

# Handling Exceptions and Logging



**Kevin Dockx**

Architect

@Kevindockx | [www.kevindockx.com](http://www.kevindockx.com)

# Coming Up



**Handling exceptions in minimal APIs**

**Improving error responses with problem details**

**Logging in minimal APIs**



# Handling Exceptions in Minimal APIs

Handling exceptions is a common task for any ASP.NET Core application

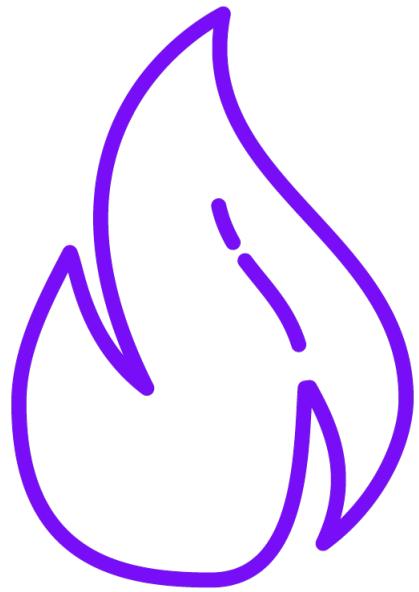
- It's not that different for minimal APIs



# Handling Exceptions in Minimal APIs



# Handling Exceptions in Minimal APIs



**Developer exception page  
middleware**



# Developer Exception Page Middleware

**Exposes stack traces for unhandled exceptions**

- Useful during development
- Should **NOT** be used outside of a development environment



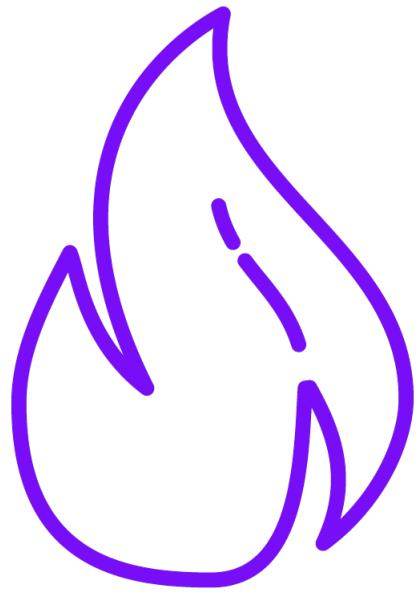
# Developer Exception Page Middleware

**Enabled by default when**

- Running in the development environment
- **AND** the app is set up with a call into `WebApplication.CreateBuilder`



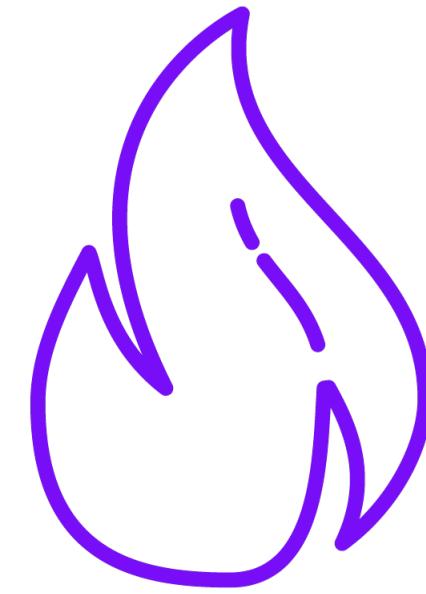
# Handling Exceptions in Minimal APIs



**Developer exception page  
middleware**



# Handling Exceptions in Minimal APIs



**Developer exception page  
middleware**



**Exception handler middleware**



# Exception Handler Middleware

Produces an error payload without exposing stack traces

- Useful when **NOT** working in a development environment
- Handles & logs exceptions



# Exception Handler Middleware

**NOT enabled by default**

- Enable with a call into  
`app.UseExceptionHandler`



# Demo



## Using the developer exception page middleware



# Demo



## Using the exception handler middleware



```
{  
  "type": "https://tools.ietf.org/html/rfc7231#section-6.6.1",  
  "title": "An error occurred while processing your request.",  
  "status": 500  
}
```

## Improving Error Responses with Problem Details



```
// enable  
builder.Services.AddProblemDetails();
```

## Improving Error Responses with Problem Details

Provided via [IProblemDetailsService](#)

Default implementation is included with ASP.NET Core



# Effects of Enabling the Default Problem Details Service



The developer exception page middleware will automatically generate a problem details response



The exception handler middleware will automatically generate a problem details response



The status code page middleware can be configured to generate problem details responses for empty bodies (e.g.: 400, 404, ...)



## Demo



**Improving error responses with  
problem details**



# Logging in Minimal APIs

**Not different from “regular” ASP.NET Core**

- Configure a Logger
- Inject it to log where needed



# Demo



## Logging in minimal APIs



# Summary



## Developer exception page middleware

- Returns stack traces, for use during development

## Exception handler middleware

- Handles and logs exceptions but doesn't return stack traces, for use when not in development



# Summary



**If possible, return a problem details payload in the error response body**

- Standardized response for API errors

**Log like you would log in “regular” ASP.NET Core**



**Up Next:**

# **Implementing Business Logic with Endpoint Filters**

---

