MVC

Christen Zarif Foad

Filters in .Net Core

- Filters are used to run code before or after certain stages in the request processing pipeline.
- There are many built-in filters for **authorization, logging, caching, exception handling** and so on.
- You can also create custom filters to handle concerns for your application in the way you want
- Filters also help to avoid code-duplication across action methods.

Different Types of Filters

- **Filters** implement the IFilterMetadata interface of the Microsoft.AspNetCore.Mvc.Filters namespace.

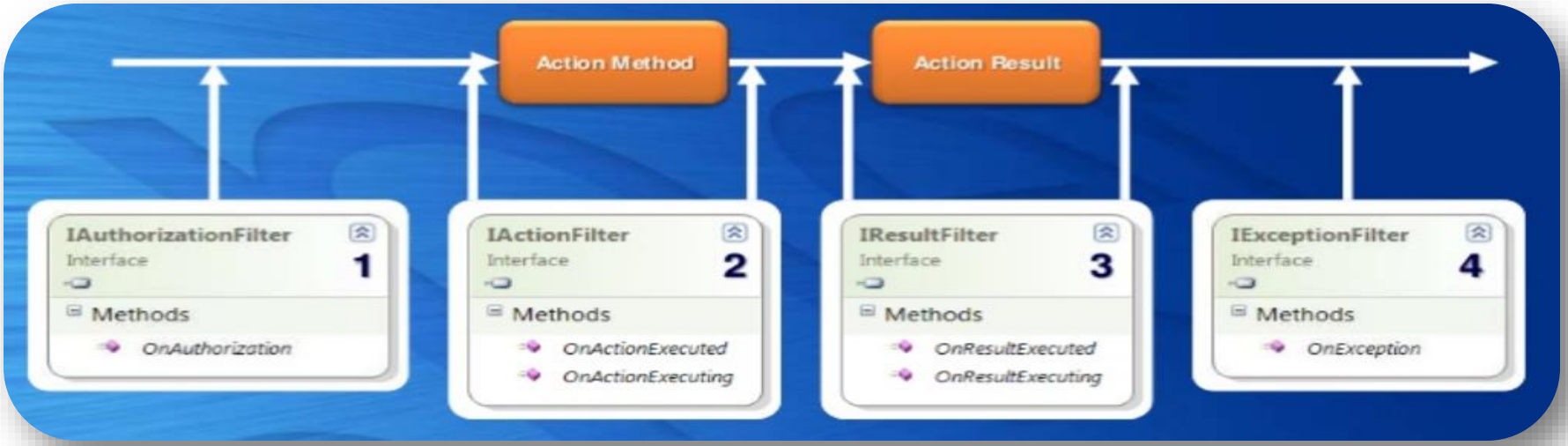
Different Types of Filters

- There are many **different types of filters**, important ones are defined in the below table along with the interface that contains them.

Filter	Interfaces that contains them	Description
Authorization	IAuthorizationFilter, IAsyncAuthorizationFilter	Used to apply authorization and security policy
Action	IActionFilter, IAsyncActionFilter	Used to perform a specify work immediately before or after an action method is performed
Result	IResultFilter, IAsyncResultFilter	Used to perform a specify work immediately before or after the result from an action method is processed
Exception	IExceptionFilter, IAsyncExceptionFilter	Used to handle exceptions

Order of Execution of Filters

- Filters are executed in the following order:
  - **Authorization Filters** are the first to execute.
  - Then comes the **Action Filters**.
  - At the last the **Result Filters** are executed.
  - The **Exception Filters** are executed only when an exception occurs.

