Instructor example

Web University by Harsha Vardhan

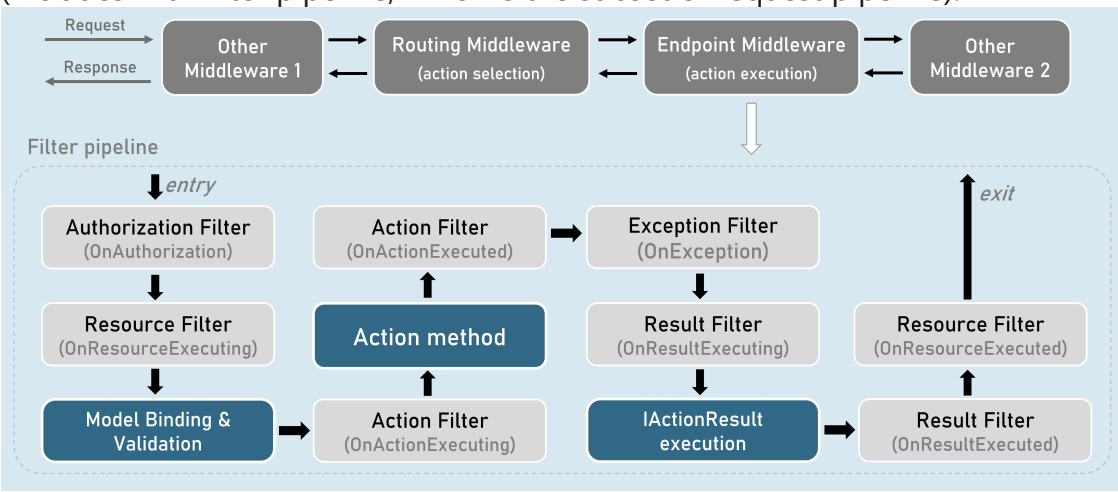
Explain different types of filters

Filters provide the capability to run the code before or after the specific stage in request processing pipeline, it could be either MVC app or Web API service. Filters performs the tasks like Authorization, Caching implementation, Exception handling etc. ASP.NET Core also provide the option to create custom filters. There are 5 types of filters supported in ASP.NET Core Web apps or services.

- **Authorization filters** run before all or first and determine the user is authorized or not.
- **Resource filters** are executed after authorization. OnResourceExecuting filter runs the code before rest of filter pipeline and OnResourceExecuted runs the code after rest of filter pipeline.
- **Action filters** run the code immediately before and after the action method execution. Action filters can change the arguments passed to method and can change returned result.
- **Exception filters** used to handle the exceptions globally before writing the response body
- **Result filters** allow to run the code just before or after successful execution of action results.

Explain request processing pipeline [or] filter pipeline in asp.net core?

The following diagram explains the complete request processing pipeline (includes with filter pipeline, which is the subset of request pipeline).



How cookies work in asp.net core?

A cookie is a small amount of data that is persisted across requests and even sessions. Cookies store information about the user. The browser stores the cookies on the user's computer. Most browsers store the cookies as key-value pairs.

- At first, server sends a cookie (key/value pair) by using a response header called "Set-Cookie". Then the browser receives it and stores the cookie in the browser memory.
- For each subsequent request, the same cookie will be sent to the server with "Cookie" request header.

Write a cookie in ASP.NET Core:

```
Response.Cookies.Append(key, value);
```

Delete a cookie in ASP.NET Core:

```
Response.Cookies.Delete(somekey);
```

How do you short circuit the request in an action filter?

If I don't want the action method to be executed, I'll short-circuit the request in OnActionExecuting() method of Action filter, as it executes before execution of the action method.

```
public class MyActionFilter : IActionFilter
{
    public void OnActionExecuting(ActionExecutingContext context)
    {
       if (condition)
       {
            //short-circuit the request
            context.Result = some_action_result;
       }
    }

    public void OnActionExecuted(ActionExecutedContext context)
    {
    }
}
```

How do you use dependency injection in action filter?

The ServiceFilterAttribute or TypeFilterAttribute helps me to have constructor injection (DI) in an action filter class.

If the filter class needs arguments to constructor, I would use TypeFilterAttribute; otherwise ServiceFilterAttribute.

```
public class FilterClassName : IActionFilter
{
   private readonly IService _service;

   public FilterClassName(IService service)
   {
       _service = service;
   }
   public void OnActionExecuting(ActionExecutingContext context)
   {
       //TO DO: before logic here
   }
   public void OnActionExecuted(ActionExecutedContext context)
   {
      //TO DO: after logic here
   }
}
```

How do you override order of filters?

I would implement an interface called IOrderFilter to have a property called "Order" that assigns to an int value.

The before methods execute in ascending order of "Order" property value; and the after methods execute in descending order.

```
public class FilterClassName : IActionFilter, IOrderedFilter
{
  public int Order { get; set; } //Defines sequence of execution
  public FilterClassName(int order)
  {
    Order = order;
  }
  public void OnActionExecuting(ActionExecutingContext context)
  {
    //TO DO: before logic here
  }
  public void OnActionExecuted(ActionExecutedContext context)
  {
    //TO DO: after logic here
  }
}
```

How will you add global filters?

Global filters once added to the "Filters" collection in the Startup (Program) class, will be applied to all action methods of all controllers in the application.

```
builder.Services.AddControllersWithViews(options => {
   options.Filters.Add<FilterClassName>(); //add by type
   //or
   options.Filters.Add(new FilterClassName()); //add filter instance
});
```

Your submission

Cesar David Guerrero Regino

Explain different types of filters

Explain request processing pipeline [or] filter pipeline in asp.net core?

How cookies work in asp.net core?

How do you short circuit the request in an action filter?

How do you use dependency injection in action filter?

How do you override order of filters?

How will you add global filters?