Debugging and Error Handling in ASP.NET Core 6 Web API

Handling Errors and Formatting Responses

Nertil Poci

Software Architect

Overview

Multiple Environments

- Development
- Production

Error Handler Page

- Visualize errors in a common format

Catch Errors Using Filters

- Modify response based on error

Enrich Status Code Response Information

- Custom help links and names for status codes

Override Default Error Generator

Setting Up Multiple API Environments

Demo

Setting up Multiple Environments

- Create multiple app environment for testing

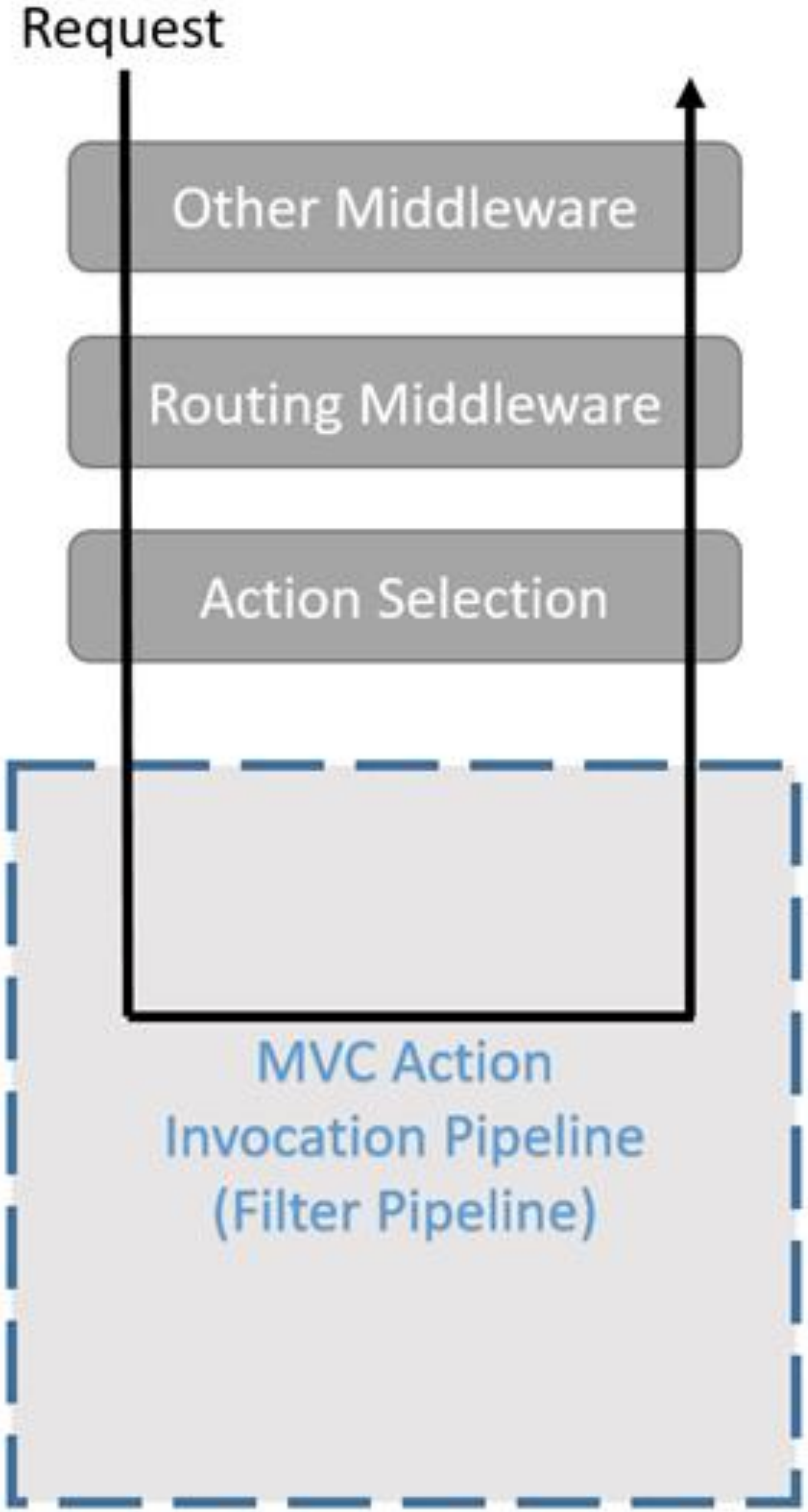
Creating Custom Error Handler Pages

Demo

Creating custom error handler page

- Control how errors are displayed using an error page

Modifying API Responses Using Filters


Request
Other Middleware
Routing Middleware
Action Selection
MVC Action
Invocation Pipeline
(Filter Pipeline)

Filter Types

Authorization Filter

Runs first.
Determine whether the user is authorized for the request.

Resource Filters

After Authorization.

Action Filters

Run immediately before and after an action method is called.

Exception Filters

Apply global policies to unhandled exceptions that occur before the response body has been written

Result Filters

Run immediately before and after the execution of action results.
Run only when the action method executes successfully.

Demo

Modifying API Responses using Filters

- Create filters
- Modify response output using filters

Mapping Error Codes to Instruction Links

Demo

Mapping Error Codes to Instruction Links

- Map error codes to documentation links

Customizing Default Error Details

Demo

Customizing Default Error Details

- Implement ProblemDetailssFactory
- Customize output of the error

Summary

New Environment

- Production

Custom Error Handler

- Production
- Development

Exception Filter

Client Error Mapping

- Enrich status code information using client error mapping

Override Default Problems Factory with Custom Implementation