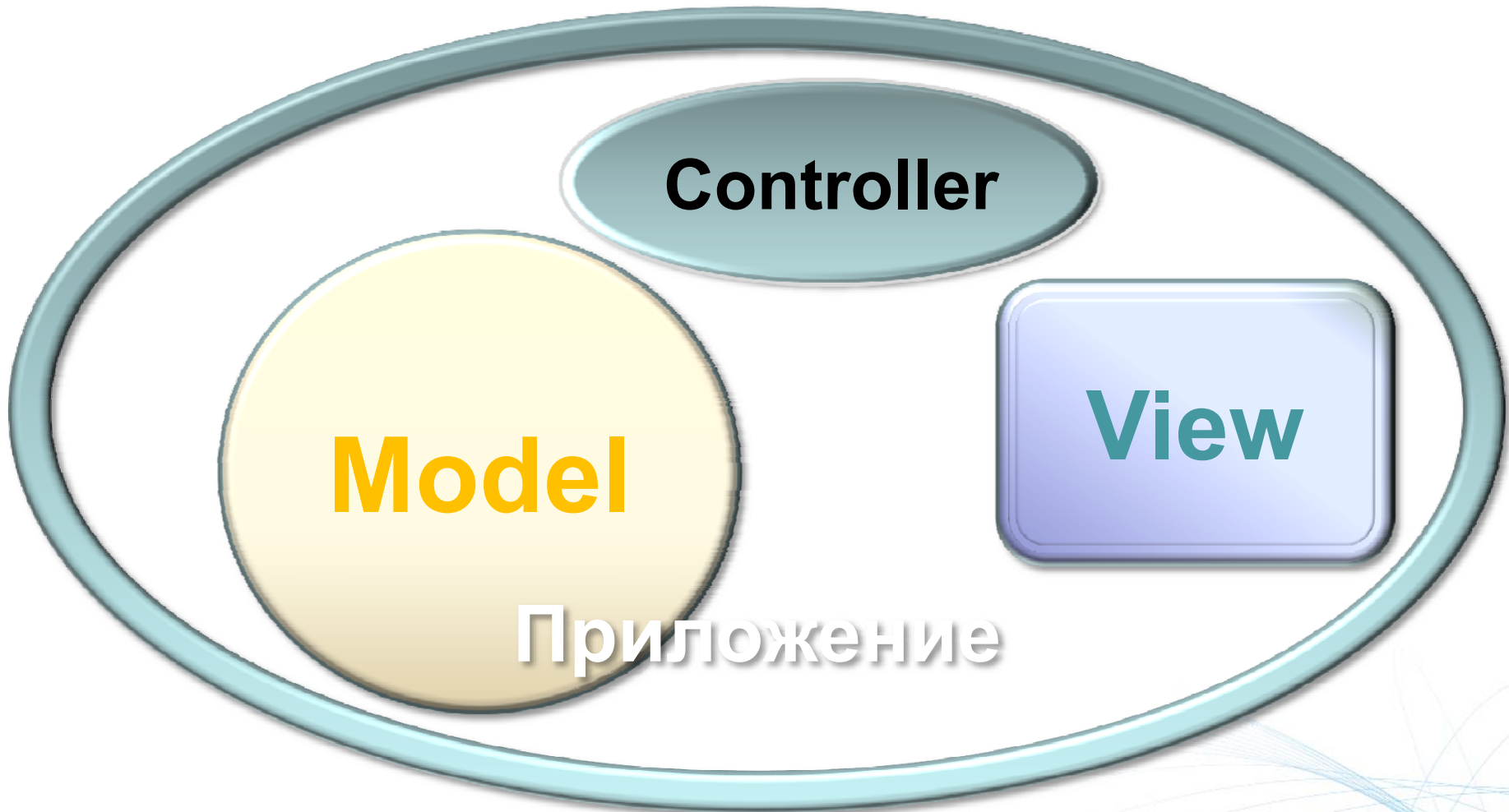




# ASP.NET MVC

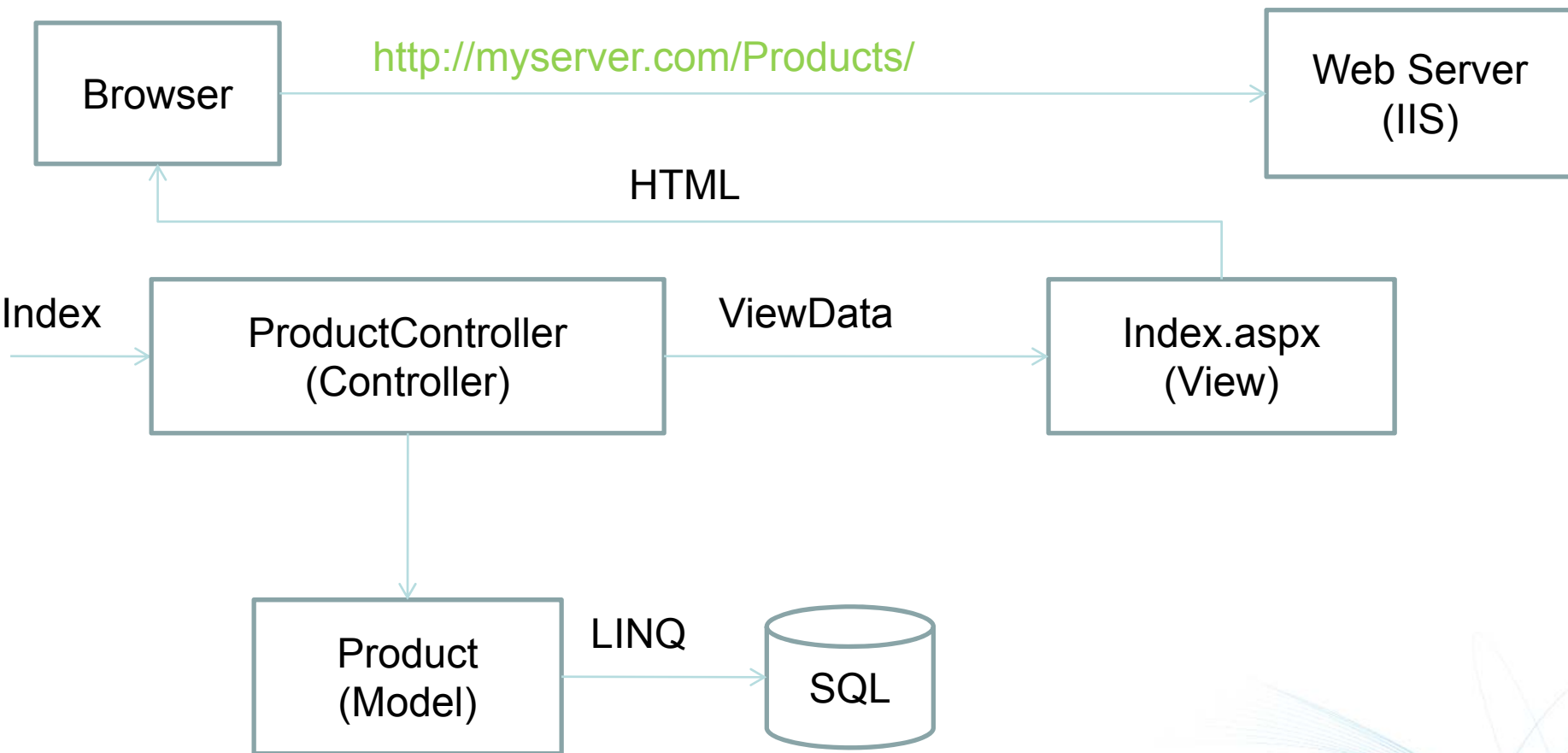


## Главный принцип - разделение





## MVC в реализации Microsoft





## Установка