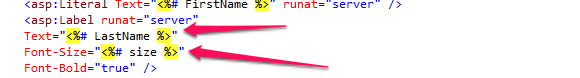
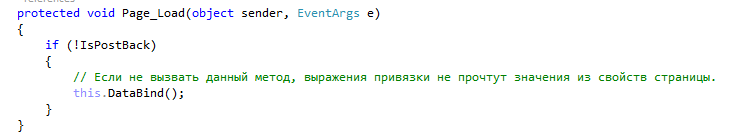
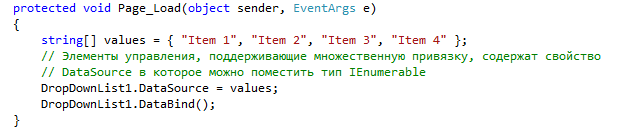
Выражения <%# %> позволяют привязать свойства элементов управления к защищенным или открытым полям и свойствам текущей страницы. Для того что бы фактически произвести привязку и извлечь значения из свойств, нужно вызвать метод DataBind();





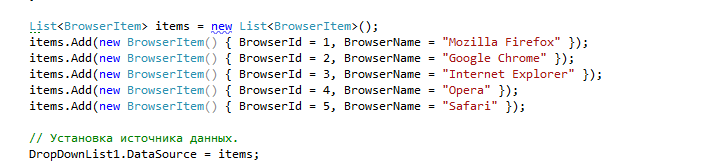


Привязка коллекции





Привязать собственный класс



// Имя свойства одного элемента из источника данных, которое будет использоваться для

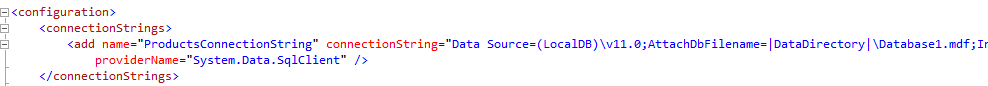
// отображения названия пункта в элементе управления.

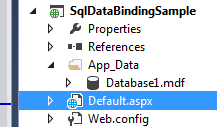
DropDownList1.DataTextField = "BrowserName";

// Имя свойства для Value элемента в списке.

DropDownList1.DataValueField = "BrowserId";

Привязываем базу данных





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

<div>

<table>

<tr>

<td>Продукт</td>

<td>

<%--В атрибутах DataTextField и DataValueField указаны имена колонок базы данных.--%>

<asp:DropDownList ID="DropDownList1" runat="server"

DataSourceID="ProductsSqlDataSource"

DataTextField="Name"

DataValueField="Id">

</asp:DropDownList>

<%--Источник данных--%>

<asp:SqlDataSource ID="ProductsSqlDataSource"

runat="server"

ConnectionString="<%$ ConnectionStrings:ProductsConnectionString %>"

