# 模型绑定

## 创建项目

项目名称：MvcModels

模板：Basic

下面是项目的基础文件及其内容：

模型类：Person.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace MvcModels.Models

{

public partial class Person

{

public int PersonId { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public DateTime BirthDate { get; set; }

public Address HomeAddress { get; set; }

public bool IsApproved { get; set; }

public Role Role { get; set; }

}

public class Address

{

public string Line1 { get; set; }

public string Line2 { get; set; }

public string City { get; set; }

public string PostalCode { get; set; }

public string Country { get; set; }

}

public enum Role

{

Admin,

User,

Guest

}

}

控制器：HomeController.cs

using MvcModels.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace MvcModels.Controllers

{

public class HomeController : Controller

{

private Person[] personData =

{

new Person{PersonId = 1, FirstName = "Adam",LastName = "Freeman",Role = Role.Admin},

new Person{PersonId = 2 ,FirstName = "Steven",LastName = "Sanderson",Role = Role.Admin},

new Person{PersonId = 3 ,FirstName = "Jacqui",LastName = "Griffyth",Role = Role.User},

new Person{PersonId = 4 ,FirstName = "John",LastName = "Smith",Role = Role.User},

new Person{PersonId = 5 ,FirstName = "Anne",LastName = "Jones",Role = Role.Guest}

};

public ActionResult Index(int id)

{

Person dataItem = personData.Where(p => p.PersonId == id).First();

return View(dataItem);

}

}

}

视图：Index.cshtml（强类型）

@model MvcModels.Models.Person

@{

ViewBag.Title = "Index";

}

<h2>Person</h2>

<div><label>ID:</label>@Html.DisplayFor(m => m.PersonId)</div>

<div><label>First Name:</label>@Html.DisplayFor(m => m.FirstName)</div>

<div><label>Last Name:</label>@Html.DisplayFor(m => m.LastName)</div>

<div><label>Role:</label>@Html.DisplayFor(m => m.Role)</div>

CSS样式：Site.css

…

label {

display: inline-block;

width: 100px;

font-weight: bold;

margin: 5px;

}

form label {

float: left;

}

input.text-box {

float: left;

margin: 5px;

}

button[type=submit] {

margin-top: 5px;

float: left;

clear: left;

}

form div {

clear: both;

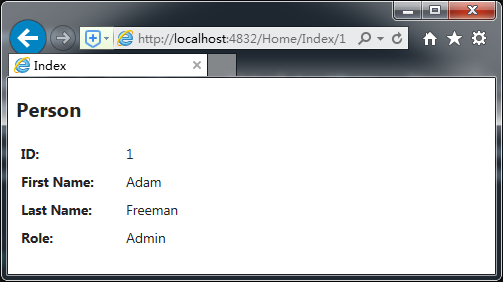
}

…

## 理解模型绑定

模型绑定在HTTP请求和C#（指的是MVC中的动作方法）直接起到了桥梁的作用。大多数MVC项目都在某种程度上依赖模型绑定。

看看刚才创建的项目是否能够正常启动，并查看模型绑定是否正常工作：



从上图看我们的模型绑定是没有问题的，这个页面对应的URL是Home/Index/1，它包含了查看Person对象的PersonId属性值，如：Home/Index/1。

MVC自动地对该URL进行了解析，并在调用Home控制器中的Index方法时获取1作为其参数，因此使得程序能够按照预期运行。这种将URL片段转换成int型方法参数的过程是模型绑定的一个例子。

当程序接收到请求并由路由引擎处理时，模型绑定过程便开始了。该示例中未对默认路由进行调整，也就不再这里展示了。

动作调用器会使用该路由信息推断出，为该请求进行服务所需要的是Index动作方法。但此时还不能调用，直到获取该方法的参数所需的有用的值。

默认的动作调用器，ControllerActionInvoker，要依靠模型绑定器来生成调用动作所需要的数据对象。模型绑定器由IModeBinder接口所定义：

namespace System.Web.Mvc

{

public interface IModelBinder

{

object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext);

}

}

## 使用默认的模型绑定器

### 绑定简单类型

### 绑定复合类型

### 绑定到数组与集合

## 手工调用模型绑定

## 定制模型绑定系统

### 创建自定义的值提供器

### 创建自定义模型绑定器