Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

Global exception handling in asp.net core mvc

Suggested Videos

Part 57 - Handling 404 not found in asp.net core mvc | Text | Slides
Part 58 - Centralised 404 error handling in ASP.NET Core | Text | Slides
Part 59 - UseStatusCodePagesWithRedirects vs UseStatusCodePagesWithReExecute | Text | Slides

In this video we will discuss how to implement **global exceptions handler in ASP.NET Core MVC**

Throwing an Exception in ASP.NET Core

Consider the following Details action method in HomeController. We are deliberately throwing an exception using the throw keyword.

```
public ViewResult Details(int? id)
{
    throw new Exception("Error in Details View");

    // Rest of the code
}
```


Complete Tutorials

JavaScript tutorial

Bootstrap tutorial

Angular tutorial for beginners

Angular 5 Tutorial for beginners

Important Videos

The Gift of Education

Web application for your business

UseDeveloperExceptionPage Middleware in ASP.NET Core

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

    // Rest of the code
}
```

We have plugged in the DeveloperExceptionPage middleware into the HTTP request processing pipeline for the Development environment. So if we run the application in Development environment, then we see the following developer exception page if there is an unhandled exception.

How to become .NET developer

Resources available to help you

Dot Net Video Tutorials

ASP.NET Core Tutorial

Angular 6 Tutorial

Angular CRUD Tutorial

Angular CLI Tutorial

Angular 2 Tutorial

Design Patterns

SOLID Principles

ASP.NET Web API

Bootstrap

AngularJS Tutorial

jQuery Tutorial

JavaScript with ASP.NET Tutorial

JavaScript Tutorial

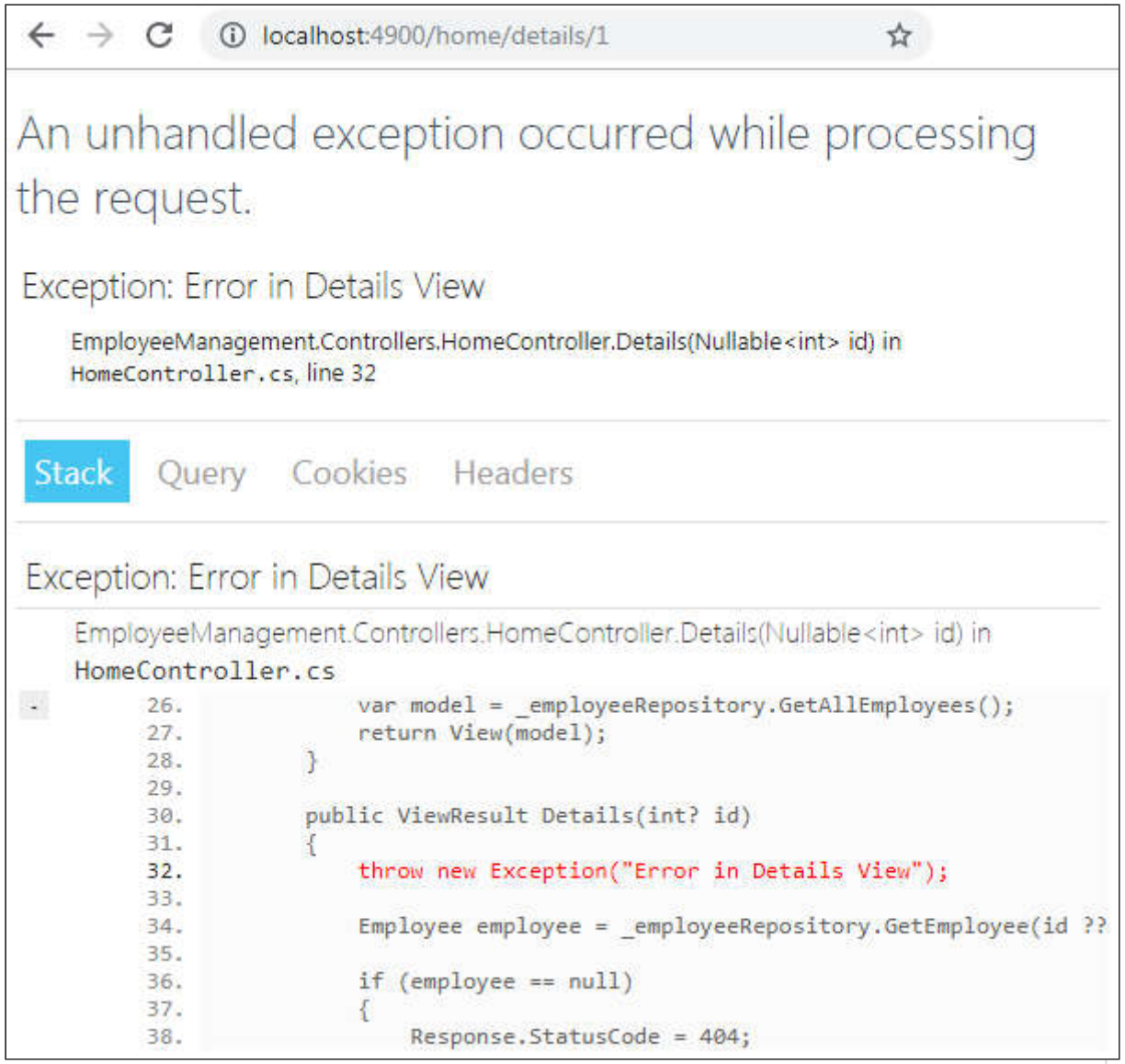
Charts Tutorial

LINQ

LINQ to SQL

LINQ to XML

Entity Framework



We discussed the significance and the use of DeveloperExceptionPage middleware in Part 13 of ASP.NET Core tutorial

As the name implies, the DeveloperExceptionPage middleware must be used only on the Development environment. Using this page on a non-development environment like Production for example is a security risk as it contains detailed exception information that could be used by an attacker. Also this exception page does not make any sense to

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Dot Net Interview Questions

Slides

Entity Framework

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

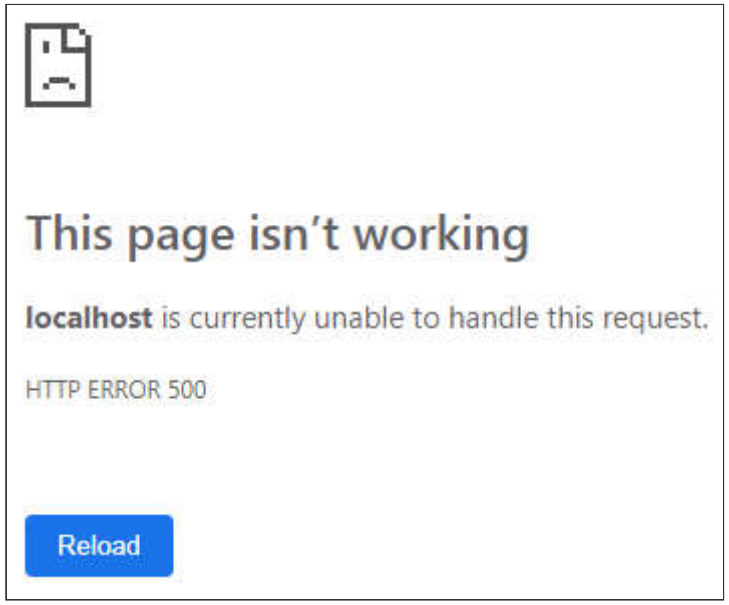
the end user.

Unhandled Exception on Non-Development Environment in ASP.NET Core

On your local development machine to simulate running the application on production environment set ASPNETCORE_ENVIRONMENT variable to Production.