SelectCommand="SELECT [Id], [Name] FROM [Products]"></asp:SqlDataSource>

</td>

</tr>

<tr>

<td>Адрес доставки</td>

<td>

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

</td>

</tr>

<tr>

<td>&nbsp;</td>

<td>

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

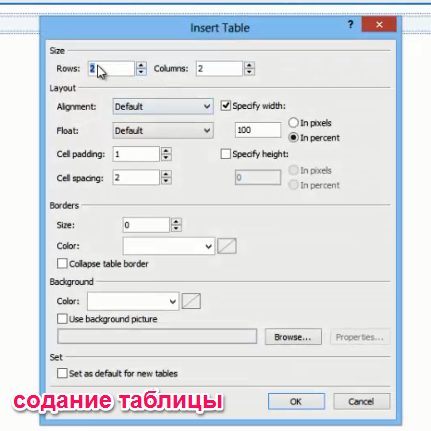
</td>

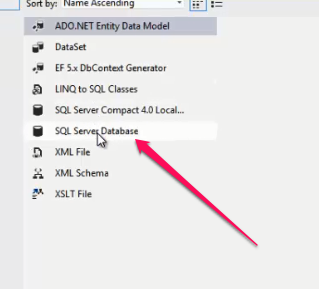
</tr>

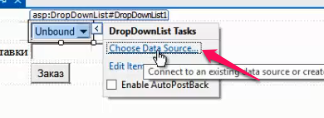
</table>

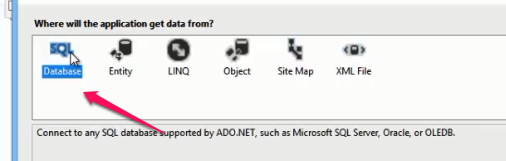
</div>

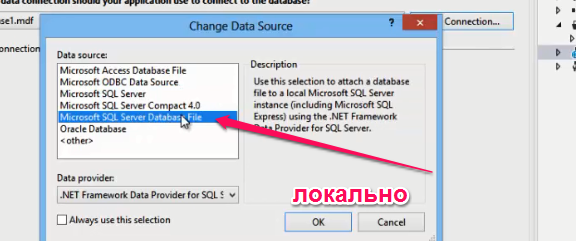
</form>

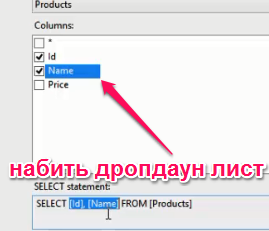


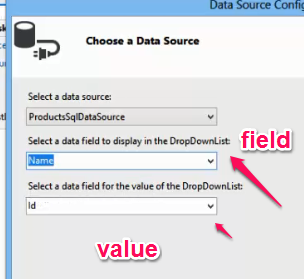




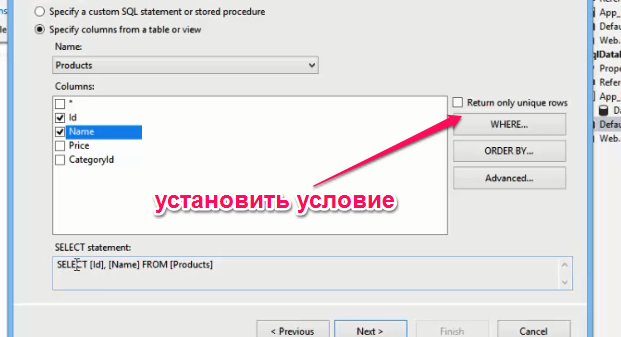


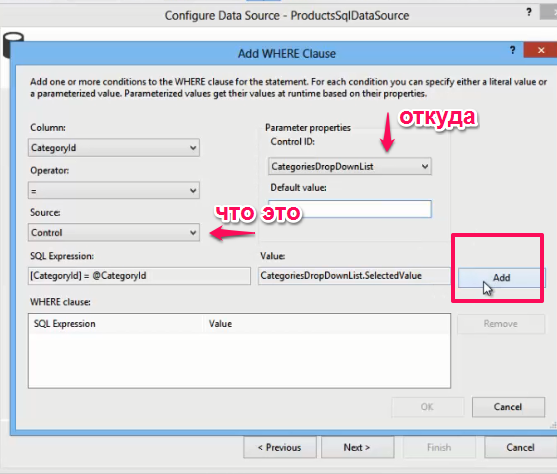




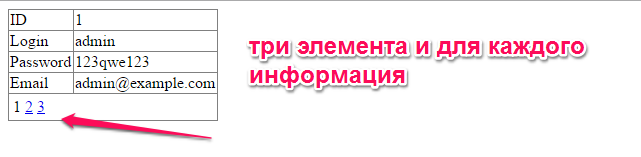








<asp:DetailsView



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

<div>

<asp:DetailsView ID="DetailsView1" runat="server" Height="50px" Width="125px" AutoGenerateRows="False"

DataKeyNames="ID" Получает или задает массив, содержащий имена ключевых полей для источника данных.

DataSourceID="SqlDataSource1"

AllowPaging="True" //возможность перелистывать элементы>

<%--Каждый элемент BoundField привязывает элемент разметки к данным из источника данных--%>

<Fields>

<asp:BoundField DataField="ID" HeaderText="ID" InsertVisible="False" ReadOnly="True" />

<asp:BoundField DataField="Login" HeaderText="Login" />

<asp:BoundField DataField="Password" HeaderText="Password" />

<asp:BoundField DataField="Email" HeaderText="Email" />

</Fields>

</asp:DetailsView>

<%--строка подключения извлекается из файла web.config--%>

<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:DatabaseConnection %>"

SelectCommand="SELECT \* FROM [Users]"></asp:SqlDataSource>

</div>

</form>

С возможностью редактирования

Делается с помощью расширения SqlDataSource

<asp:SqlDataSource ID="SqlDataSource1" runat="server"

ConnectionString="<%$ ConnectionStrings:DatabaseConnection %>"

DeleteCommand="DELETE FROM [Users] WHERE [ID] = @ID"

InsertCommand="INSERT INTO [Users] ([Login], [Password], [Email]) VALUES (@Login, @Password, @Email)"

SelectCommand="SELECT \* FROM [Users]"

