**Hands-On: Stage 3 - Exception filter, Simple usage of Entity Framework, Knowledge on using log4net for logging - Day 64 – Hands on 1**

1. **Logging**

**HomeController.cs**

using Logging.Filter;

using Logging.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

namespace Logging.Controllers

{

    public class HomeController : Controller

    {

        [MyExceptionFilter]

        public IActionResult Index()

        {

            int a = 5;

            int b = 0;

            int divison = a / b;

            ViewBag.Message = "The Division is: " +divison;

            return View();

        }

        public IActionResult Privacy()

        {

            return View();

        }

    }

}

**MyExceptionFilter.cs**

using log4net;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Filters;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace Logging.Filter

{

    public class MyExceptionFilter : ExceptionFilterAttribute, IExceptionFilter

    {

        private readonly ILog \_logger = LogManager.GetLogger(typeof(MyExceptionFilter));

        public override void OnException(ExceptionContext context)

        {

            \_logger.Error(context.Exception.Message);

            context.ExceptionHandled = true;

            context.Result = new ViewResult() { ViewName = "CustomErrors" };

        }

    }

}

**CustomErrors.cshtml**

@\*

    For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860

\*@

@{

    ViewBag.Title = "Error";

}

<div style="background-color: #A52A2A; color: White; height: 10px;">

</div>

<div style="background-color: #F5F5DC; color: White; height: 170px;">

    <div style="padding:20px;">

        <h3 style="color: Black;">

            Application Custom Error:

        </h3>

        <h4 style="color: Black;">

            Sorry, an Divide by Zero error occurred while processing your request.

        </h4>

        <br />

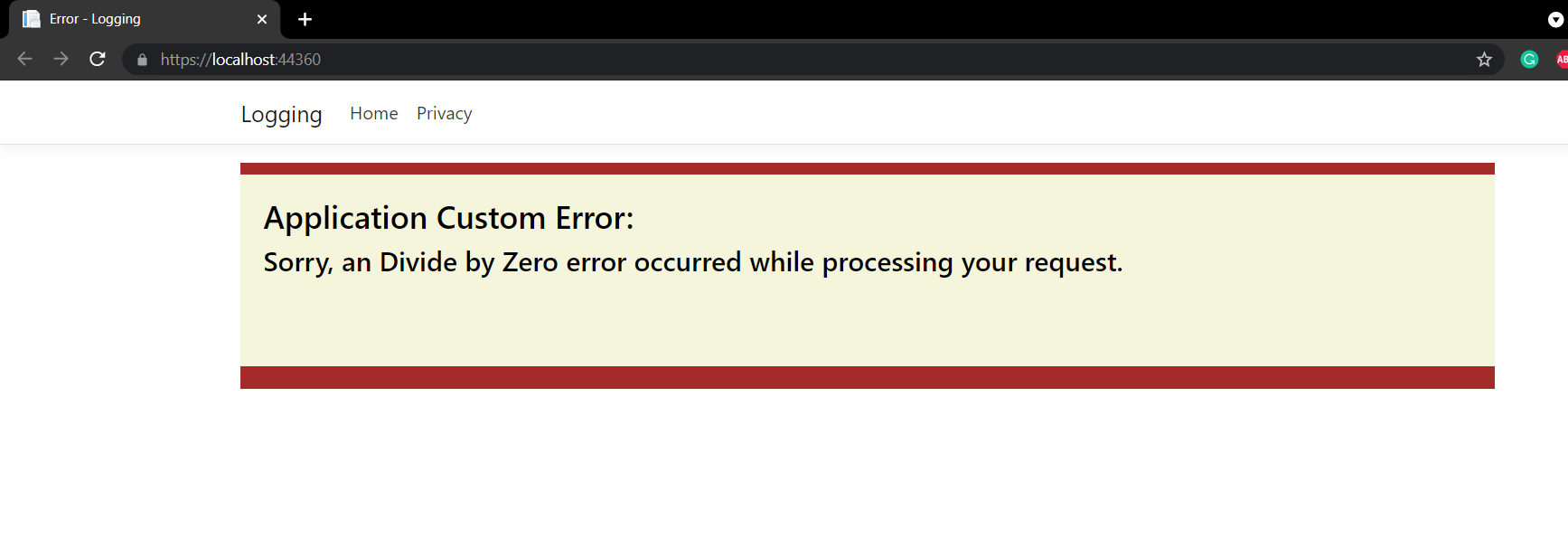
        <br />

    </div>

</div>

<div style="background-color: #A52A2A; color: White; height: 20px;">

</div>

**OUTPUT**

1. **Log4Net usage for logging**

**HomeController.cs**

using log4net;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Log4Net.Controllers

