

C# и .NET

Web

Форум

Найти..



C# 5.0 и .NET 4.5

WPF

ТЕМЫ WPF

SILVERLIGHT 5

EXPRESSION BLEND 4

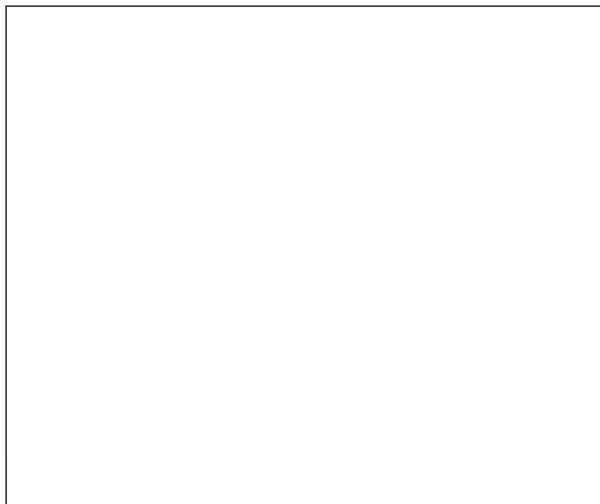
РАБОТА С БД

LINQ

ASP.NET

WINDOWS 8/10

Проект интернет-магазина

[ASP.NET](#) --- [Интернет магазин на ASP.NET Web Forms](#) --- [Проект интернет-магазина](#)

Исходный код проекта

В этой и следующих статьях мы создадим новый проект Visual Studio по имени GameStore, используя шаблон ASP.NET Empty Web Application (Пустое веб-приложение ASP.NET). Мы начнем с настройки модели данных и базы данных, после чего добавим другие необходимые средства.

Мы будем выделять каждый шаг при создании приложения, так что вы сможете видеть, как взаимодействуют друг с другом разнообразные средства, и постепенно

воспроизводить пример интернет-магазина на собственном компьютере.

Создание структуры папок

В простых примерах мы помещали все файлы приложения в корневую папку проекта, что нормально для простых приложений, но может усложнить управление крупными проектами. В этом проекте мы собираемся быть более организованными и следовать структуре, которую применяем в реальных проектах. В таблице ниже описаны все созданные папки и даны пояснения относительно типа контента, который будет помещен в каждую из них (ряд таких категорий будет объясняться при создании соответствующего контента).

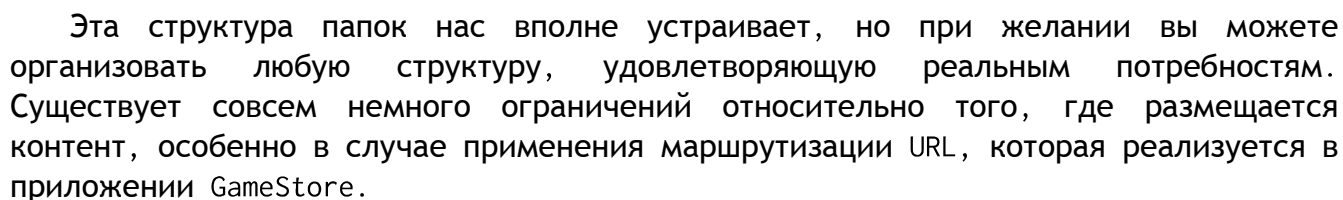
Назначение папок, добавленных в проект GameStore

| Имя папки | Описание |
|-------------------|---|
| App_Start | По соглашению эта папка содержит классы, которые выполняют начальное конфигурирование приложения при его запуске. Мы будем использовать эту папку во время конфигурирования маршрутизации URL и настройки подключаемых библиотек JavaScript |
| Content | По соглашению эта папка содержит статический контент, такой как файлы таблиц стилей CSS |
| Controls | Эта папка будет содержать пользовательские элементы управления |
| Models | Эта папка будет содержать классы моделей данных |
| Models\Repository | Эта папка будет содержать классы, которые используются для реализации постоянного хранилища для классов моделей данных |
| Pages | Эта папка будет содержать файлы веб-форм |
| Pages\Admin | Эта папка будет содержать файлы веб-форм для административных средств приложения GameStore |
| Pages\Helpers | Эта папка будет содержать вспомогательные классы, с которыми работают веб-формы |
| Scripts | Эта папка будет содержать JavaScript-файлы |