UpdateCommand="UPDATE [Users] SET [Login] = @Login, [Password] = @Password, [Email] = @Email WHERE [ID] = @ID">

<DeleteParameters>

<asp:Parameter Name="ID" Type="Int32" />

</DeleteParameters>

<InsertParameters>

<asp:Parameter Name="Login" Type="String" />

<asp:Parameter Name="Password" Type="String" />

<asp:Parameter Name="Email" Type="String" />

</InsertParameters>

<UpdateParameters>

<asp:Parameter Name="Login" Type="String" />

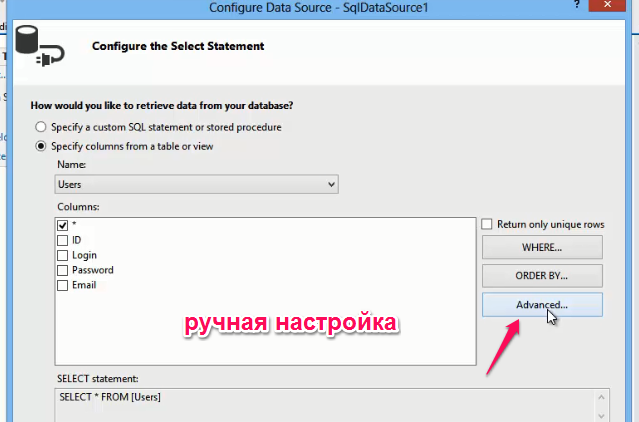
<asp:Parameter Name="Password" Type="String" />

<asp:Parameter Name="Email" Type="String" />

<asp:Parameter Name="ID" Type="Int32" />

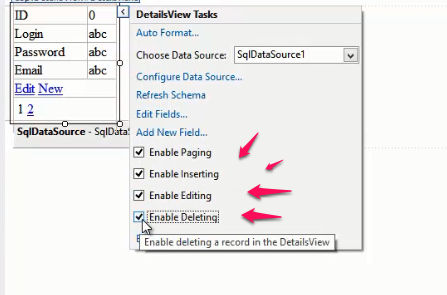
</UpdateParameters>

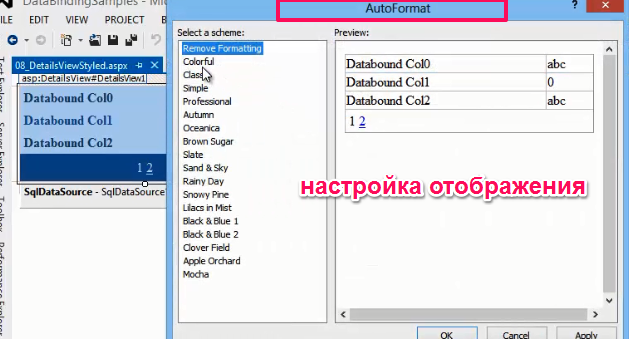
</asp:SqlDataSource>

Только обязательно должен быть первичный ключ







Пользовательский источник данных

/// <summary>

/// Класс представляющий источник данных для элементов управления на страницах

/// </summary>

public class ProductsDatabase

{

private static List<Product> \_products;

public static List<Product> Products

{

get

{

if (\_products == null)

{

\_products = new List<Product>();

\_products.Add(new Product() { Id = 1, Name = "Mobile phone", Price = 220 });

\_products.Add(new Product() { Id = 2, Name = "Laptop", Price = 999 });

}

return \_products;

}

set { \_products = value; }

}

// Получение всех элементов из источника данных

public List<Product> SelectAllProducts()

{

return Products;

}

// Добавление нового элемента в источник данных.

public void AddProduct(Product p)

{

Products.Add(p);

}

Переопределили метод Equals чтобы находл элементы по ID

// Удаление элемента из источника данных.

public void RemoveProduct(Product p)

{

Products.Remove(p);

}

// Обновление элемента в источнике данных.

public void UpdateProduct(Product p)

{

int index = Products.IndexOf(p);

Products.Remove(p);

Products.Insert(index, p);

}

}

/// <summary>

/// Класс представляющий одну запись в источнике данных.

