

Developer Exception Page Middleware in ASP.NET Core || UseDeveloperExceptionPage

.

Understanding Developer Exception Page Middleware:

- ▶ By default, the ASP.NET Core application simply returns a status code for an exception that is not handled by the application.
- ▶ Let us understand this with an example by modifying the `Configure()` method of the startup class where we throw an exception.

Throw the Error:- not display proper error page.

- `app.Run(async (context) =>`
- `{`
- `throw new Exception("Error Occured");`
- `//not reached the code here after throw the exception`
- `await context.Response.WriteAsync("Hello World!");`
- `});`



The website cannot display the page

HTTP 500

Most likely causes:

- The website is under maintenance.
- The website has a programming error.

What you can try:

- Retype the address.
- [Go back to the previous page.](#)
- [More information](#)

How to use DeveloperExceptionPage Middleware in ASP.NET Core Application?

- ▶ If you want your application to display a page that shows the detailed information about the unhandled exception, then you need to use the Developer Exception Page middleware. Let us modify the `Configure()` method as shown below to add the Developer Exception Page middleware which will handle the unhandled exception that occurred in our application

User developer Exception Page.

- `public void Configure(IApplicationBuilder app, IHostingEnvironment env)`
- `{ if (env.IsDevelopment())`
- `{`
- `app.UseDeveloperExceptionPage();`
- `}`
- `app.Run(async (context) =>`
- `{`
- `throw new Exception("Error Ocuured");`
- `//not reached the code here after throw the exception`
- `await context.Response.WriteAsync("Hello World!");`
- `});`
- `}`

```
public void Configure(IApplicationBuilder app, IHostingEnv  
{  
    if(env.IsDevelopment()){  
        app.UseDeveloperExceptionPage();  
    }  
  
    app.Run(async (context) =>  
    {  
        throw new Exception("Error Ocuured");  
        await context.Response.WriteAsync(" Hello World");  
    });
```


An unhandled exception occurred while processing the request.

Exception: Error Ocuured

Ex3_ExceptionPageMiddelWare.Startup+<>c+<<Configure>b_1_0>d.MoveNext() in Startup.cs, line 30

Stack Query Cookies Headers

Exception: Error Ocuured

Ex3_ExceptionPageMiddelWare.Startup+<>c+<<Configure>b_1_0>d.MoveNext() in **Startup.cs**
30. **throw new Exception("Error Ocuured");**
Microsoft.AspNetCore.Diagnostics.DeveloperExceptionPageMiddleware.Invoke(HttpContext context)