дополнений

- **Visual Studio 2008 SP1**

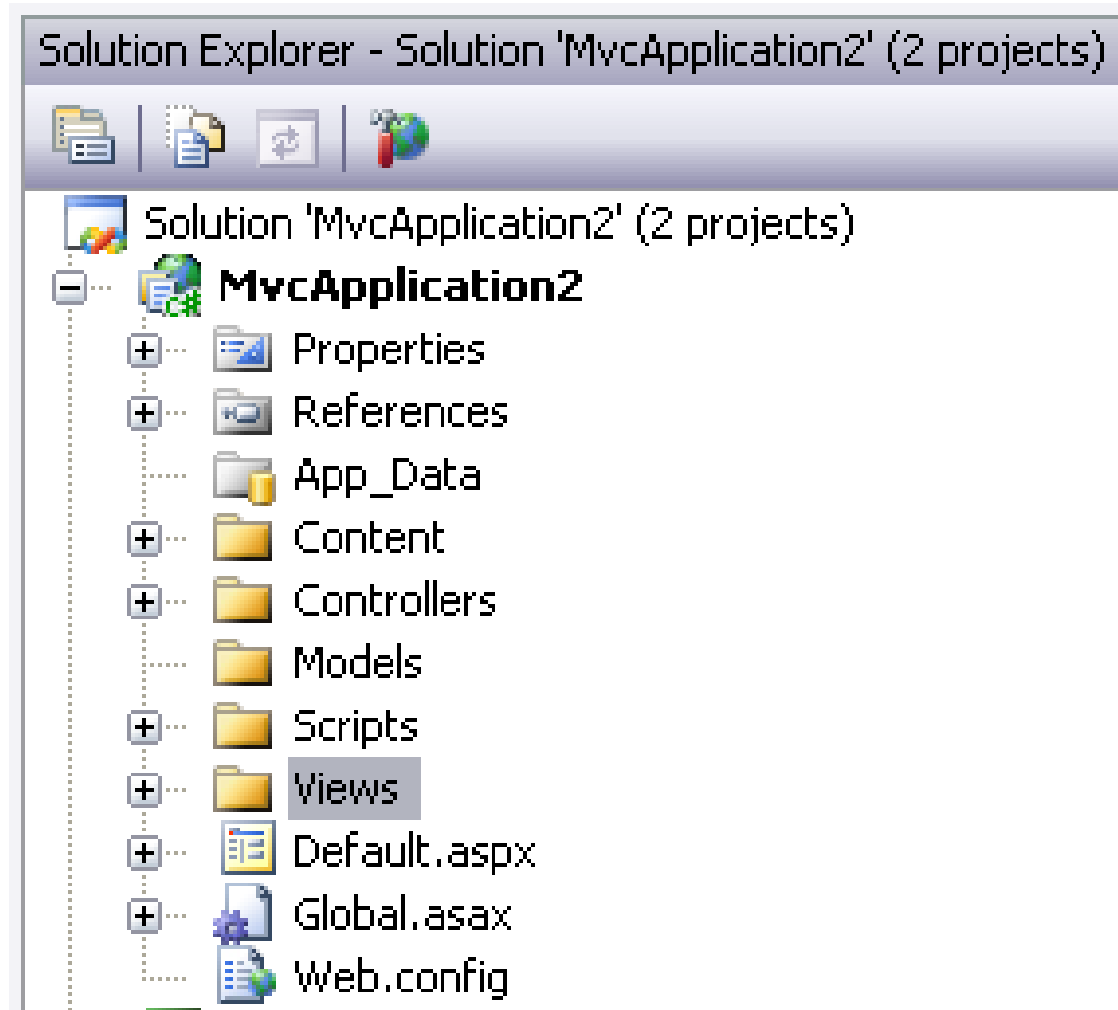
<http://www.microsoft.com/downloads/details.aspx?FamilyId=FBEE1648-7106-44A7-9649-6D9F6D58056E&displaylang=en>

