

# 第94期-处理WebAPI中的错误

2020年3月23日 15:15

## 开发人员异常页

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
}
```

## 自定义异常处理程序

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseExceptionHandler("/error-local-development");
    }
    else
    {
        app.UseExceptionHandler("/error");
    }
}
```

```
[ApiController]
public class ErrorController : ControllerBase
{
    [Route("/error-local-development")]
    public IActionResult ErrorLocalDevelopment(
        [FromServices] IWebHostEnvironment webHostEnvironment)
    {
        if (webHostEnvironment.EnvironmentName != "Development")
        {
            throw new InvalidOperationException(
                "This shouldn't be invoked in non-development environments.");
        }

        var context = HttpContext.Features.Get<ExceptionHandlerFeature>();

        return Problem(
            detail: context.Error.StackTrace,
            title: context.Error.Message);
    }
}
```

```

        [Route("/error")]
        public IActionResult Error() => Problem();
    }

```

## 使用过滤器处理异常

```

public class HttpResponseException : Exception
{
    public int Status { get; set; } = 500;

    public object Value { get; set; }
}

```

```

public class HttpResponseExceptionFilter : IActionFilter, IOrderedFilter
{
    public int Order { get; set; } = int.MaxValue - 10;

    public void OnActionExecuting(ActionExecutingContext context) { }

    public void OnActionExecuted(ActionExecutedContext context)
    {
        if (context.Exception is HttpResponseException exception)
        {
            context.Result = new ObjectResult(exception.Value)
            {
                StatusCode = exception.Status,
            };
            context.ExceptionHandled = true;
        }
    }
}

```

```
options.Filters.Add(new HttpResponseExceptionFilter());
```

## 验证失败错误响应

验证失败时通过 [ValidationProblemDetails](#) 封装错误。

使用 [InvalidModelStateResponseFactory](#) 构造验证失败的错误响应。

```

services.AddControllers()
    .ConfigureApiBehaviorOptions(options =>
    {
        options.InvalidModelStateResponseFactory = context =>

```

```

        {
            var result = new
BadRequestObjectResult(context.ModelState);

            result.ContentTypes.Add(MediaTypeNames.Application.Json);
            result.ContentTypes.Add(MediaTypeNames.Application.Xml);

            return result;
        };
    });

```

## 客户端错误响应

MVC 会将 400 错误结果转换为具有 ProblemDetails 的结果。

### 实现 ProblemDetailsFactory

MVC 使用 Microsoft.AspNetCore.Mvc.ProblemDetailsFactory 生成 ProblemDetails 和 ValidationProblemDetails 的所有实例，使用 ApiBehaviorOptionsSetup 调度。

```

public void ConfigureServices(IServiceCollection serviceCollection)
{
    services.AddTransient<ProblemDetailsFactory, CustomProblemDetailsFactory>
();
}

```

### 使用 ApiBehaviorOptions.ClientErrorMapping 定义 Link 和 Title

```

services.AddControllers()
    .ConfigureApiBehaviorOptions(options =>
        options.SuppressModelStateInvalidFilter = true;
        options.SuppressMapClientErrors = true;
        options.ClientErrorMapping[404].Link
= "https://httpstatuses.com/404";
    });

```