptionFilter. ActionFilterAttribute class implements IActionFilter and IResultFilter.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.IO;

namespace MVCDemo.Common
{
    public class TrackExecutionTime : ActionFilterAttribute, IExceptionFilter
    {
        public override void OnActionExecuting(ActionExecutingContext filterContext)
        {
            string message = "\n" +
filterContext.ActionDescriptor.ControllerDescriptor.ControllerName +
                " -> " + filterContext.ActionDescriptor.ActionName + " -> OnActionExecuting
\t- " +
                DateTime.Now.ToString() + "\n";
            LogExecutionTime(message);
        }

        public override void OnActionExecuted(ActionExecutedContext filterContext)
        {
            string message = "\n" +
filterContext.ActionDescriptor.ControllerDescriptor.ControllerName +
                " -> " + filterContext.ActionDescriptor.ActionName + " -> OnActionExecuted \t-
" +
                DateTime.Now.ToString() + "\n";
            LogExecutionTime(message);
        }
        public override void OnResultExecuting(ResultExecutingContext filterContext)
        {
            string message = filterContext.RouteData.Values["controller"].ToString() +
                " -> " + filterContext.RouteData.Values["action"].ToString() +
                " -> OnResultExecuting \t- " + DateTime.Now.ToString() + "\n";
            LogExecutionTime(message);
        }

        public override void OnResultExecuted(ResultExecutedContext filterContext)
        {
            string message = filterContext.RouteData.Values["controller"].ToString() +
                " -> " + filterContext.RouteData.Values["action"].ToString() +
                " -> OnResultExecuted \t- " + DateTime.Now.ToString() + "\n";
            LogExecutionTime(message);
            LogExecutionTime("---------------------------------------------------------\n");
        }

        public void OnException(ExceptionContext filterContext)
        {
            string message = filterContext.RouteData.Values["controller"].ToString() + " -> "
+
                filterContext.RouteData.Values["action"].ToString() + " -> " +
                filterContext.Exception.Message + " \t- " + DateTime.Now.ToString() + "\n";
            LogExecutionTime(message);
            LogExecutionTime("---------------------------------------------------------\n");
        }

        private void LogExecutionTime(string message)
        {
            File.AppendAllText(HttpContext.Current.Server.MapPath("~/Data/Data.txt"),
message);
        }
    }
}
```

Step 4: Add a HomeController. Copy and paste the following code.

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Java Video Tutorials

Part 1 : Video | Text | Slides

Part 2 : Video | Text | Slides

Part 3 : Video | Text | Slides

Interview Questions

C#

SQL Server

Written Test

```
public class HomeController : Controller
{
    [TrackExecutionTime]
    public string Index()
    {
        return "Index Action Invoked";
    }

    [TrackExecutionTime]
    public string Welcome()
    {
        throw new Exception("Exception ocuured");
    }
}
```

Please Note: TrackExecutionTime class is present in MVCDemo.Common namespace.

Build the application and navigate to **/Home/Index** and then to **/Home/Welcome**. Notice that the execution times and the exception details are logged in the text file.

No comments:

Post a Comment

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Enter your comment...

Comment as: toilati123vn@g ▼

Publish Preview

Links to this post

Create a Link

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Powered by Blogger.