Show raw exception details



Type here to search



11:41
17-01-2021

Developer Exception Page Tabs

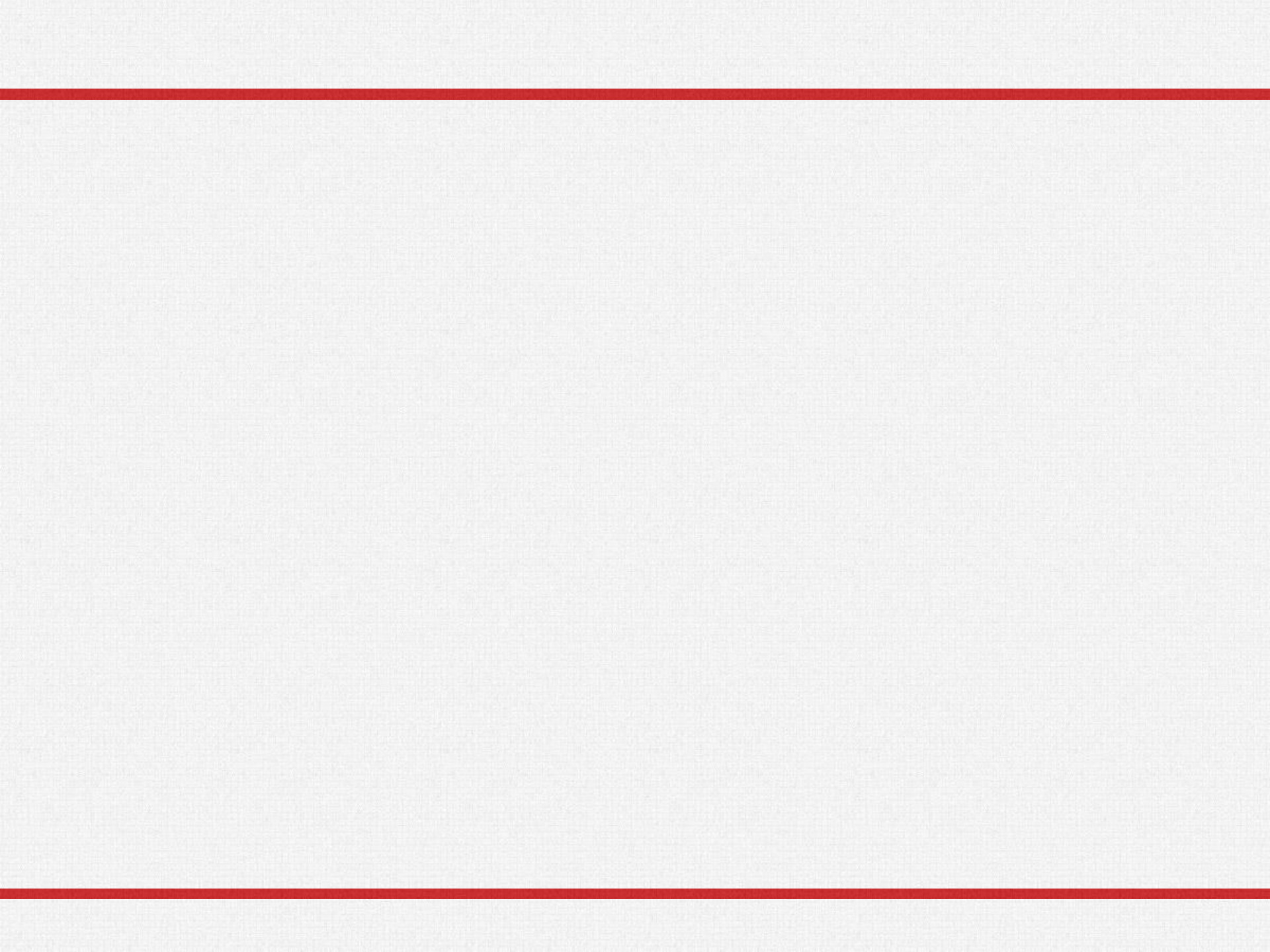
- ▶ As you can see in the above image, the Developer Exception Page contains four tabs such as Stack, Queue, Cookies, and Headers.
 - ▶ The Stack tab gives the information of stack trace which indicated where exactly the exception occurred, the file name and the line number that causes the exception.
 - ▶ Query tab gives information about the query strings.
 - ▶ The Cookies tab displays the information about the cookies set by the request
 - ▶ The Header tab gives the information about the headers which is sent by the client when makes the request.
- ▶ Now if you verify the Query tab and Cookies tab, then you will not see any information as you are not passing any query string value in URL or you are not setting the cookies in the request.
- ▶ **Note:** Please Enable the Developer Exception Page Middleware only when the application is running in the Development environment. You don't want to share detailed exception information when the application is running in the production environment.

How to Customize the UseDeveloperExceptionPage Middleware in ASP.NET Core?

- ▶ If you want then you can also customize the UseDeveloperExceptionPage middleware. The point that you need to remember is whenever you want to customize a middleware component in ASP.NET Core then you need to use the respective Options object. For example
 - ▶ UseDeveloperExceptionPage => to customize this middleware use DeveloperExceptionPageOptions object
 - ▶ UseDefaultFiles => to customize this middleware use DefaultFilesOptions object
 - ▶ UseStaticFiles => to customize this middleware use StaticFileOptions object
 - ▶ UseFileServer => to customize this middleware use FileServerOptions object
- ▶ As we are going to customize the **UseDeveloperExceptionPage()** middleware component, so we need to use the **DeveloperExceptionPageOptions** object

Where do we need to configure the `UseDeveloperExceptionPage` Middleware?

- ▶ We need to configure the `UseDeveloperExceptionPage()` Middleware as early as possible in the application's request processing pipeline so that it can handle the unhandled exception and then display the Developer Exception Page with the detailed information about the exception.
- ▶ Let us see what happened when we configure the `UseDeveloperExceptionPage()` middleware after the middleware using is Registered using the `Run()` method.



Thanks