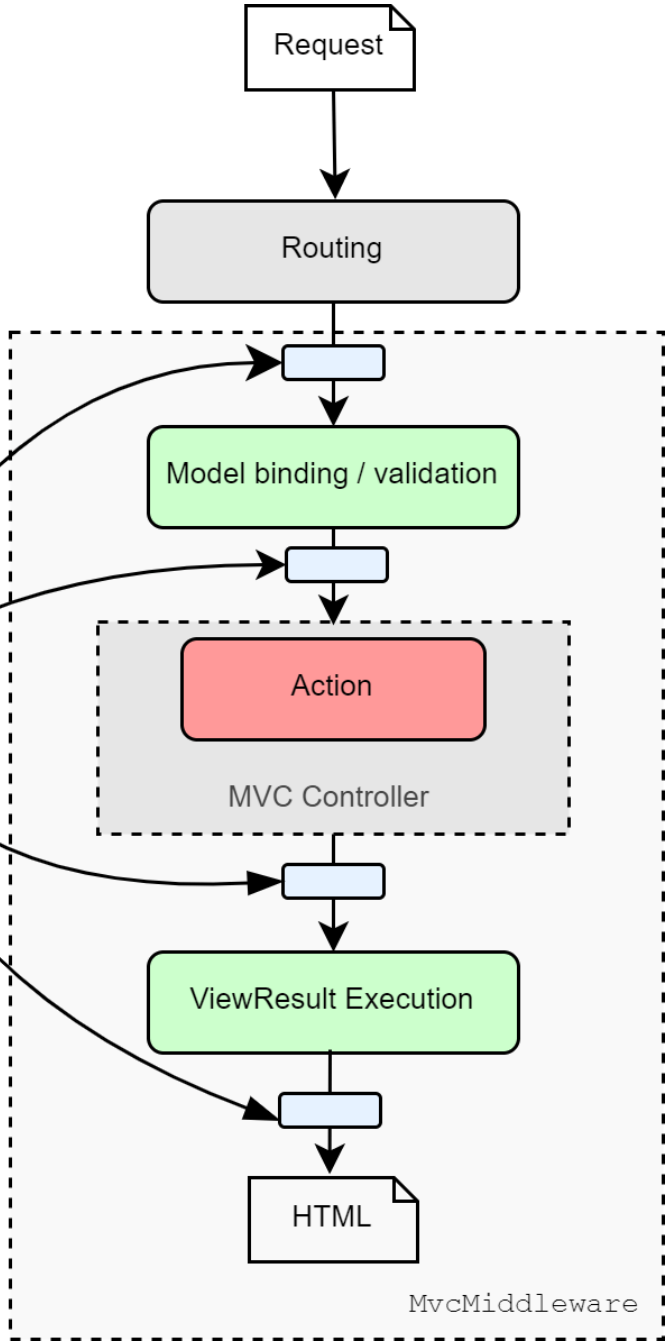


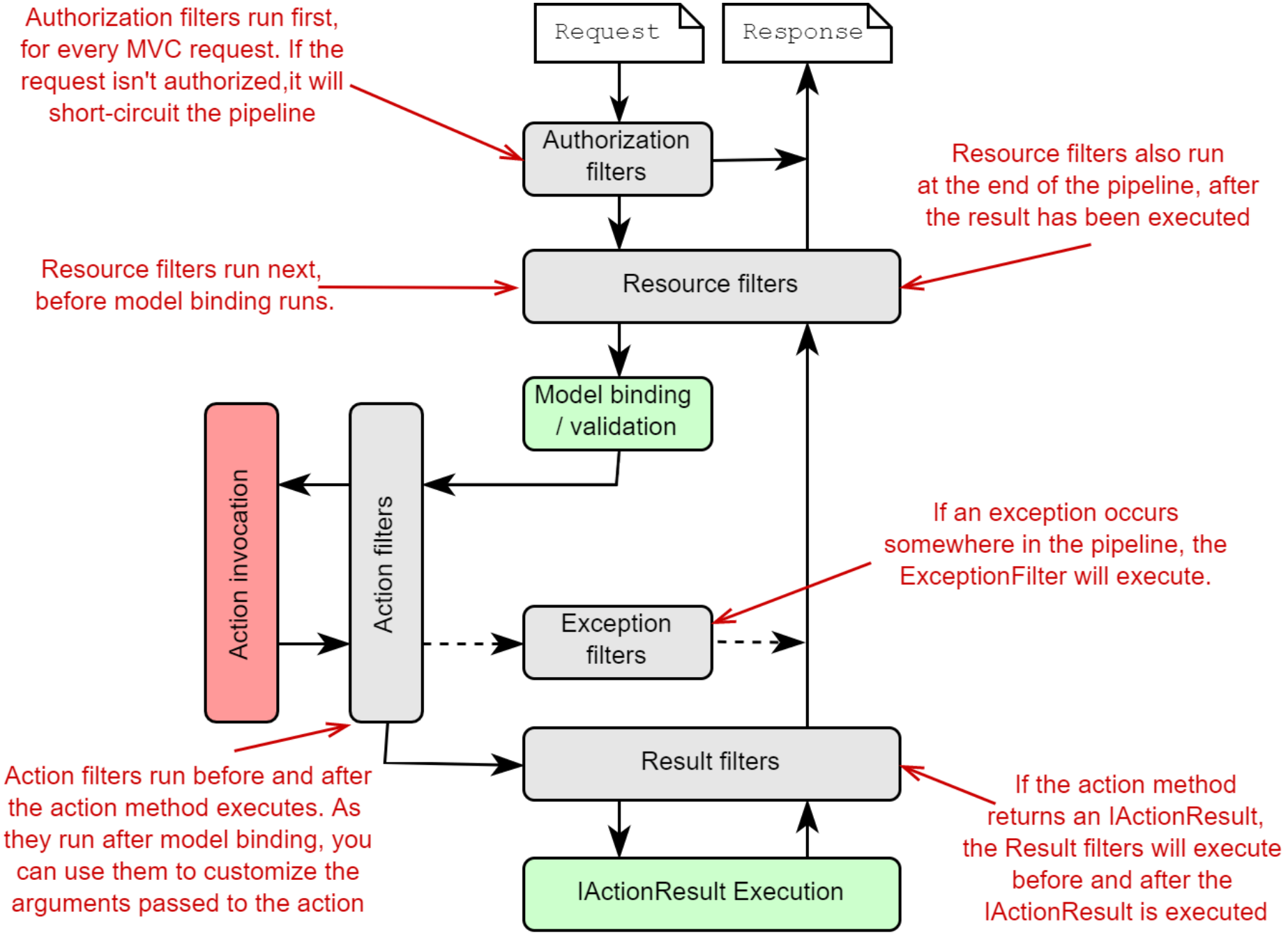
1. A request is received to the URL /recipe/1

2. The routing module matches the request to the RecipeController.View action and sets id=1

3. A variety of different MVC filters run as part of the MVC middleware

4. Filters run before model binding, before the action method runs, and before and after the IActionResult is executed




Authorization filters run first, for every MVC request. If the request isn't authorized,it will short-circuit the pipeline
Request
Response
Authorization filters
Resource filters also run at the end of the pipeline, after the result has been executed
Resource filters run next, before model binding runs.
Resource filters
Model binding / validation
Action invocation
Action filters
If an exception occurs somewhere in the pipeline, the ExceptionFilter will execute.
Exception filters
Result filters
Action filters run before and after the action method executes. As they run after model binding, you can use them to customize the arguments passed to the action
If the action method returns an IActionResult, the Result filters will execute before and after the IActionResult is executed
IActionResult Execution

- Different between AddIdentity & AddIdentityCore
- AUTOMAPPER;
- PageList
- MVC Area :
- Caching
- Repository Patter In mvc