```
"ASPNETCORE_ENVIRONMENT": "Production"
```

By default, if there is an unhandled exception in a non-development environment like production, we see the following default page.



Notice, other than displaying that there was a 500 http error we do not see any other information. Error 500 means, there was an error on the server which the server did not know how to handle.

This default page is not very useful for the end user. We want to handle the exception and redirect to the user to custom error view which is more useful and meaningful.

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Java Video Tutorials

Part 1 : Video | Text | Slides

Part 2 : Video | Text | Slides

Part 3 : Video | Text | Slides

Interview Questions

C#

SQL Server

Written Test

Exception Handling in ASP.NET Core

Step 1 : For a non-development environment, add the Exception Handling Middleware to the request processing pipeline using UseExceptionHandler() method. We do this in the Configure() method of the Startup class. Exception Handling Middleware looks for ErrorController.

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Error");
    }

    // Rest of the code
}
```

Step 2 : Implement the ErrorController that retrieves the exception details and returns the custom Error view. In a production application, we do not display the exception details on the error view. We instead log them to a database table, file, event viewer etc, so a developer can review them and provide a code fix if required. We will discuss logging in a later video.

```
public class ErrorController : Controller
{
    [AllowAnonymous]
    [Route("Error")]
    public IActionResult Error()
    {
        // Retrieve the exception Details
        var exceptionHandlerPathFeature =
              HttpContext.Features.Get<IExceptionHandlerPathFeature>();

        ViewBag.ExceptionPath = exceptionHandlerPathFeature.Path;
        ViewBag.ExceptionMessage = exceptionHandlerPathFeature.Error.Message;
        ViewBag.StackTrace = exceptionHandlerPathFeature.Error.StackTrace;
```

```
            return View("Error");
        }
    }
```

Please note : IExceptionHandlerPathFeature is in Microsoft.AspNetCore.Diagnostics namespace.

Step 3 : Implement Error View

```
<h3>An occured while processing your request. The support
    team is notified and we are working on the fix</h3>
<h5>Please contact us on pragim@pragimtech.com</h5>
<hr />
<h3>Exception Details:</h3>
<div class="alert alert-danger">
    <h5>Exception Path</h5>
    <hr />
    <p>@ViewBag.ExceptionPath</p>
</div>

<div class="alert alert-danger">
    <h5>Exception Message</h5>
    <hr />
    <p>@ViewBag.ExceptionMessage</p>
</div>

<div class="alert alert-danger">
    <h5>Exception Stack Trace</h5>
    <hr />
    <p>@ViewBag.StackTrace</p>
</div>
```

No comments:

Post a Comment

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Enter your comment...

Comment as: toilati123vn@g ▼

Publish Preview

Links to this post

Create a Link

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)