/// </summary>

public class Product

{

public int Id { get; set; }

public string Name { get; set; }

public double Price { get; set; }

public Product()

{

}

public override bool Equals(object obj)

{

Product temp = obj as Product;

if (temp == null)

{

return false;

}

return temp.Id == this.Id;

}

// При переопределении метода Equals метод GetHashCode желательно переопределить.

public override int GetHashCode()

{

return this.Id;

}

}

Подключение

<body>

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

<div>

<asp:DetailsView ID="DetailsView1" runat="server" Height="50px" Width="125px" DataKeyNames="ID"

AllowPaging="True" AutoGenerateRows="False" DataSourceID="ObjectDataSource1">

<Fields>

<asp:BoundField DataField="Id" HeaderText="Id" />

<asp:BoundField DataField="Name" HeaderText="Name" />

<asp:BoundField DataField="Price" HeaderText="Price" />

<asp:CommandField ShowDeleteButton="True" ShowEditButton="True" ShowInsertButton="True" />

</Fields>

</asp:DetailsView>

<!--Представляет бизнес объект, представляющий данные для элемента управления.-->

<asp:ObjectDataSource ID="ObjectDataSource1" runat="server"

DataObjectTypeName="Product"

TypeName="ProductsDatabase"

DeleteMethod="RemoveProduct"

InsertMethod="AddProduct"

SelectMethod="SelectAllProducts"

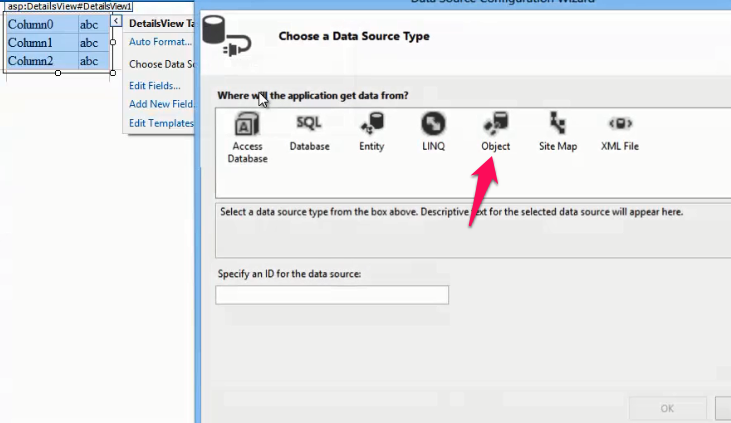
UpdateMethod="UpdateProduct"></asp:ObjectDataSource>

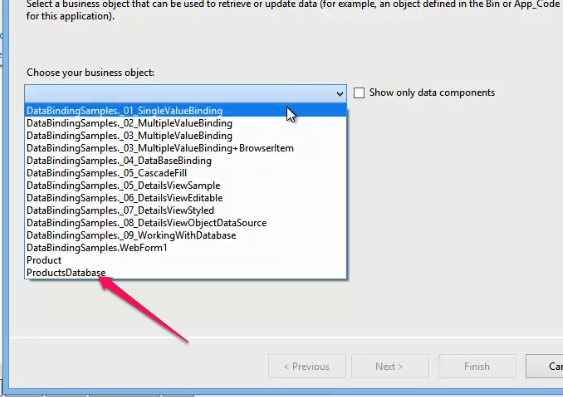
</div>

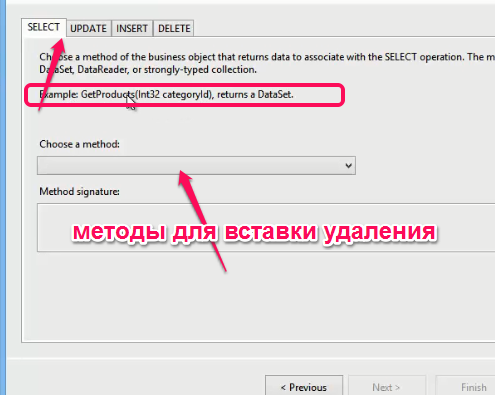
</form>

</body>

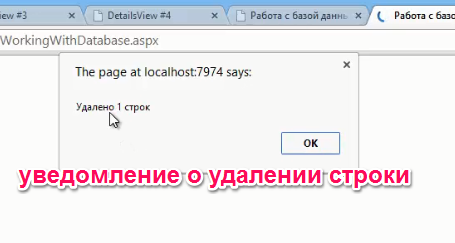
**Автоматизация**







Добавление в коде о уведомлении на какое либо действие



// Добавление в ответ пользователю тега скрипт, в котором с помощью javascript функции alert выводиться сообщение.

string script = string.Format("alert('Удалено {0} строк');", affectedRows);

для какой странице

уникальное имя

текст

то что должен быть помещен скрипт в блок скриптов, а не просто на страницу (между тегами скрипт)

ClientScript.RegisterClientScriptBlock(this.GetType(), "MessageBox", script, true);

