Handler

public class Handler : IHttpHandler

{

public void ProcessRequest(HttpContext context)

{

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

context.Response.Write(DateTime.Now.ToLongTimeString());

}

public bool IsReusable

{

get

{

return true;

}

}

}

<head runat="server">

<title>AJAX использование через JavaScript</title>

<script type="text/javascript">

var xhr;

// обработчик события загрузки страницы.

window.onload = function () {

// Создание объекта запроса.

xhr = new XMLHttpRequest();

// функция-обработчик срабатывает при изменении свойства readyState

// Значения свойства readyState:

// 0 - Метод open() еще не вызывался

// 1 - Метод open() уже был вызван, но метод send() еще не вызывался.

// 2 - Метод send() был вызван, но ответ от сервера еще не получен

// 3 - Идет прием данных от сервера. Для значения 3 Firefox вызывает обработчик события несколько раз IE только один раз.

// 4 - Ответ от сервера полностью получен (Запрос успешно завершен).

xhr.onreadystatechange = function () {

if (xhr.readyState == 4) {

if (xhr.status == 200) {

document.getElementById("output").innerHTML = xhr.responseText;

}

}

}

var button1 = document.getElementById("button1");

button1.onclick = clickHandler;

}

// обработчик нажатия кнопки.

function clickHandler() {

// Настройка объекта на отправку асинхронного запроса.

xhr.open("GET", "Handler.ashx");

xhr.send(null);

}

</script>

</head>

<body>

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

<div>

<input type="button" value="AJAX кнопка" id="button1" />

<p id="output" style="color: Green; font-weight: bold;">

</p>

</div>

</form>

</body>

\*\*\*

WebService

Add -> webservice.asmx

Default.aspx

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="AJAX\_Basics\_ASP.Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head id="Head1" runat="server">

<title>AJAX использование в ASP.NET</title>

<script type="text/javascript">

window.onload = function () {

// Вызываем метод HelloWorld на сервисе WebService1

// Первый параметр - callback функция, которая запустится в случае успешной обработки запроса

// Второй параметр - callback функция, которая запустится в случае ошибки на сервере

AJAX\_Basics\_ASP.WebService1.HelloWorld(onComplete, onError);

}

function onComplete(result) {

alert(result);

}

function onError(error) {

alert(error.\_message);

}

</script>

</head>

<body>

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

<div>

<%--ScriptManager генерирует ссылки на библиотеки ASP.NET AJAX JavaScript--%>

<asp:ScriptManager ID="ScriptManager1" runat="server">

<%--Определяем место расположение службы ссылку на прокси JavaScrip код которой будет добавлена ScriptManager'ом--%>

<Services>

<asp:ServiceReference Path="~/WebService1.asmx" />

</Services>

</asp:ScriptManager>

</div>

</form>

</body>

</html>

Service.asxm

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.

[System.Web.Script.Services.ScriptService] // Разрешает вызовы JSON из клиентов.

public class WebService1 : System.Web.Services.WebService

{

[WebMethod] // Атрибут разрешает удаленный вызов метода.

public string HelloWorld()

{

return "Hello World";

}

}

}

\*\*\*

WevService

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="AJAX\_Basics\_ASP2.Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

<head runat="server">

<title>AJAX использование в ASP.NET</title>

<script type="text/javascript">

function search() {

// $get - псевдоним для функции document.getElementById() из библиотеки ASP.NET AJAX

var text = $get("textBoxName").value;

AJAX\_Basics\_ASP2.WebService1.FindCustomer(text, onComplete, onError)

}

function onComplete(result) {

if (result) {

$get("firstName").innerHTML = "First Name - " + result.FirstName;

$get("lastName").innerHTML = "Last Name - " + result.LastName;

$get("age").innerHTML = "Age - " + result.Age;

}

else {

alert("Поиск не дал результатов");

}

}

function onError(error) {

alert("ERROR:" + error.\_message);

}

</script>

</head>

<body>

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

<div>

<%--ScriptManager генерирует ссылки на библиотеки ASP.NET AJAX JavaScript--%>

<asp:ScriptManager runat="server">

<%--Определяем место расположение службы ссылку на прокси JavaScrip код которой будет добавлена ScriptManager'ом--%>

<Services>

<asp:ServiceReference Path="~/WebService1.asmx" />

</Services>

</asp:ScriptManager>

Введите имя или фамилию для поиска

<input type="text" id="textBoxName" />

<br />

<input type="button" value="Поиск" onclick="search()" />

<br />

<p id="firstName"></p>

<p id="lastName"></p>

<p id="age"></p>

</div>

</form>

</body>

</html>

Service.asxm

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

[System.Web.Script.Services.ScriptService]

public class WebService1 : System.Web.Services.WebService

{

[WebMethod]

public Customer FindCustomer(string text)

{

return \_customers.Where(x => x.FirstName == text || x.LastName == text).FirstOrDefault();

}

private List<Customer> \_customers = new List<Customer>();

public WebService1()