- **ASP.NET MVC 0.1**

<http://www.microsoft.com/downloads/details.aspx?FamilyID=53289097-73ce-43bf-b6a6-35e00103cb4b&displaylang=en>



## Новый тип проекта ASP.NET Web Application





Запуск приложения



## Маршрутизация

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

    routes.MapRoute(
        "Default",
        "{controller}/{action}/{id}",
        new { controller="Home", action="Index", id="" }
    );
}
```



## Маршрутизация

**http://localhost:6046/{controller}/{action}/{id}**

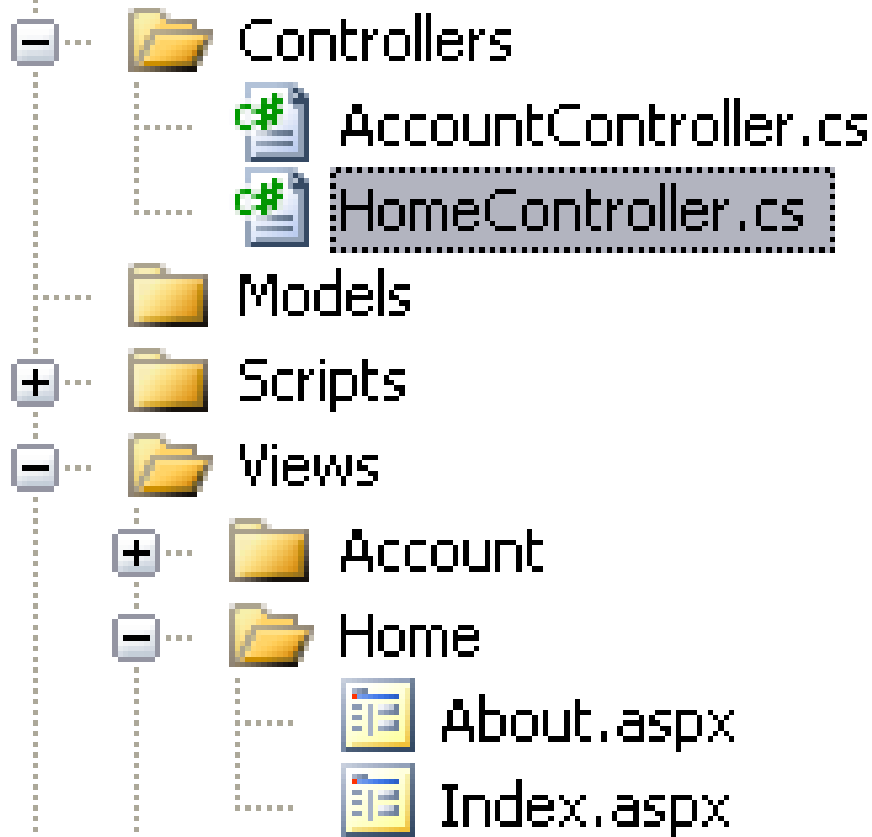


## Маршрутизация

<http://localhost:6046/Home/Index>



## Принятые соглашения





## Передача данных представлению

```
public ActionResult Index()  
{  
    ViewData["Message"] = "Welcome to ASP.NET MVC!";  
    return View();  
}
```



## Передача данных представлению

```
<asp:Content  
    ID="indexContent "  
    ContentPlaceHolderID="MainContent«  
    runat="server">
```

```
<h2><%= Html.Encode(ViewData[ "Message" ] ) %></h2>
```

```
<p>
```

```
To learn more about ASP.NET MVC visit <a  
    href="http://asp.net/mvc" title="ASP.NET MVC  
    Website">http://asp.net/mvc</a>.
```

