

C# и .NET

Web

Форум

Найти..



C# 5.0 И .NET 4.5

WPF

ТЕМЫ WPF

SILVERLIGHT 5

EXPRESSION BLEND 4

РАБОТА С БД

LINQ

ASP.NET

WINDOWS 8/10

Админ панель: управление каталогом

[ASP.NET](#) --- [Интернет магазин на ASP.NET Web Forms](#) --- [Админ панель: управление каталогом](#)



Исходный код проекта

В этой статье мы добавим для администратора поддержку управления контентом каталога товаров. Соглашение по управлению коллекцией элементов предусматривает отображение списка этих элементов, а также предоставление пользователям набора полей ввода, с помощью которых можно редактировать существующие элементы и вставлять новые.

Для такой функциональности мы должны обеспечить возможность создания, чтения, обновления и

удаления элементов в хранилище данных. Все вместе эти действия называются **CRUD** (**creating, reading, updating, deleting** - **создание, чтение, обновление, удаление**), и необходимость в операциях CRUD для приложений возникает настолько часто, что в приложениях ASP.NET Framework имеются развитые средства, упрощающие реализацию упомянутых действий. В этом разделе мы продемонстрируем одно такое средство при построении функции управления каталогом GameStore - элемент управления ListView.

Расширение хранилища

Перед тем как приступить к добавлению функциональности веб-формы, понадобится расширить класс хранилища, чтобы можно было добавлять, модифицировать и удалять записи о товарах в базе данных. В примере ниже показано определение методов SaveGame() и DeleteGame(), которые добавлены в класс Repository из файла \Models\Repository\Repository.cs:

```
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;

namespace GameStore.Models.Repository
{
    public class Repository
    {
        private EFDbContext context = new EFDbContext();

        public IEnumerable<Game> Games
        {
            // ...
        }

        public IEnumerable<Order> Orders
        {
            // ...
        }

        public void SaveGame(Game game)
        {
            if (game.GameId == 0)
            {
                game = context.Games.Add(game);
            }
            else
            {
                Game dbGame = context.Games.Find(game.GameId);
                if (dbGame != null)
                {
                    dbGame.Name = game.Name;
                    dbGame.Description = game.Description;
                    dbGame.Price = game.Price;
                    dbGame.Category = game.Category;
                }
            }
            context.SaveChanges();
        }
    }
}
```

Удаление Game будет приводить к удалению из базы данных всех Order, которые включают этот Game. Мы не можем удалить только Game, потому что это вызовет ошибку ссылочной целостности на сервере базы данных. Сложный запрос LINQ в методе DeleteGame() обеспечивает удаление из таблиц Orders и OrderLines тех строк, которые зависят от удаляемого Game, позволяя поддерживать целостность базы данных.

Объекты Game, для которых нет соответствующих строк в базе данных, идентифицируются по нулевым значениям свойства GameID. В случае если объект Game имеет ненулевое значение в свойстве GameID, производится обновление существующей строки в базе данных.

