MiddleWare Creation in .net core

Suppose we have requirement to sent client a custom Exception message and for development , message should be full details then Will create ExceptionMessage Class

Public class APIResponse

{

Private int StatusCode;

Private string Message;

Public APIResponse(int statuscode, string message)

{

StatusCode=statuscode;

Message= message??= GetDefaultMessage(statuscode);

}

Private string GetDefaultMessage(int statuscode)

{

Return Statuscode switch

{

400 => “Bad Request”,

401 => “Not Authorize”

404 => “Resource Not found”,

500 => “Server Error”

503 => “Service not available”

}

}

}

Public class APIException : APIResponse

{

Private string Details;

Public APIException(int Statuscode,string message=null, string details=null)

{

Details= details;

}

}

// Creation of Middleware

Public class ExceptionMiddleWare

{

Private readonly RequestDelegate \_next;

Private readonly ILogger< ExceptionMiddleWare> \_logger;

Private readonly IHostEnvironment \_env;

Public ExceptionMiddleWare(RequestDelegate next, ILogger< ExceptionMiddleWare> logger, IHostEnvironment env)

{

\_next=next;

\_logger=logger;

\_env=env;

}

Public async Task InvokeAsync(HttpContext context)

{

Try

{

await \_next(context); // if there is no error then execute next Middleware in pipeline

}

Catch(Exception ex)

{

\_logger.LogError(ex,ex.Message);

Context.Respose.ContentType=”application/json”;

Context.Response.StatusCode=(int)HttpStatusCode.InternalServerError;

Var response = \_env.IsDevelopment()?

New APIException((int)HttpStatusCode.InternalServerError,ex.Message, ex.StackTrace.ToString():

New APIException((int)HttpStatusCode.InternalServerError);

Var json = JsonSerializer.Serilize(response);

Await context.Reponse.WriteAsync(json);

}

}

}

// In Startup.cs

Public void Configure(IApplicationBuilder app, IHostEnvironment env)

{

App.UseMiddleWare<ExceptionMiddleWare>();

}