```
</p>
```

```
</asp:Content>
```



## Передача данных через DataView

- Простой и быстрый способ передачи данных
- Строковый индекс нужно вводить руками, долго, не удобно и может привести к опечаткам
- Такие ошибки не обнаруживаются на этапе компиляции
- Необходимо приведение к типам

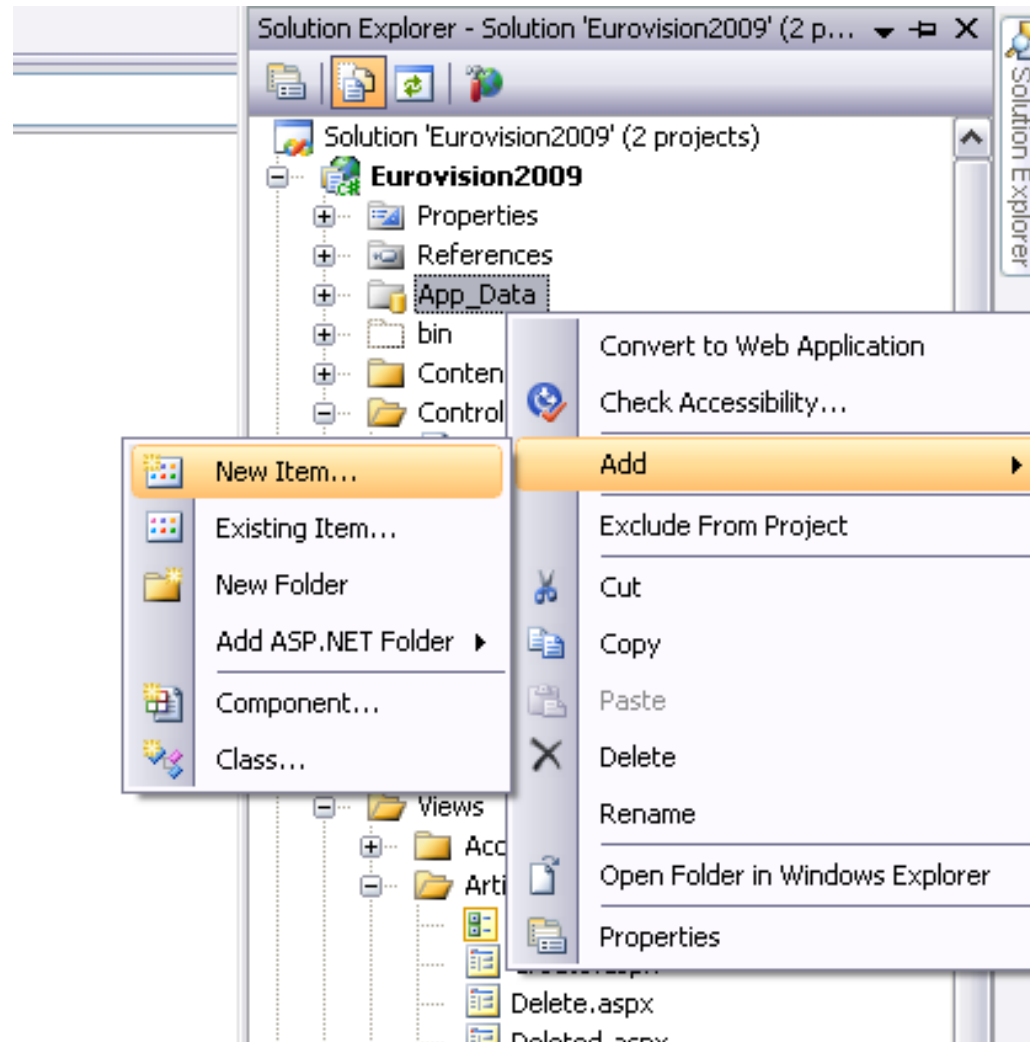


## Способ передачи данных ViewModel

- Безопасность типов
- Проверка при компиляции
- Удобство редактирования
- Существуют способы автоматической инициализации моделей