Добавление веб-формы

Для поддержки средств управления товарами мы создали веб-форму \Pages\Admin\Games.aspx, воспользовавшись построенной ранее мастер-страницей \Pages\Admin\Admin.Master. Содержимое файла Games.aspx представлено в примере ниже:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Games.aspx.cs" Inherits
    MasterPageFile="~/Pages/Admin/Admin.Master" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" runat="serv
    <asp:ListView ID="ListView1" ItemType="GameStore.Models.Game" SelectMethod="G
        DataKeyNames="GameId" UpdateMethod="UpdateGame" DeleteMethod="DeleteGame"
        InsertMethod="InsertGame" InsertItemPosition="LastItem" EnableViewState="
        runat="server">
        <LayoutTemplate>
            <div class="outerContainer">
                <table id="productsTable">
                    <tr>
                        <th>Название игры</th>
                        <th>Описание</th>
                        <th>Категория</th>
                        <th>Цена</th>
                    </tr>
                    <tr runat="server" id="itemPlaceholder"></tr>
                </table>
            </div>
        </LayoutTemplate>
        <ItemTemplate>
            <tr>
                <td><%# Item.Name %></td>
                <td class="description"><span><%# Item.Description %></span></td>
```

Мы не собираемся вдаваться в детали работы элемента управления ListView, но вы можете получить представление об этом, взглянув на структуру разметки (более подробно он описан в статье ["Элемент управления ListView"](#)).

Обратите внимание, что ListView содержит несколько разных шаблонов. Эти шаблоны применяются для генерации смеси HTML-разметки и кода JavaScript, которая отображает объект данных и позволяет пользователю редактировать и удалять существующие элементы данных, а также создавать новые.

Шаблоны, которые мы определили, содержат обычные HTML-элементы и несколько элементов управления Button. Эти элементы управления используются для запуска различных CRUD-операций в ListView или для переключения между разными шаблонами. Элемент управления ListView обладает множеством функциональных средств, и его поведение можно настраивать для каждого аспекта управления данными. Разумеется, вы не обязаны пользоваться элементом управления ListView, и мы иногда в своих проектах предпочитаем писать код HTML и JavaScript самостоятельно, однако инфраструктура Web Forms содержит множество сложных

элементов управления, ориентированных на данные, таких как `ListView`, которые позволяют легко строить расширенную функциональность.

Создание методов CRUD

Элементу управления `ListView` ничего не известно о способе получения и сохранения объектов данных `Game`, поэтому мы должны реализовать в классе отделенного кода ряд методов, взаимодействующих с хранилищем. В примере ниже приведено содержимое файла `\Pages\Admin\Games.aspx.cs`:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using GameStore.Models;
using GameStore.Models.Repository;
using System.Web.ModelBinding;

namespace GameStore.Pages.Admin
{
    public partial class Games : System.Web.UI.Page
    {
        private Repository repository = new Repository();

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        public IEnumerable<Game> GetGames()
        {
            return repository.Games;
        }

        public void UpdateGame(int GameID)
        {
            Game myGame = repository.Games
                .Where(p => p.GameId == GameID).FirstOrDefault();
            if (myGame != null && TryUpdateModel(myGame,
                new FormValueProvider(ModelBindingExecutionContext)))
            {
                repository.SaveGame(myGame);
            }
        }

        public void DeleteGame(int GameID)
        {
            Game myGame = repository.Games
```

Легко понять, как эти методы отображаются на операции CRUD. Метод `GetGames()` читает объекты данных. Методы `InsertGame()`, `UpdateGame()` и `DeleteGame()` отвечают за создание, обновление и удаление объектов данных.

Мы сообщаем элементу управления `ListView` об этих методах с помощью последовательности атрибутов в веб-форме:

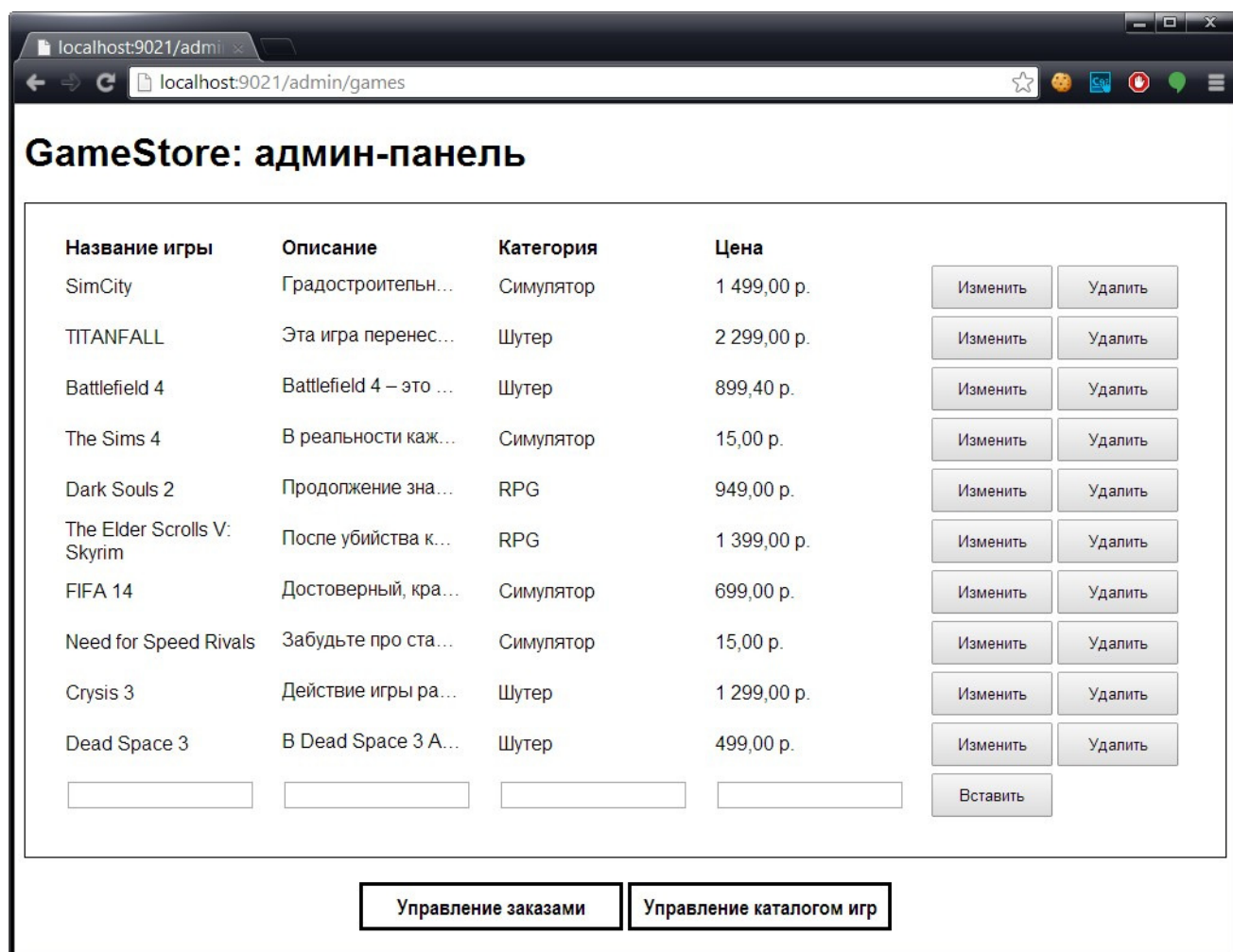
```
...  
<asp:ListView ID="ListView1"  
    SelectMethod="GetGames"  
    UpdateMethod="UpdateGame"  
    DeleteMethod="DeleteGame"  
    InsertMethod="InsertGame"  
    ...>  
...
```

Элемент управления `ListView` заботится о генерации кода HTML и JavaScript, требуемого для браузера, и обеспечивает вызов наших методов CRUD в ответ на действия, предпринимаемые пользователем. Нам не приходится иметь дело с отдельными запросами, анализом данных формы и прочими аспектами этого процесса.

Тестирование средств управления каталогом

Чтобы увидеть элемент управления `ListView` в работе, запустите приложение и перейдите на URL вида `/admin/games`. Отобразится список товаров из базы данных в форме сетки, как показано на рисунке ниже:

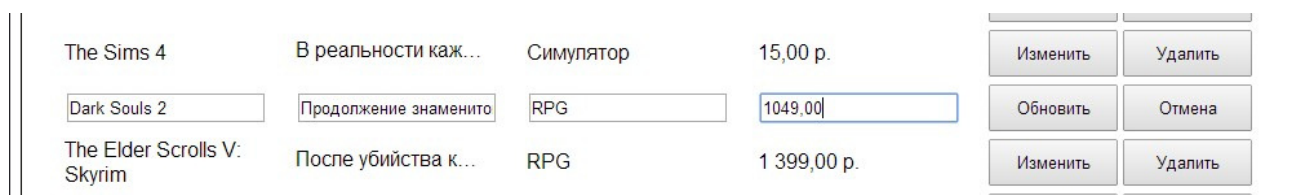




The screenshot shows a web browser window with the address bar displaying 'localhost:9021/admin/games'. The page title is 'GameStore: админ-панель'. It features a table with the following columns: 'Название игры' (Game Name), 'Описание' (Description), 'Категория' (Category), and 'Цена' (Price). Each row represents a game and includes 'Изменить' (Edit) and 'Удалить' (Delete) buttons. At the bottom of the table are four empty input fields and a 'Вставить' (Insert) button. Below the table are two buttons: 'Управление заказами' (Manage Orders) and 'Управление каталогом игр' (Manage Game Catalog).

Название игры	Описание	Категория	Цена		
SimCity	Градостроительн...	Симулятор	1 499,00 р.	Изменить	Удалить
TITANFALL	Эта игра перенес...	Шутер	2 299,00 р.	Изменить	Удалить
Battlefield 4	Battlefield 4 – это ...	Шутер	899,40 р.	Изменить	Удалить
The Sims 4	В реальности каж...	Симулятор	15,00 р.	Изменить	Удалить
Dark Souls 2	Продолжение зна...	RPG	949,00 р.	Изменить	Удалить
The Elder Scrolls V: Skyrim	После убийства к...	RPG	1 399,00 р.	Изменить	Удалить
FIFA 14	Достоверный, кра...	Симулятор	699,00 р.	Изменить	Удалить
Need for Speed Rivals	Забудьте про ста...	Симулятор	15,00 р.	Изменить	Удалить
Crysis 3	Действие игры ра...	Шутер	1 299,00 р.	Изменить	Удалить
Dead Space 3	В Dead Space 3 A...	Шутер	499,00 р.	Изменить	Удалить
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Вставить	

Каждая строка содержит подробные сведения об одном товаре, а также кнопки для редактирования и удаления товара. Щелчок на кнопке "Изменить" приводит к замене подробных сведений о товаре встроенным редактором:



The screenshot shows a form for editing a game. It has four input fields for 'The Sims 4': 'В реальности каж...', 'Симулятор', '15,00 р.', and a price field containing '1049,00'. To the right are 'Изменить' (Edit) and 'Удалить' (Delete) buttons. Below these are 'Обновить' (Update) and 'Отмена' (Cancel) buttons. At the bottom are 'Изменить' (Edit) and 'Удалить' (Delete) buttons.

The Sims 4	В реальности каж...	Симулятор	15,00 р.	Изменить	Удалить
<input type="text" value="Dark Souls 2"/>	<input type="text" value="Продолжение знаменито"/>	<input type="text" value="RPG"/>	<input type="text" value="1049,00"/>	Обновить	Отмена
The Elder Scrolls V: Skyrim	После убийства к...	RPG	1 399,00 р.	Изменить	Удалить

Щелчок на кнопке "Отмена" позволяет отменить внесенные изменения, а щелчок на кнопке "Обновить" — сохранить изменения в базе данных. Наконец, заполнив пустые поля редактора в нижней строке сетки и щелкнув на кнопке "Вставить", можно добавить новый элемент в базу данных.

Осторожно используйте кнопку "Удалить". Как упоминалось ранее, удаление товара вызывает удаление всех заказов, в состав которых входит этот товар.

При работе с данными вы сможете оценить как достоинства, так и недостатки применения элементов управления. С одной стороны, мы получаем действительно сложную функциональность быстро и легко, с минимальными усилиями. С другой стороны, мы отказываемся от контроля и понимания того, как работает приложение. В своих проектах мы склонны следовать подходу, который используется при

разработке приложена GameStore. Это значит, что мы берем на себя полный контроль над пользовательскими частями приложения и пишем довольно много кода самостоятельно, но применяем элементы управления вроде ListView для административных страниц, которые используются менее часто, поэтому им можно уделять меньше внимания.

.NET Excel



Excel R/W, Lightning Fast Upgrades & e-iceblue.com

Alexandr Erohin ★ alexerohinzzz@gmail.com © 2011 - 2016