**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

# МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

***Факультет Информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки 09.03.02 «Информационные системы и технологии»**

# ЛАБОРАТОРНАЯ РАБОТА № \_9\_\_

Дисциплина: BackEnd-разработка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тема: *Обработка ошибок в веб-приложении на основе ASP.NET*

*Core*\_\_\_\_\_\_\_\_\_\_

**Выполнил(а): студент(ка) группы \_\_221-3711\_\_\_\_\_\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_\_\_Костоваров А. С.

(Фамилия И.О.)

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

## Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_*

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

## Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва**

**2024**

В данной лабораторной работе мы ознакомились с методами обработки ошибок в веб-приложениях на ASP.NET Core. Для демонстрации обработки ошибок создадим приложение, обрабатывающее несколько типов ошибок с перенаправлением пользователя на кастомные страницы исключений.

Разберем методы обработки ошибок:

Обработка исключений:

Использование UseExceptionHandler для перенаправления на пользовательскую страницу ошибок в случае возникновения необработанных исключений.

Обработка ошибок HTTP-статуса:

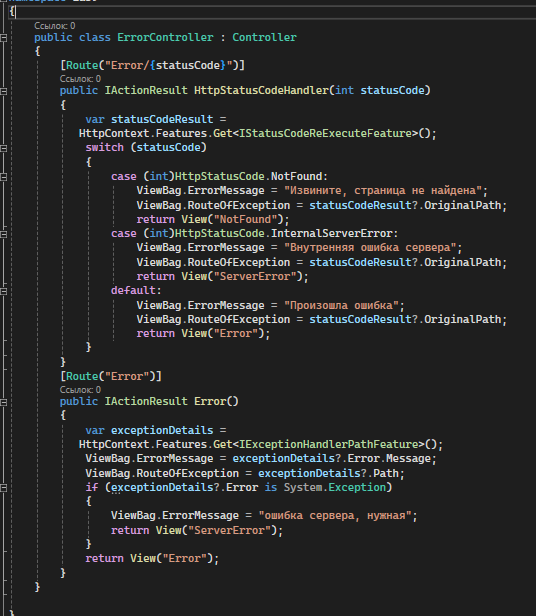
Использование UseStatusCodePagesWithReExecute для перенаправления на пользовательские страницы ошибок при возникновении ошибок HTTP-статуса Создание пользовательских страниц ошибок:

Создание кастомных представлений для отображения сообщений об ошибках (примеры будут далее).

Настроим файл program



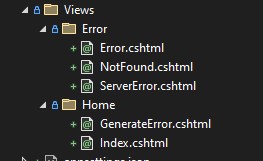
Будем использовать 2 типа страниц, одна общая для большинства типов ошибок и особые страница в зависимости от статуса исключения. Так же для этого выберем использование продукционной среды, т.к в противном случае будут использоваться страницы исключений режима разработчика.



Создадим контроллер обработки ошибок, он содержит метод получающий статус ошибки и перенаправляющий на соответствующее представление, если для статуса нет страницы, пользователь будет перенаправлен на общую страницу ошибки.

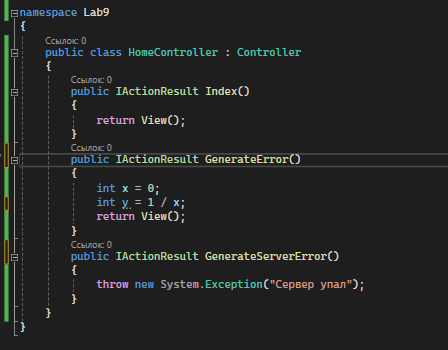
Так же настроим метод для общих исключений.

Далее нам нужно создать ошибки и страницы для них.



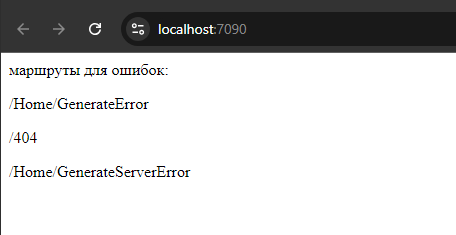
Создаем 3 типа страниц, для стандартной ошибки, ошибки 404 (страница не найдена) и 500 (ошибка сервера).