Создайте указанные папки, чтобы они были готовы к добавлению контента разных типов в этой и последующих статьях. После создания папок окно Solution Explorer (Проводник решения) должно выглядеть примерно так, как показано на

C# Read/Write Spreadsheet

No automation, Lightning
Fast, Easy Free Support,
Upgrades & Try!



Вас может интересовать, почему мы продолжаем использовать шаблон ASP.NET Empty Web Application (Пустое веб-приложение ASP.NET), хотя в списке доступен также шаблон ASP.NET Web Forms Application (Приложение ASP.NET Web Forms). В Visual Studio включен набор шаблонов, которые помогают начать процесс разработки за счет добавления файлов и функций, широко применяемых в проектах разных типов. Шаблон ASP.NET Web Forms Application предназначен для

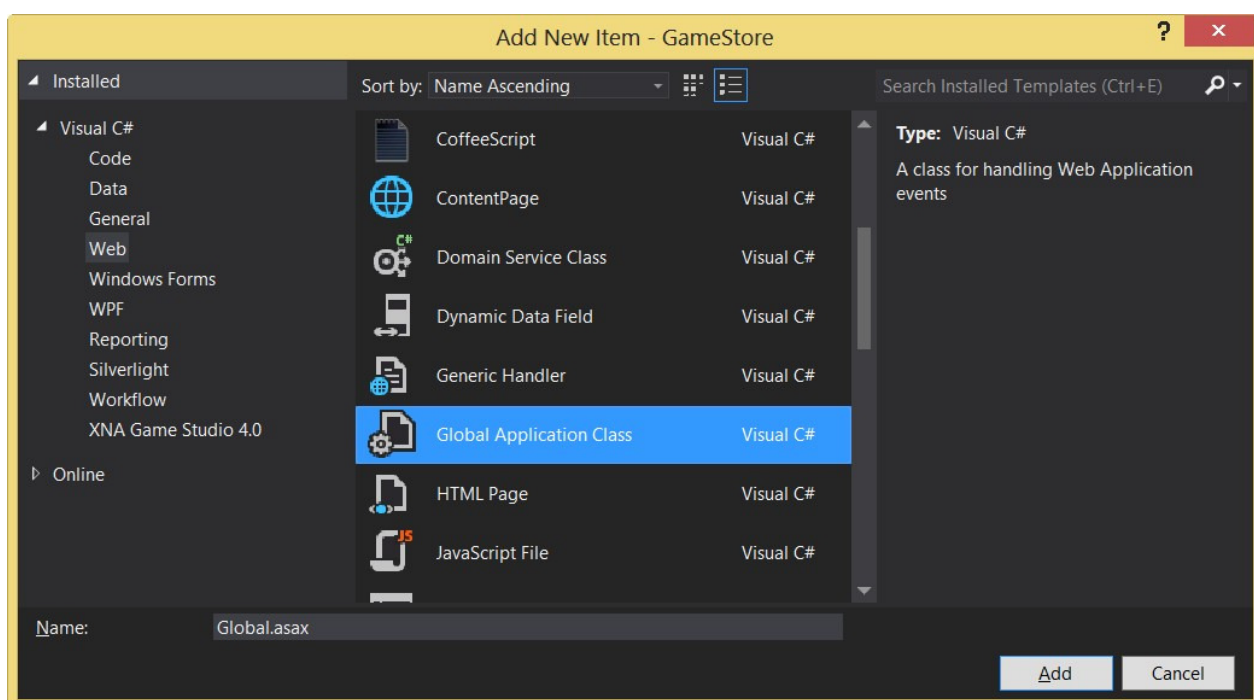
проектов Web Forms. Мы не считаем эти шаблоны особенно полезными и хотели бы получать в проектах только средства и файлы, которые нужны, а не все разновидности обобщенных и неиспользуемых файлов. Мы не особо высокого мнения относительно предлагаемых шаблонов для всех типов проектов .NET, и они не содержат ничего такого, что нельзя было бы легко добавить самостоятельно.