{

    public class HomeController : Controller

    {

        private static readonly ILog Log = LogManager.GetLogger(typeof(HomeController));

        public ActionResult Index()

        {

            try

            {

                Log.Debug("Log4Net usage for logging in ASP.NET MVC");

                Log.Info("First");

                Log.Warn("Second");

                throw new NullReferenceException();

            }

            catch (Exception exp)

            {

                Log.Error("Error");

                Log.Fatal("Fatal");

            }

            return View();

        }

        public ActionResult About()

        {

            ViewBag.Message = "Your application description page.";

            return View();

        }

        public ActionResult Contact()

        {

            ViewBag.Message = "Your contact page.";

            return View();

        }

    }

}

**Global.asax.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Optimization;

using System.Web.Routing;

namespace Log4Net

{

    public class MvcApplication : System.Web.HttpApplication

    {

        protected void Application\_Start()

        {

            log4net.Config.XmlConfigurator.Configure();

            AreaRegistration.RegisterAllAreas();

            FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

            RouteConfig.RegisterRoutes(RouteTable.Routes);

            BundleConfig.RegisterBundles(BundleTable.Bundles);

        }

    }

}

1. **Create a .Net core web application with a controller that is scaffolded with Entity framework options.**

**HomeController.cs**

using AccountDetails.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

namespace AccountDetails.Controllers

{

    public class HomeController : Controller

    {

        private AccountDbContext context;

        public HomeController(AccountDbContext dbContext)

        {

            context = dbContext;

        }

        public async Task<IActionResult> Index()

        {

            List<Account> accList = await context.Accounts.ToListAsync();

            return View(accList);

        }

        public IActionResult Create()

        {

            return View();

        }

        [HttpPost]

        [ValidateAntiForgeryToken]

        public async Task<IActionResult> Create([Bind("AccountId,AccountName")] Account obj)

        {

            if (ModelState.IsValid == true)

            {

                await context.Accounts.AddAsync(obj);

                context.SaveChanges();

            }

            else

            {

            return NotFound();

            }

            return RedirectToAction("Index");

        }

        public async Task<IActionResult> Edit(int id)

        {

            Account obj = await context.Accounts.FindAsync(id);

            return View(obj);

        }

        [HttpPost]

        [ValidateAntiForgeryToken]

        public async Task<IActionResult> Edit([Bind("AccountId,AccountName")] Account obj)

        {

            if (ModelState.IsValid == true)

            {

                context.Update(obj);

                await context.SaveChangesAsync();

            }

            else

            {

                return NotFound();

            }

            return RedirectToAction("Index");

        }

        public async Task<IActionResult> Delete(int id)

        {

            Account obj = await context.Accounts.FindAsync(id);

            return View(obj);

        }

        [HttpPost]

        [ActionName("Delete")]

        [ValidateAntiForgeryToken]

        public async Task<IActionResult> DeleteConfirm(int id)

        {

            Account obj = await context.Accounts.FindAsync(id);

            context.Accounts.Remove(obj);

            await context.SaveChangesAsync();

            return RedirectToAction(" Index ");

        }

    }

}

**AccountDbContext.cs**

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace AccountDetails.Models

{

    public class Account

    {

        public int AccountId { get; set; }

        public string AccountName { get; set; }

    }

    public class AccountDbContext:DbContext

    {

        public DbSet<Account> Accounts { get; set; }

    public AccountDbContext(DbContextOptions<AccountDbContext> options):base(options)

        {

        }

    }

}