После этого в Home котроллере создадим действия по вызову 2х ошибок.

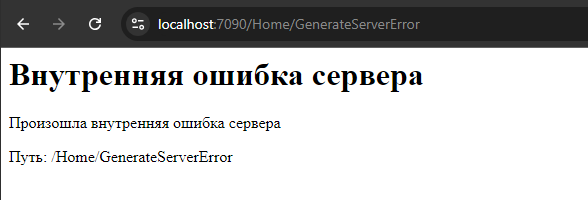


Данные действия вызывают ошибку деления на ноль и ошибку сервера.

Перейдем к тестам и демонстрации пользовательских страниц ошибок. Запускаем приложения без режима отладки и переходим на действия указанные на главной странице.



Перейдем к ошибке деления на ноль



Program

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddControllersWithViews();

var app = builder.Build();

if (app.Environment.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Error");

app.UseHsts();

}

app.UseStatusCodePagesWithReExecute("/Error/{0}");

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

HomeController

using Microsoft.AspNetCore.Mvc;

namespace Lab9

{

public class HomeController : Controller

{

public IActionResult Index()

{

return View();

}

public IActionResult GenerateError()

{

int x = 0;

int y = 1 / x;

return View();

}

public IActionResult GenerateServerError()

{

throw new System.Exception("Сервер упал");

}

}

}

ErrorController

public class ErrorController : Controller

{

[Route("Error/{statusCode}")]

public IActionResult HttpStatusCodeHandler(int statusCode)

{

var statusCodeResult =

HttpContext.Features.Get<IStatusCodeReExecuteFeature>();

switch (statusCode)

{

case (int)HttpStatusCode.NotFound:

ViewBag.ErrorMessage = "Извините, страница не найдена";

ViewBag.RouteOfException = statusCodeResult?.OriginalPath;

return View("NotFound");

case (int)HttpStatusCode.InternalServerError:

ViewBag.ErrorMessage = "Внутренняя ошибка сервера";

ViewBag.RouteOfException = statusCodeResult?.OriginalPath;

return View("ServerError");

default:

ViewBag.ErrorMessage = "Произошла ошибка";

ViewBag.RouteOfException = statusCodeResult?.OriginalPath;

return View("Error");

}

}

[Route("Error")]

public IActionResult Error()

{

var exceptionDetails =

HttpContext.Features.Get<IExceptionHandlerPathFeature>();

ViewBag.ErrorMessage = exceptionDetails?.Error.Message;

ViewBag.RouteOfException = exceptionDetails?.Path;

if (exceptionDetails?.Error is System.Exception)

{

ViewBag.ErrorMessage = "ошибка сервера, нужная";

return View("ServerError");

}

return View("Error");

}

}

Cshtml

@{

ViewData["Title"] = "Произошла ошибка";

}

<h1>Произошла ошибка</h1>

<p>@ViewBag.ErrorMessage</p>

@if (!string.IsNullOrEmpty(ViewBag.RouteOfException))

{

<p>Путь: @ViewBag.RouteOfException</p>

}

@{

ViewData["Title"] = "Страница не найдена";

}

<h1>Страница не найдена</h1>

<p>Страница по адресу "@ViewBag.RouteOfException" не найдена.</p>

@{

ViewData["Title"] = "Внутренняя ошибка сервера";

}

<h1>Внутренняя ошибка сервера</h1>

<p>Произошла внутренняя ошибка сервера</p>

@if (!string.IsNullOrEmpty(ViewBag.RouteOfException))

{

<p>Путь: @ViewBag.RouteOfException</p>

}

@{

ViewData["Title"] = "Генерация ошибки";

}

<h1>Генерация ошибки</h1>

<p>Это не должно тобразиться</p>

@{

ViewData["Title"] = "Дом";

}

<p>маршруты для ошибок:</p>

<p>/Home/GenerateError</p>

<p>/404</p>

<p>/Home/GenerateServerError</p>