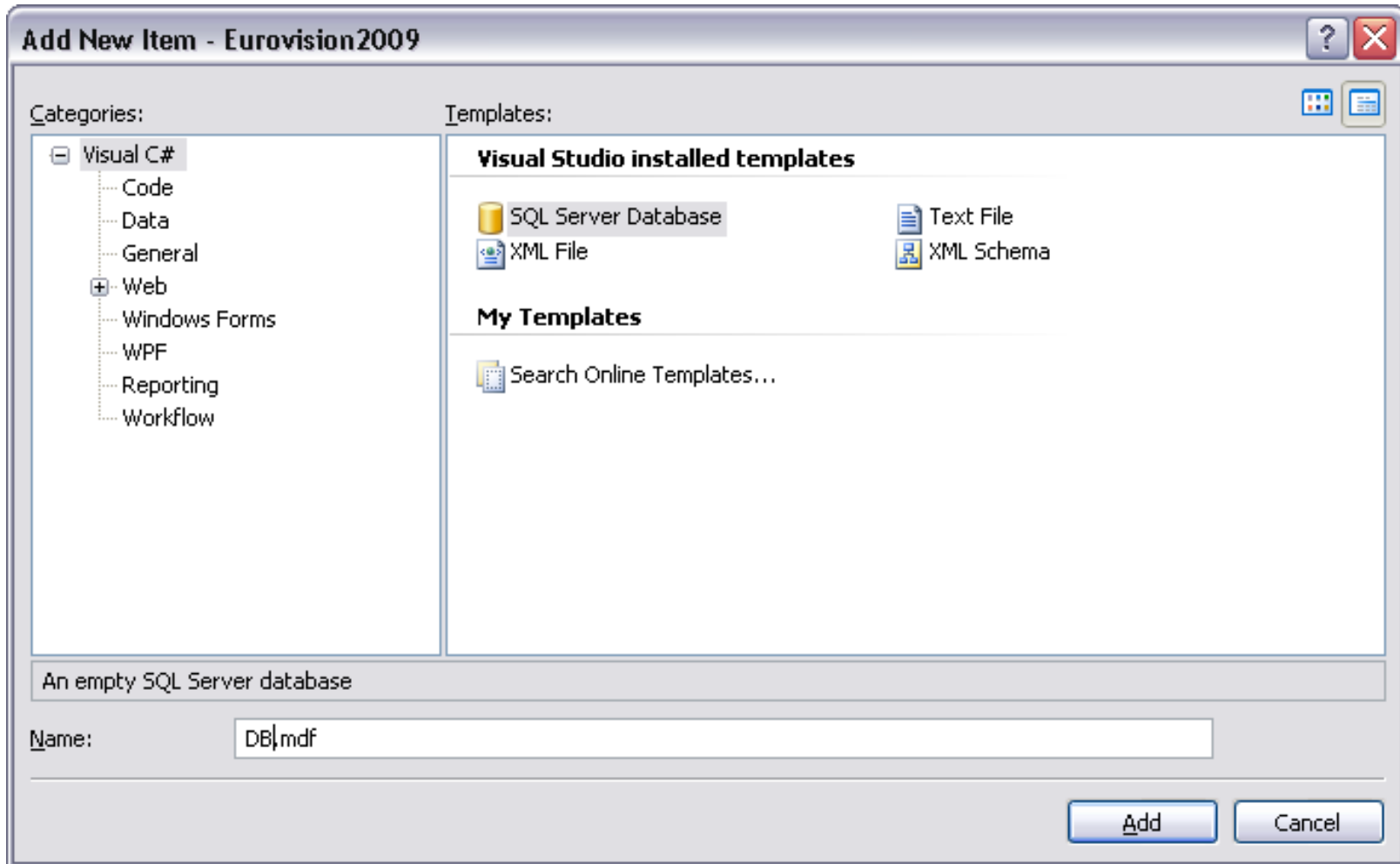


## Добавление базы данных в приложение



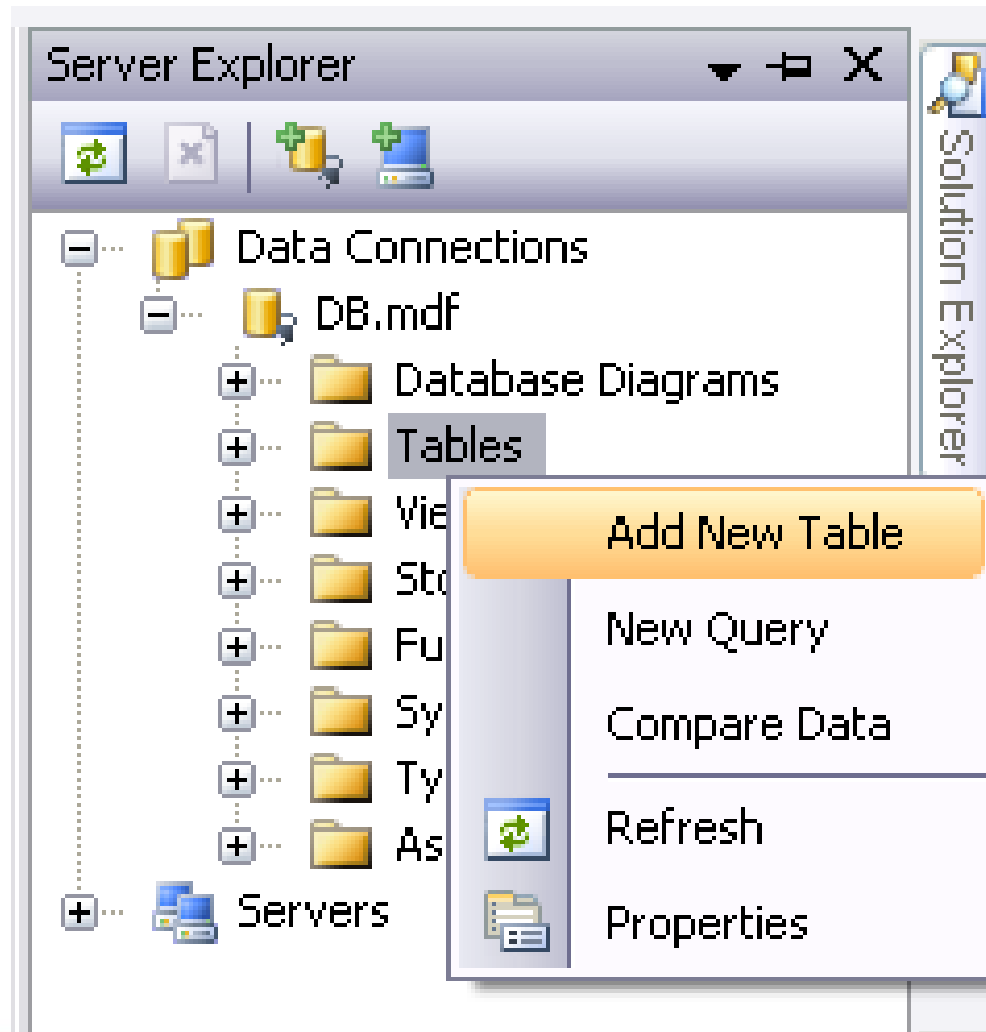


## Добавление базы данных в приложение





## Добавление базы данных в приложение





## Добавление базы данных в приложение

dbo.Artist: Ta...P\_DATA\DB.MDF | HomeController.cs | ArtistForm.aspx | Deleted.aspx | Delete.aspx

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
Country	nvarchar(50)	<input type="checkbox"/>
Performer	nvarchar(50)	<input type="checkbox"/>
