HttpHandlers

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>Default</title>

</head>

<body>

<form id="form1" runat="server">

<div>

<a href="MyHandler.aspx">Ссылка на HTTP обработчик</a>

</div>

</form>

</body>

</html>

MyFirsGandler.cs

// Для того что бы экземпляры данного класса могли использоваться для обработки входящих запросов,

// необходимо зарегистрировать класс MyFirstHandler как обработчик в файле web.config

public class MyFirstHandler : IHttpHandler

{

// Свойство используется для того, что бы определить может ли один и тот же экземпляр

// класса HTTP обработчика использоваться для обработки нескольких запросов.

// В большинстве случаев свойство возвращает значение false - на каждый запрос создается новый экземпляр

// обработчика, но в тех ситуациях, когда создание обработчика связанно с большими накладными расходами,

// есть смысл указать, что обработчик может использоваться повторно, вернув значение true в данном свойстве.

public bool IsReusable

{

get { return false; }

}

// Данный метод будет запущен когда поступит запрос адресованный текущему HTTP обработчику.

public void ProcessRequest(HttpContext context)

{

context.Response.Write("Hello from HttpHandler");

}

}

WebConfig

<configuration>

<system.web>

<compilation debug="true" targetFramework="4.5" />

<httpRuntime targetFramework="4.5" />

</system.web>

<!--Регистрация класса MyFirstHandler как обработчика, который доступен по адресу MyHndler.aspx-->

<system.webServer>

<handlers>

<add name="MyFirstHandler" verb="GET" path="MyHandler.aspx" type="HttpHandlers.MyFirstHandler"/>

</handlers>

</system.webServer>

</configuration>

\*\*\*

GenericHandler

Add->New Item->Generic Handler – ashx

В даном файле не нужно реестрировать в веб конфиге

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<a href="01\_TextHandler.ashx">Text Handler</a><br />

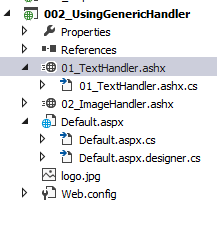
<a href="02\_ImageHandler.ashx">Image Handler</a>

</div>

</form>

</body>

</html>



public class \_01\_TextHandler : IHttpHandler

{

public void ProcessRequest(HttpContext context)

{

// Определение типа ответа

context.Response.ContentType = "text/plain";

// Формирование ответа

context.Response.Write("Hello World");

}

public bool IsReusable

{

get

{

return false;

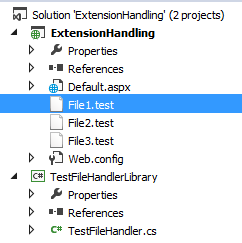
}

}

}

\*\*\*

Создание таблиц из xml-файла



<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>\*.test file handler</title>

</head>

<body>

<form id="form1" runat="server">

<div>

<p>

В данном приложении показано как с помощью HTTP обработчиков наладить обработку запросов к определенному

типу ресурсов. В данном примере обработчик будет срабатывать при запросах к файлам с расширением \*.test

При этом обработчик будет считывать контент файла и преобразовывать его в HTML разметку

</p>

<a href="File1.test">File1.test</a><br />

<a href="File2.test">File2.test</a><br />

<a href="File3.test">File3.test</a><br />

</div>

</form>

</body>

File1.test

<?xml version="1.0"?>

<UserList>

<User>

<FirstName>Olga</FirstName>

<LastName>Hansen</LastName>

</User>

<User>

<FirstName>Tove</FirstName>

<LastName>Svendson</LastName>

</User>

<User>

<FirstName>Kari</FirstName>

<LastName>Pettersen</LastName>

</User>

</UserList>

WebConfig

<configuration>

<system.web>

<compilation debug="true" targetFramework="4.5" />

<httpRuntime targetFramework="4.5" />

</system.web>

<system.webServer>

<handlers>

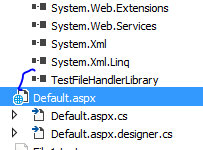
<add name="TestFileHandler" path="\*.test" verb="GET" type="TestFileHandlerLibrary.TestFileHandler, TestFileHandlerLibrary" />

</handlers>

</system.webServer>

</configuration>

Зборка



public class TestFileHandler : IHttpHandler

{

public bool IsReusable

{

get { return false; }

}

public void ProcessRequest(HttpContext context)

{

// получение физического пути к запрашиваемому файлу

string filePath = context.Request.PhysicalPath;

// Открываем файл на чтение.

using (FileStream stream = new FileStream(filePath, FileMode.Open))

{

// Загружаем контент файла.

XDocument document = XDocument.Load(stream);

// Выбираем элементы Users и извлекаем вложенные элементы FirstName и LastName.

var allUsers = from user in document.Descendants("User")

select new {

FirstName = user.Element("FirstName").Value,

LastName = user.Element("LastName").Value

};

// На основе полученных данных строим таблицу.

context.Response.Write("<html><body><table border='1'>");

foreach (var user in allUsers)

{

context.Response.Write("<tr>");

context.Response.Write("<td>" + user.FirstName + "</td>");

context.Response.Write("<td>" + user.LastName + "</td>");

context.Response.Write("</tr>");

}

context.Response.Write("</table></body></html>");

}

}

\*\*\*

AXD Files

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>Использование AXD файлов</title>

<style>

strong {

color:red;

}

</style>

</head>

<body>

<form id="form1" runat="server">

<div>

<a href="/Images/">Images/</a>

<br />

<a href="/Images/view.axd">Images/<strong>view.axd</strong></a>

<br />

<a href="/Images/Subfolder/view.axd">Images/Subfolder/<strong>view.axd</strong></a>

</div>

</form>

</body>

</html>

Web config

<configuration>

<system.web>

<compilation debug="true" targetFramework="4.5" />

<httpRuntime targetFramework="4.5" />

<trace enabled="true"/>

</system.web>

<system.webServer>

<handlers>

<add name="view" verb="\*" path="view.axd" type="AXDFileSample.PictureViewHandler"/>

</handlers>

</system.webServer>

</configuration>

PictureviewHandler

/// <summary>

/// Этот обработчик будет срабатывать, когда производиться запрос к файлу view.axd. (самого axd файла может и не быть в каталоге).