{

\_customers.Add(new Customer() { FirstName = "Jhon", LastName = "Doe", Age=27 });

\_customers.Add(new Customer() { FirstName = "Tom", LastName = "Ronald", Age=24 });

\_customers.Add(new Customer() { FirstName = "Jane", LastName = "Roe", Age=32 });

}

}

public class Customer

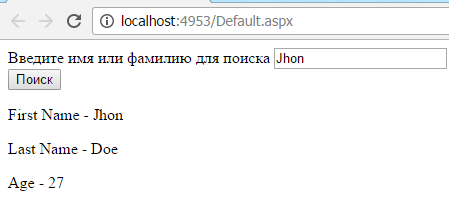
{

public string FirstName { get; set; }

public string LastName { get; set; }

public int Age { get; set; }

}



\*\*\*

SeverControls

UpdatePanel

</head>

<body>

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

<div>

<asp:ScriptManager runat="server" />

<%--UpdatePanel - задает область страницы, которая будет обновляться асинхронно--%>

<asp:UpdatePanel ID="UpdatePanel1" runat="server">

<ContentTemplate>

<asp:Label runat="server" BackColor="Red" ForeColor="White" ID="Label1" />

<asp:Button ID="Button1" Text="Обновить таймер №1" runat="server" OnClick="Button1\_Click" />

</ContentTemplate>

</asp:UpdatePanel>

<br />

<br />

<br />

<asp:UpdatePanel runat="server">

<ContentTemplate>

<asp:Label runat="server" BackColor="Green" ForeColor="Yellow" ID="Label2" />

<asp:Button ID="Button2" Text="Обновить таймер №2" runat="server" OnClick="Button2\_Click" />

</ContentTemplate>

</asp:UpdatePanel>

<br />

<br />

<br />

<asp:UpdatePanel ID="UpdatePanel2" runat="server">

<ContentTemplate>

<asp:Label runat="server" BackColor="Blue" ForeColor="White" ID="Label3" />

<asp:Button ID="Button3" Text="Обновить таймер №3" runat="server" OnClick="Button3\_Click" />

</ContentTemplate>

</asp:UpdatePanel>

</div>

</form>

</body>

public partial class \_01\_UpdatePanel : System.Web.UI.Page

{

protected void Button1\_Click(object sender, EventArgs e)

{

Label1.Text = DateTime.Now.ToLongTimeString();

}

protected void Button2\_Click(object sender, EventArgs e)

{

Label2.Text = DateTime.Now.ToLongTimeString();

}

protected void Button3\_Click(object sender, EventArgs e)

{

Label3.Text = DateTime.Now.ToLongTimeString();

}

}

\*\*\*

UpdatePanelError

<head runat="server">

<title>UpdatePanel ExceptionSample #2</title>

<script type="text/javascript">

// метод pageLoad вызывается автоматически если присутствует в сценарии

function pageLoad() {

var pageManager = Sys.WebForms.PageRequestManager.getInstance();

// Событие endRequest происходит по завершению каждой асинхронной отправки запроса.

pageManager.add\_endRequest(endRequest);

}

// Обработчик, который проверяет наличие ошибки, отображает ее, если она есть, и устанавливает статус ошибки, как обработанный.

function endRequest(sender, args) {

if (args.get\_error() != null) {

$get("ErrorOutput").innerHTML = args.get\_error().message;

args.set\_errorHandled(true);

}

}

</script>

</head>

<body>

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

<div>

<asp:ScriptManager ID="ScriptManager1" runat="server" />

<asp:UpdatePanel ID="UpdatePanel1" runat="server">

<ContentTemplate>

<asp:Button ID="Button1" Text="Обновить" runat="server" />

</ContentTemplate>

</asp:UpdatePanel>

<span id="ErrorOutput" style="background-color:Red; color:White;"></span>

</div>

</form>

</body>

protected void Page\_Load(object sender, EventArgs e)

{

if (IsPostBack)

{

throw new ApplicationException("Обновление страницы потерпело неудачу!");

}

}

\*\*\*

ConditionalUpdate

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

<div>

<asp:ScriptManager ID="ScriptManager1" runat="server" />

<%--UpdatePanel - задает область страницы, которая будет обновляться асинхронно--%>

<%--Always - панель обновляется на любую обратную отправку, которая выполняется любым контролом на странице--%>

<asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Always">

<ContentTemplate>

<asp:Label runat="server" BackColor="Red" ForeColor="White" ID="Label1" />

<asp:Button ID="Button1" Text="Обновить таймер" runat="server" />

</ContentTemplate>

</asp:UpdatePanel>

<br />

<br />

<br />

<%--Conditional - обновление панели происходит в следующих случаях:

1. Был явно вызван метод Update для UpdatePanel

2. Сработал один из триггеров UpadtePanel

3. Элемент управления находящийся в UpdatePanel выполнил обратную отправку

--%>

<asp:UpdatePanel ID="UpdatePanel2" runat="server" UpdateMode="Conditional">

<ContentTemplate>

<asp:Label runat="server" BackColor="Green" ForeColor="Yellow" ID="Label2" />

<asp:Button ID="Button2" Text="Обновить таймер" runat="server" />

</ContentTemplate>

</asp:UpdatePanel>

<br />

<br />

<br />

<asp:UpdatePanel ID="UpdatePanel3" runat="server">

<ContentTemplate>

<asp:Label runat="server" BackColor="Blue" ForeColor="White" ID="Label3" />

<asp:Button ID="Button3" Text="Обновить таймер" runat="server" />

</ContentTemplate>

</asp:UpdatePanel>

</div>

</form> protected void Page\_Load(object sender, EventArgs e)

{

Label1.Text = DateTime.Now.ToLongTimeString();

Label2.Text = DateTime.Now.ToLongTimeString();

Label3.Text = DateTime.Now.ToLongTimeString();

}

}

\*\*\*

UpdatePanelTrigger

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

<div>

<asp:ScriptManager runat="server" />

<table border="0" cellpadding="0" cellspacing="0">

<tr>

<td>

<asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Conditional">

<ContentTemplate>

<%=DateTime.Now.ToLongTimeString() %>

</ContentTemplate>

<Triggers>

<asp:AsyncPostBackTrigger ControlID="Button1" />

</Triggers>

</asp:UpdatePanel>

</td>

<td>

<asp:UpdatePanel ID="UpdatePanel2" runat="server" UpdateMode="Conditional">

<ContentTemplate>

<%=DateTime.Now.ToLongTimeString() %>

</ContentTemplate>

<Triggers>

<asp:AsyncPostBackTrigger ControlID="Button2" />

</Triggers>

</asp:UpdatePanel>

</td>

<td>

<asp:UpdatePanel ID="UpdatePanel3" runat="server" UpdateMode="Conditional">

<ContentTemplate>

<%=DateTime.Now.ToLongTimeString() %>

</ContentTemplate>

<Triggers>

<asp:AsyncPostBackTrigger ControlID="Button3" />

</Triggers>

</asp:UpdatePanel>

</td>

</tr>

<tr>

<td>

<asp:Button ID="Button1" Text="Обновить 1" runat="server" />

</td>

<td>

<asp:Button ID="Button2" Text="Обновить 2" runat="server" />

</td>

<td>

<asp:Button ID="Button3" Text="Обновить 3" runat="server" />

</td>

</tr>

</table>

</div>

</form>

TimerSample

<body>

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

<div>

<asp:ScriptManager runat="server" />

<asp:UpdatePanel runat="server">

<ContentTemplate>

<%--Элемент управления, выполняющий обратный запрос с определенной периодичностью--%>

<asp:Timer ID="Timer1" runat="server" Interval="1000"></asp:Timer>

<%= DateTime.Now.ToLongTimeString() %>

</ContentTemplate>

</asp:UpdatePanel>

</div>

</form>

</body>

\*\*\*

TimerSimple

<body>

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

<div>

<asp:ScriptManager runat="server" />

<asp:UpdatePanel runat="server">

<ContentTemplate>

<asp:Timer ID="Timer1" runat="server" OnTick="Timer\_Tick" Interval="1000">

</asp:Timer>

<asp:Label Text="0" ID="Label1" runat="server" />

</ContentTemplate>

</asp:UpdatePanel>

</div>

</form>

</body>

public int Counter

{

get

{

object val = ViewState["Counter"];

if (val != null)

{

return (int)val;

}

ViewState["Counter"] = 0;

return 0;

}

set { ViewState["Counter"] = value; }

}

protected void Timer\_Tick(object sender, EventArgs e)

{

Counter++;

Label1.Text = Counter.ToString();

}

\*\*\*

UpdateProgress

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

<div>

<asp:ScriptManager runat="server" />

<asp:UpdatePanel ID="UpdatePanel1" runat="server" UpdateMode="Conditional">

<ContentTemplate>

<asp:Label ID="Label1" Text="Date: " runat="server" />

</ContentTemplate>

<Triggers>

<asp:AsyncPostBackTrigger ControlID="Button1" />

</Triggers>

</asp:UpdatePanel>

<br />

<%--

Данный блок будет становиться видимым в момент выполнения асинхронного обратного запроса.

Для того, что бы блок был видимым только при обновлении конкретного UpdatePanel следует установить свойтсов AssociatedUpdatePanelID

--%>

<asp:UpdateProgress runat="server">

<ProgressTemplate>

<img src="ajax-loader.gif" />

</ProgressTemplate>

</asp:UpdateProgress>

<br />

<asp:Button ID="Button1" Text="Обновить" runat="server" OnClick="Button\_Click" />

</div>

</form>

protected void Button\_Click(object sender, EventArgs e)

{

Thread.Sleep(5000);

Label1.Text = "Date: " + DateTime.Now.ToLongTimeString();

}