Songs	nvarchar(50)	<input type="checkbox"/>
Points	int	<input type="checkbox"/>

Column Properties

Collation	<data
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes



## Добавление базы данных в приложение

**Choose Name** ? X

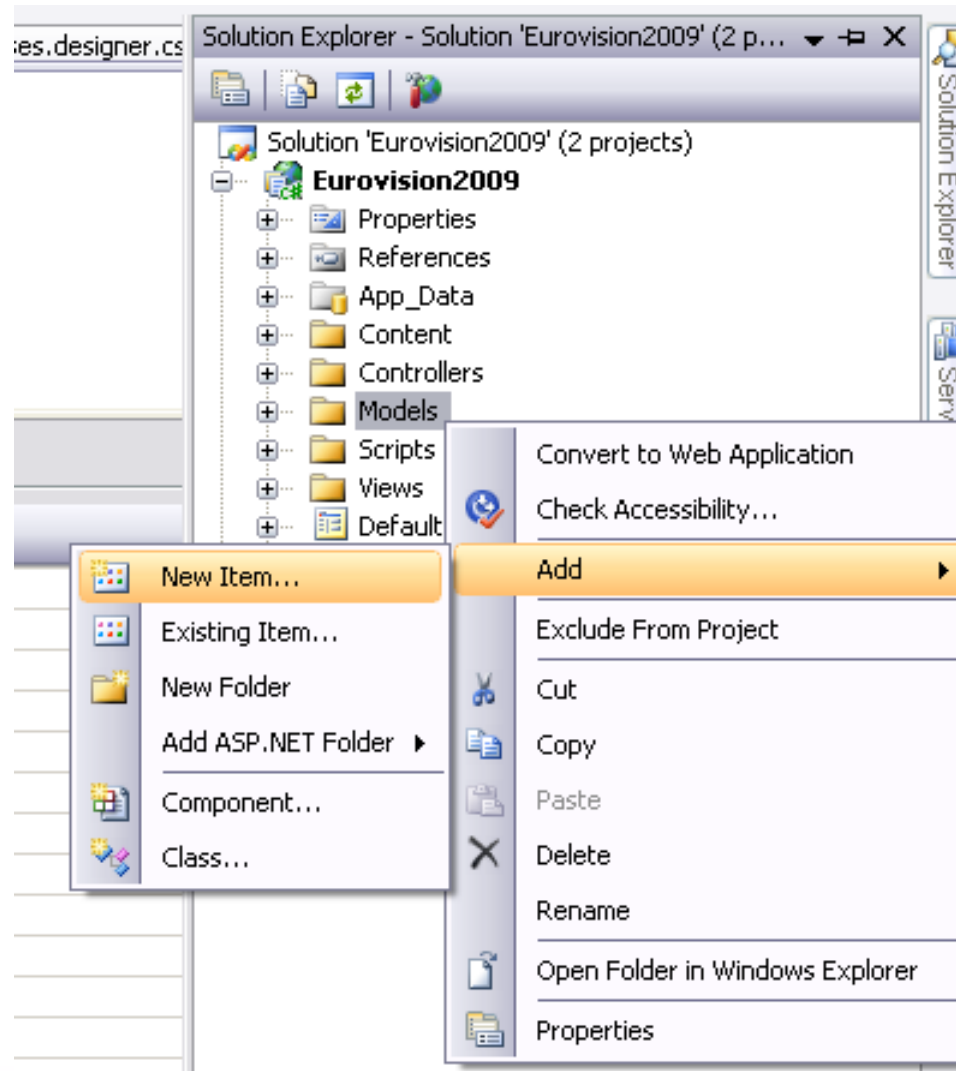
Enter a name for the table:

Artist|

OK Cancel

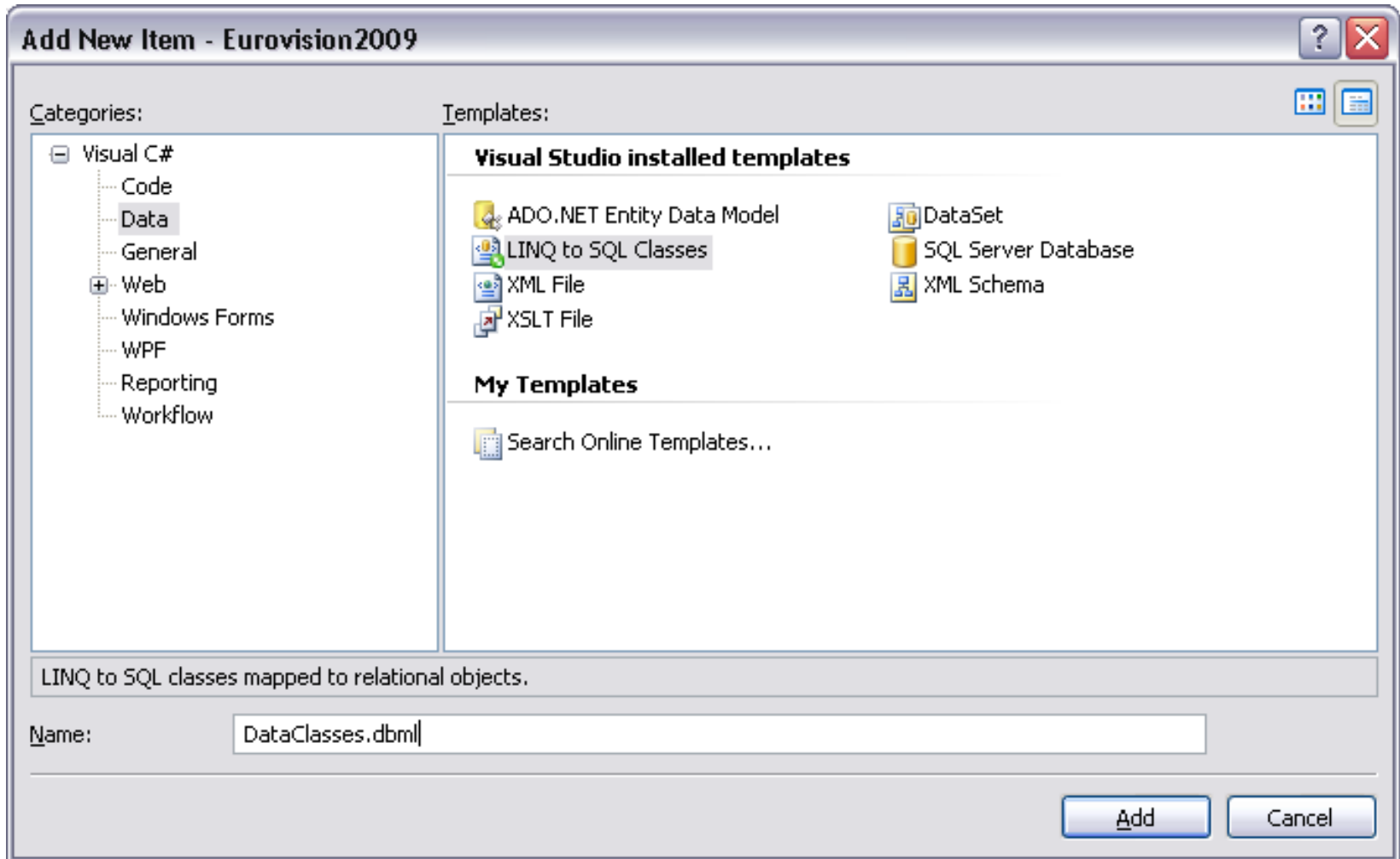


## Создание модельных классов



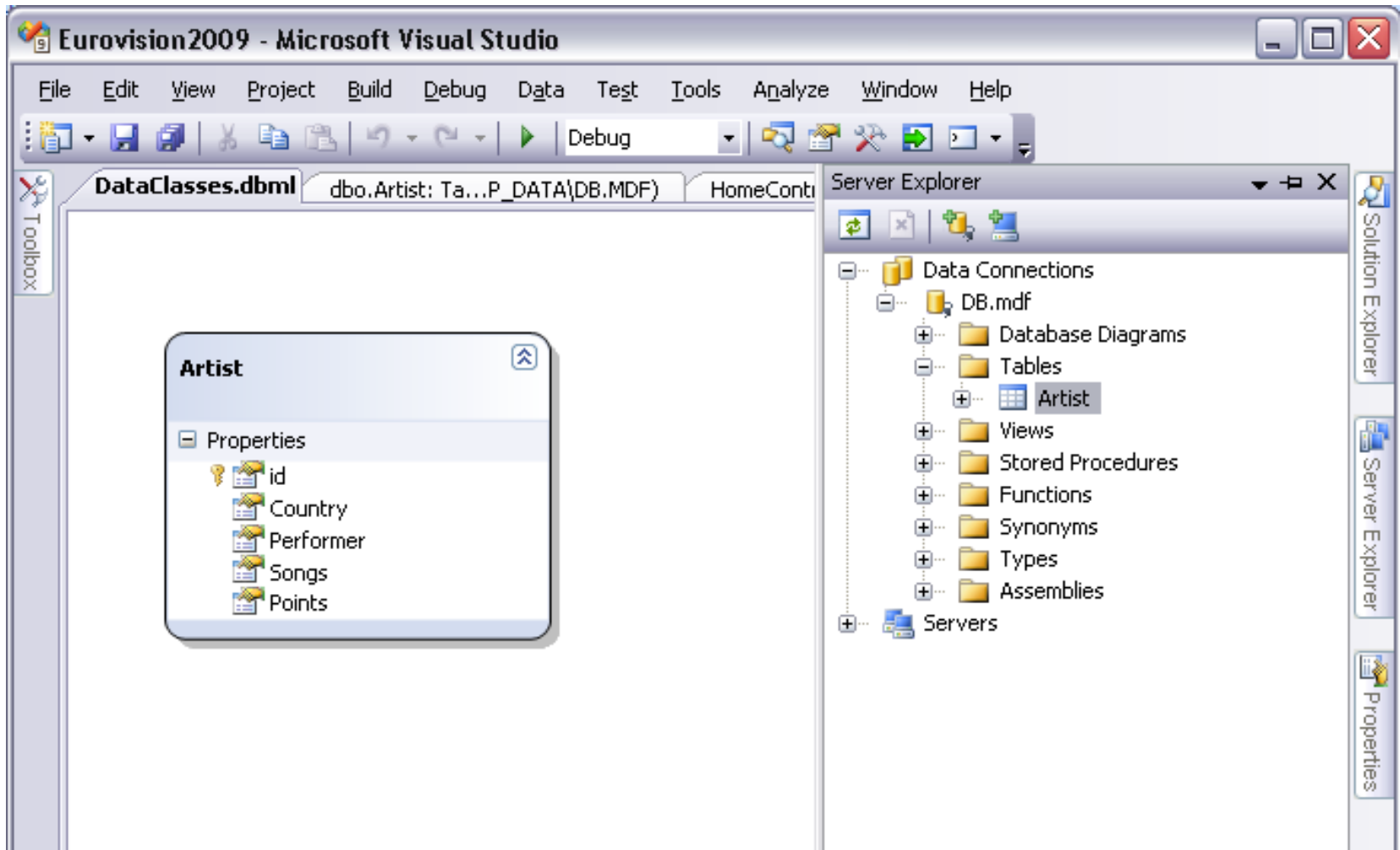