Мы обычно всегда выбираем вариант пустого проекта, поскольку хотим объяснить, каким образом работает каждое важное средство, и показать, как добавить или установить такое средство в любом приложении Web Forms. В собственных проектах вы можете использовать предварительно сконфигурированные шаблоны — и многие так поступают. Однако мы уверены, что вы многое выиграете, если начнете с основ и поймете предназначение и роль всего, что добавляете самостоятельно.

Добавление глобального класса приложения

Когда приложение Web Forms запускается, исполняющая среда ищет глобальный класс приложения, который служит для реагирования на события жизненного цикла приложения и обычно используется для выполнения одноразовых задач по настройке. Глобальные классы приложений имеют суффикс .aspx и по соглашению существует один такой класс по имени Global.aspx. Для запуска в работу приложения Web Forms наличие глобального класса приложения не обязательно, но большинство приложений требуют определенного начального конфигурирования, и некоторыми средствами, используемыми в приложении GameStore, проще всего управлять через глобальный класс приложения.

Чтобы добавить глобальный класс приложения, щелкните правой кнопкой мыши на элементе GameStore в окне Solution Explorer и выберите в контекстном меню пункт Add --> New Item. Найдите шаблон Global Application Class, как показано на рисунке ниже, удостоверьтесь, что в качестве имени указано Global.aspx, и щелкните на кнопке Add (Добавить).



Среда Visual Studio создаст файл Global.aspx и файл класса отделенного кода

по имени `Global.asax.cs`. Способ обработки этих файлов в Visual Studio отражает способ их использования — двойной щелчок на файле `Global.asax` в окне Solution Explorer приводит к открытию для редактирования файла `Global.asax.cs`. Начальный контент файла `Global.asax.cs`, создаваемый Visual Studio, показан ниже:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace GameStore
{
    public class Global : System.Web.HttpApplication
    {

        protected void Application_Start(object sender, EventArgs e)
        {

        }

        protected void Session_Start(object sender, EventArgs e)
        {

        }

        protected void Application_BeginRequest(object sender, EventArgs e)
        {

        }

        protected void Application_AuthenticateRequest(object sender, EventArgs e)
        {

        }

        protected void Application_Error(object sender, EventArgs e)
        {

        }

        protected void Session_End(object sender, EventArgs e)
        {

        }

        protected void Application_End(object sender, EventArgs e)
        {

        }
    }
}
```

Каждый метод позволяет определять код, который будет выполняться в ключевых точках жизненного цикла приложения — когда приложение запускается, останавливается, возникает ошибка и т.д. Для регистрации файла Global.asax никаких действий предпринимать не нужно. Исполняющая среда ASP.NET Framework находит его автоматически при запуске приложения, и его методы будут выполняться по мере возникновения соответствующих событий жизненного цикла.

Чтобы просмотреть содержимое файла Global.asax (а не класса отдельного кода), щелкните правой кнопкой мыши на элементе Global.asax в окне Solution Explorer и выберите в контекстном меню пункт View Markup (Просмотреть разметку). Открывшийся файл содержит единственную строку, которая показана ниже:

```
<%@ Application Codebehind="Global.asax.cs" Inherits="GameStore.Global" Language
```

Фрагмент кода похож на фрагменты кода в файлах веб-форм, но в нем применяется директива Application для сообщения среде ASP.NET Framework о том, что это часть глобального класса приложения, а также информации о месте, где может быть найден класс отдельного кода. Пока что мы не собираемся добавлять какой-либо специальный код в файл Global.asax.cs.

Webbaserat Affärssystem

Ledande system för företag som säljer tid och projekt. Från Visma