/// При этом, обработчик возвращает html страницу, со всеми изображениями, которые находятся в каталоге, где расположен axd файл.

///

/// В ASP.NET файлы c расширением axd обычно используют для конечных точек, указывающих на определенные сервисы. Например trace.axd - для просмотра данных по трассировке.

/// </summary>

public class PictureViewHandler : IHttpHandler

{

public bool IsReusable

{

get { return false; }

}

public void ProcessRequest(HttpContext context)

{

context.Response.Write("<html><head><title>Picture Viewer</title></head>");

context.Response.Write("<body>");

context.Response.Write(GetPageConent(context));

context.Response.Write("</body></html>");

}

private string GetPageConent(HttpContext context)

{

// Получение физического пути к запрашиваемому файлу.

string path = context.Request.PhysicalPath;

// Получаем список всех файлов по запрашиваемому пути.

List<FileInfo> files = GetAllImages(path);

StringBuilder htmlBuilder = new StringBuilder();

// Формирование списка изображений.

foreach (FileInfo file in files)

{

htmlBuilder.AppendFormat("<img src='{0}' alt='this is image' /><br /><br />", file.Name);

}

return htmlBuilder.ToString();

}

private List<FileInfo> GetAllImages(string path)

{

// Список найденных файлов.

List<FileInfo> images = new List<FileInfo>();

// Список допустимых расширений файлов.

List<string> extensions = new List<string>() { "\*.bmp", "\*.jpg", "\*.png", "\*.gif" };

// Убираем имя файла из пути.

string folderPath = path.Replace("\\view.axd", string.Empty);

DirectoryInfo dir = new DirectoryInfo(folderPath);

foreach (string ext in extensions)

{

FileInfo[] files = dir.GetFiles(ext);

if (files.Length > 0)

{

images.AddRange(files);

}

}

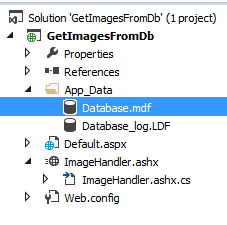
return images;

}

}

\*\*\*

Вытащить картинку з бази



<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<img src="ImageHandler.ashx?image=1" alt="Image" />

<img src="ImageHandler.ashx?image=2" alt="Image" />

<img src="ImageHandler.ashx?image=3" alt="Image" />

</div>

</form>

</body>

</html>

ImageHandler.ashx

public class ImageHandler : IHttpHandler

{

public void ProcessRequest(HttpContext context)

{

// Устанавливаем тип ответа.

context.Response.ContentType = "image/jpeg";

try

{

ReadImage(context);

}

catch

{

context.Response.End();

}

}

private static void ReadImage(HttpContext context)

{

// Получение GET параметра.

string imageId = context.Request.QueryString["image"];

string connectionString = ConfigurationManager.ConnectionStrings["local"].ConnectionString;

// Подключение к базе данных.

using (SqlConnection connection = new SqlConnection(connectionString))

{

// Извлечение фотографии по ID.

SqlCommand command = new SqlCommand("SELECT Image FROM Images WHERE ImageID=@ID", connection);

command.Parameters.AddWithValue("ID", imageId);

connection.Open();

byte[] image = command.ExecuteScalar() as byte[];

context.Response.OutputStream.Write(image, 0, image.Length);

connection.Close();

}

}

public bool IsReusable

{

get

{

return false;

}

}

}

WebConfig

<configuration>

<connectionStrings>

<add name="local" connectionString="Data Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|\Database.mdf;Integrated Security=True" />

</connectionStrings>

<system.web>

<compilation debug="true" targetFramework="4.5" />

<httpRuntime targetFramework="4.5" />

</system.web>

</configuration>

\*\*\*

Http Module

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<a href="Page1.aspx">Page1.aspx</a><br />

<a href="Page2.aspx">Page2.aspx</a>

</div>

</form>

</body>

</html>

Cs class

public class PerfomanceCounterModule : IHttpModule

{

private Stopwatch sw;

public void Dispose()

{

}

public void Init(HttpApplication context)

{

context.BeginRequest += context\_BeginRequest;

context.EndRequest += context\_EndRequest;

}

void context\_BeginRequest(object sender, EventArgs e)

{

sw = Stopwatch.StartNew();

}

void context\_EndRequest(object sender, EventArgs e)

{

sw.Stop();

HttpApplication app = (HttpApplication)sender;

app.Response.Write(@"----------------------------------------------------------------------<br/>

Время потраченное на обработку запроса " + sw.Elapsed);

}

}

}

Web config

<?xml version="1.0"?>

<!--

For more information on how to configure your ASP.NET application, please visit

http://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

<system.web>

<compilation debug="true" targetFramework="4.5" />

<httpRuntime targetFramework="4.5" />

</system.web>

<system.webServer>

<modules>

<add name="PerfomanceCouter" type="\_002\_HttpModule.PerfomanceCounterModule"/>

</modules>

</system.webServer>

</configuration>