## Создание модельных классов



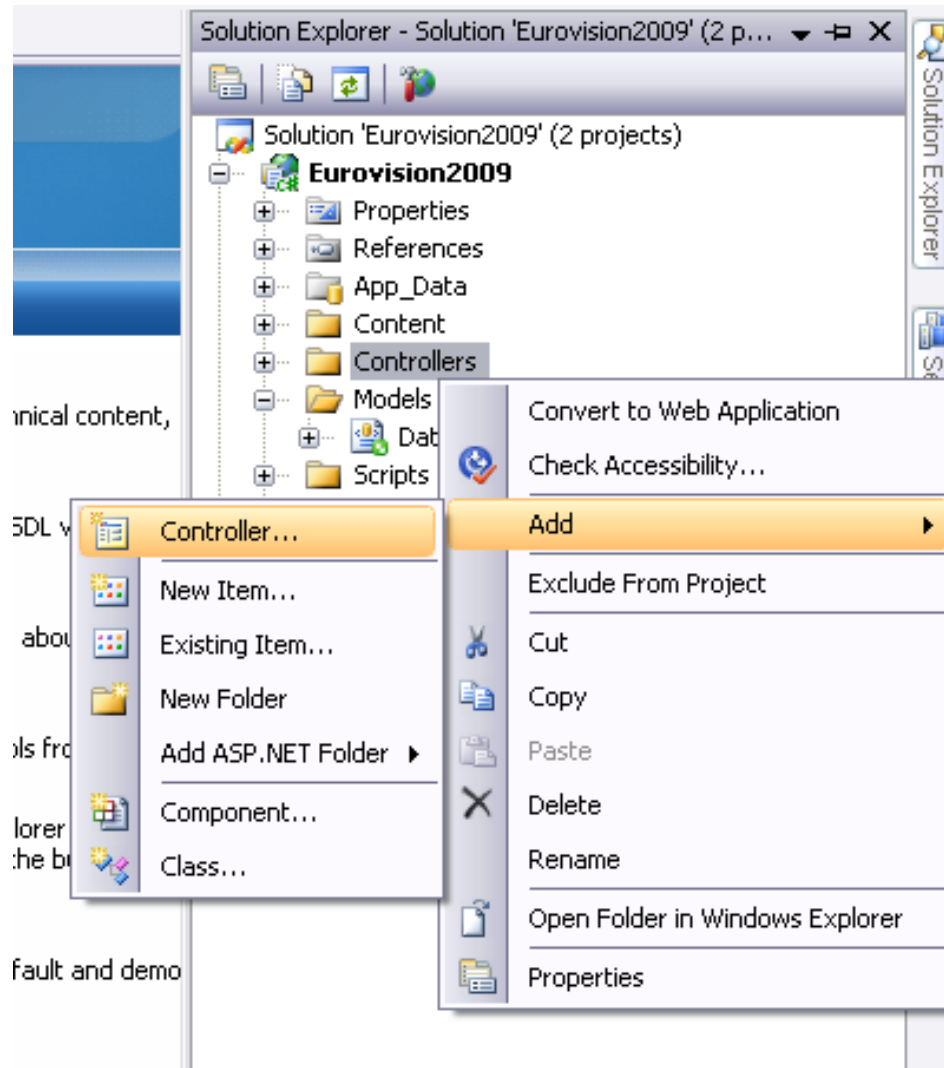


## Создание модельных классов






## Добавление нового контроллера





## Добавление нового контроллера

**Add Controller** 

Controller Name:

Add action methods for Create, Update, and Details scenarios



## Добавление нового контроллера

- Index – выводит список
- Details – показывает информацию одной записи
- Create – 2 перегруженных метода для создания
- Edit – 2 перегруженных метода для редактирования



## Передача данных с помощью ViewModel

