

Exception Filter:

```
ata > C# CustomActionFilter.cs > {} ActionFilters.Data > CustomActionFilter > CustomActionFilter(string name)
1  using System.Diagnostics;
2  using Microsoft.AspNetCore.Mvc;
3  using Microsoft.AspNetCore.Mvc.Filters;
4
5  namespace ActionFilters.Data
6  {
7      2 references
8      public class CustomActionFilter : Attribute, IActionFilter, IExceptionHandler
9      {
10         3 references
11         private readonly string _name;
12
13         1 reference
14         public CustomActionFilter()
15         {
16         }
17     }
```

```
33
34     0 references
35     public CustomActionFilter(IHostEnvironment hostEnvironment) =>
36         _hostEnvironment = hostEnvironment;
37
38     0 references
39     public void OnException(ExceptionContext context)
40     {
41         if (!_hostEnvironment.IsDevelopment())
42         {
43             // Don't display exception details unless running in Development.
44             return;
45         }
46         context.Result = new ContentResult
47         {
48             Content = context.Exception.ToString()
49         };
50     }
51 }
52 }
```

Activate Windows
Go to Settings to activate Windows.

```

C# Program.cs
1  using ActionFilters.Data;
2
3  var builder = WebApplication.CreateBuilder(args);
4
5  // Add services to the container.
6
7  builder.Services.AddControllers();
8  // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
9  builder.Services.AddEndpointsApiExplorer();
10 builder.Services.AddSwaggerGen();
11 builder.Services.AddControllers(options =>
12 {
13     options.Filters.Add(new CustomActionFilter());
14     //options.Filters.Add(new CustomExceptionHandler());
15 });
16
17 var app = builder.Build();
18
19 // Configure the HTTP request pipeline.
20 if (app.Environment.IsDevelopment())
21 {
22     //app.UseSwagger();
23     //app.UseSwaggerUI();

```

Result Filter:

```

    }
}*/
0 references
public void OnResultExecuting(ResultExecutingContext context){
    Console.WriteLine("Action Name : "+context.ActionDescriptor.DisplayName);
    Console.WriteLine("OnResultExecuting");
}
0 references
public void OnResultExecuted(ResultExecutedContext context){
    Console.WriteLine("Action Name : "+context.ActionDescriptor.DisplayName);
    Console.WriteLine("OnResultExecuted");
}
}
}

```

```

Failed to determine the https port for redirect.
Action Name : ResourceManageGroup.Controllers.HomeController.Index (ResourceManageGroup)
OnResultExecuting
Action Name : ResourceManageGroup.Controllers.HomeController.Index (ResourceManageGroup)
OnResultExecuted

```