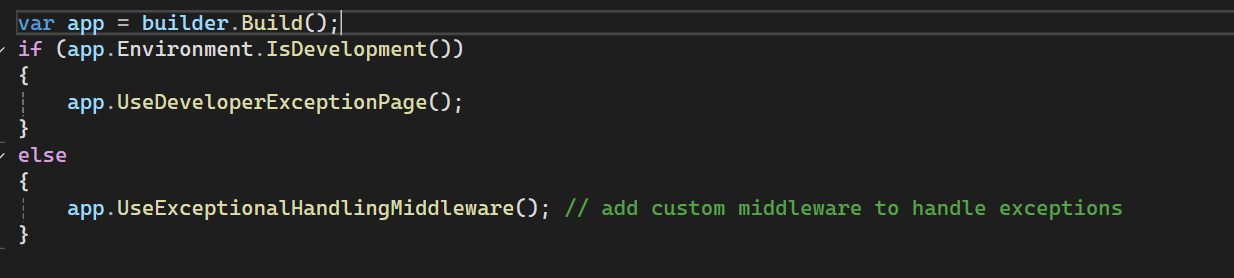
**Error Handling – AspNet**

**Exceptional Handling Middleware**

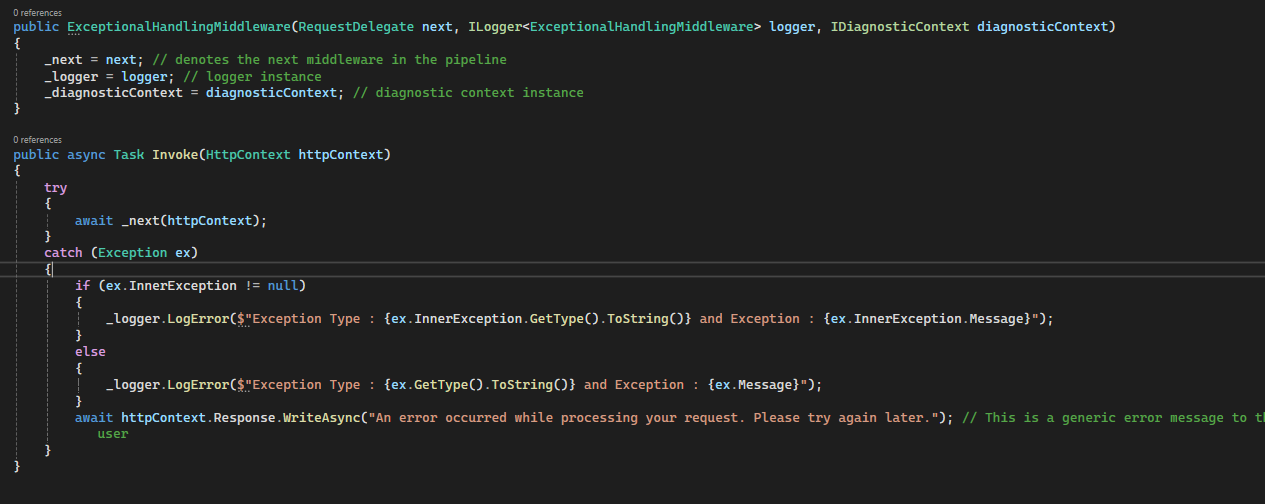
* In development env we set up to use DevelopmentException page but in production it directly gives only HTTP 500 error
* In order to trace error and its details in all environments we need to use **Exceptional Handling Middleware**
* Exceptional Handling Middleware handles all the error in the entire life cycle of a request pipeline
* It should be added as the first Middleware (before routing middleware)

Request -> Exceptional -> OtherMiddlewares -> Routing -> Others -> EndPoint Middleware(action execution)

* We need to create our own Custom Exceptional Handling Middleware
* Create a new folder “Middleware” here you can create your Middleware classes (suffix with Middleware)
* Middleware class has Invoke method which calls next middleware
* Add the custom middleware as starting point in app

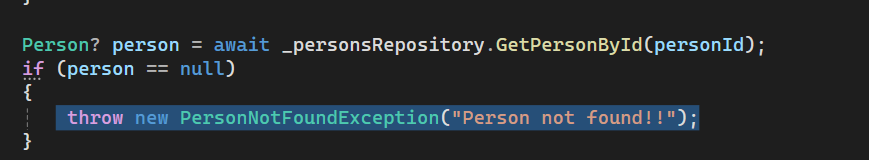


* Inside Invoke method use try catch block so that it can catch all exceptions of all next middlewares

**Custom Exception Classes:**

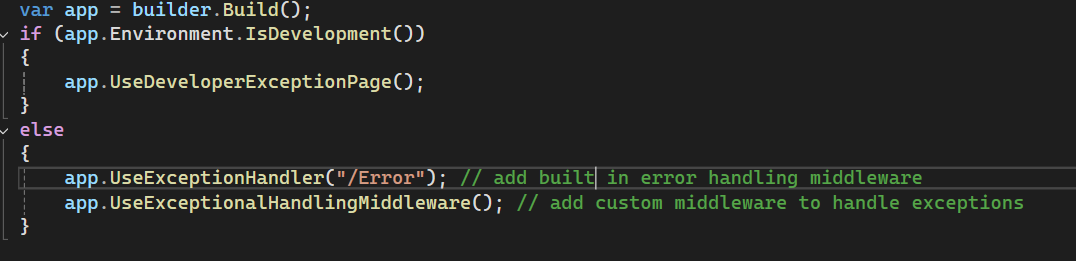
* Are user defined exception classes that inherits from System.Exception class to provide context/domain related error messages
* Eg: say like you want to update a person in db but person is not found in such cases you can throw PersonNotFound custom Exceptions instead of throwing Argument Exception as it most commonly used in all scenarios
* Every Exception class should has atleast 3 constructor recommended by Microsoft
* Create Exception classes as sperated Class library so that they can be referenced in all our projects when we want



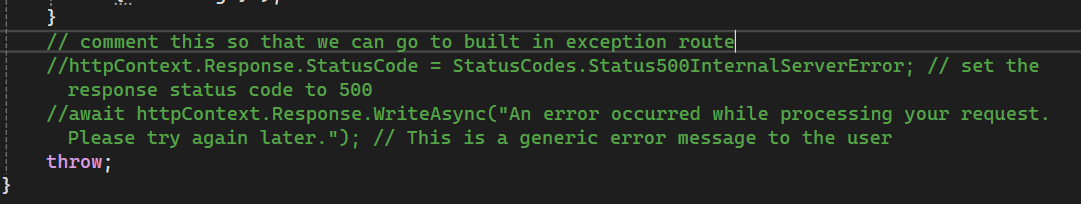


**Built in Exception Handle Middleware**

* The built-in UseExceptionalHandler() middleware redirects to same specified route when an exception occurs.
* It should be added at first even before Custom middleware and specify the route path



* It is better to create a controller with route that is configured in Exceptional Handler so that we can display context based Error messages (views)
* In your custom handling middleware make sure to add throw so that it can again go to same request and call predefined middleware



I will design and develop features for Outlook and Teams using ASP.NET, C#, and Azure, ensuring reliability with NUnit testing. I will collaborate with cross-functional teams in Agile workflows, conduct code reviews, and optimize performance by monitoring and debugging applications. Additionally, I will explore innovative technologies, refine features based on feedback, and create technical documentation while aligning development with business goals.