```
DataClassesDataContext dataClasses =  
    new DataClassesDataContext();  
  
public ActionResult Index()  
{  
    return View(dataClasses.Artists);  
}
```

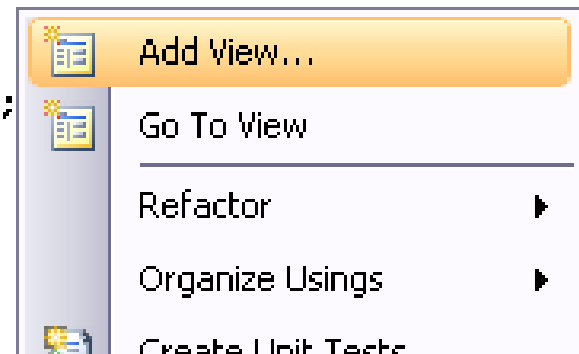


## Создание представления

```
public class ArtistController : Controller
{
    DataClassesDataContext dataClasses = new DataClassesDataContext();

    //
    // GET: /Artist/

    public ActionResult Index()
    {
        return View(dataClasses.Artists);
    }
}
```





## Создание представления

**Add View** ✕

View name:

Create a partial view (.ascx)

Create a strongly-typed view

View data class:  
 ▾

View content:  
 ▾

Select master page

...

ContentPlaceholder ID:



## Создание представлений

```
<% foreach (var item in Model) { %>
    <tr>
        <td>
            <%= Html.ActionLink("Edit", "Edit",
                new { id=item.id }) %> |
            <%= Html.ActionLink("Details", "Details",
                new { id=item.id })%>
        </td>
        <td>
            <%= Html.Encode(item.Country) %>
        </td>
        <td>
            <%= Html.Encode(item.Performer) %>
        </td>
    </tr>
<% } %>
```



## Передача данных в метод контроллера

- Через параметры строки запроса
  - Описать параметр в таблице маршрутизации. Описать в сигнатуре метода параметр с подходящим типом
- Через input элементы формы
  - Request.Form
  - FormCollection
  - Описать в сигнатуре метода объект с публичными полями, имена которых будут совпадать с именами input элементов



код  
выборка, обновление, создание



## UserControl здесь тоже есть!

- Две причины использовать UserControl:
  - Повторяемые части кода
  - Упрощение сложных форм



## Сравнение ASP.NET MVC с Web Forms

### Web Forms

- Rapid Application Development  
быстрая разработка приложений
- Огромное количество готовых библиотек
- Много знающих людей – не нужно тратить время на обучение
- Лучше подход для тяжелых приложений с большим количеством данных

### ASP.NET MVC

- Хорошо подходит для Test Driven Development
- Позволяет повторно использовать модель с другим UI
- Полный контроль над HTML кодом. Легковесные страницы
- URL хорошо индексируется поисковыми системами



## Ссылки

- <http://www.asp.net/mvc>
- <http://msdn.microsoft.com>
- <http://www.codeplex.com/aspnet>
- <http://weblogs.asp.net/scottgu/default.aspx>
- <http://www.techdays.ru>
- <http://habrahabr.ru/blogs/net>



# Цпасибо

?