窗体顶端



[返回主页](http://www.cnblogs.com/stalwart/)

[**风黑月高**](http://www.cnblogs.com/stalwart/)

**--风黑月高之时享受CODE之快乐！**

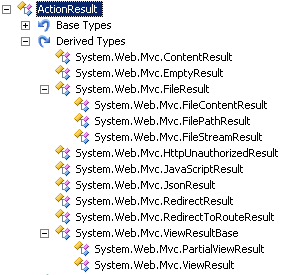
* [博客园](http://www.cnblogs.com/)
* [首页](http://www.cnblogs.com/stalwart/)
* [新闻](http://news.cnblogs.com/)
* [新随笔](http://www.cnblogs.com/stalwart/admin/EditPosts.aspx?opt=1)
* [联系](http://space.cnblogs.com/msg/send/%e9%a3%8e%e9%bb%91%e6%9c%88%e9%ab%98kable)
* [管理](http://www.cnblogs.com/stalwart/admin/EditPosts.aspx)
* [订阅](http://www.cnblogs.com/stalwart/rss) [订阅](http://www.cnblogs.com/stalwart/rss)

随笔- 77  文章- 5  评论- 4

[**MVC Controller与ActionResult的返回值**](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)

Action的要求  
• 必须是一个公有方法  
• 必须返回ActionResult类型  
• 必须是实例方法  
• 不能是范型方法  
• 没有标注NonActionAttribute  
• 不能被重载（overload）

Controller 提供了众多的方法让我们返回各种类型的 ActionResult。



**1. View**  
  
最常用的一种，用于返回一个 "标准" 页面。

protected internal virtual ViewResult View(string viewName, string masterName, object model)  
{  
  if (model != null)  
  {  
    base.ViewData.Model = model;  
  }  
  
  return new ViewResult   
  {   
    ViewName = viewName,   
    MasterName = masterName,   
    ViewData = base.ViewData,   
    TempData = base.TempData   
  };  
}  
  
public class ViewResult : ViewResultBase  
{  
  protected override ViewEngineResult FindView(ControllerContext context)  
  {  
    ViewEngineResult result = ViewEngineCollection.FindView(context, ViewName, MasterName);  
  
    if (result.View != null)  
    {  
      return result;  
    }  
  
    ...  
  }  
}

这个页面默认是 ViewPage，也可以是我们自己定义的其它模板引擎页面。  
  
MVC 还提供了强类型的 ViewPage<TModel>。

public class User  
{  
  public string Name { get; set; }  
  public int Age { get; set; }  
}  
  
public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    ViewData["message"] = "Hello, World!";  
    var model = new User { Name = "Tom", Age = 13 };  
  
    return View(model);  
  }  
}

Index.aspx

<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage<Learn.MVC.Controllers.User>" %>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
  <title>Index</title>  
</head>  
<body>  
  Name: <%= Model.Name %>; Age: <%= Model.Age %>  
</body>  
</html>

在 WebForm 时代，我们就已经习惯了将一个页面分解成多个 UserControl，现在我们依然可以这么做。htmlHelper 专门提供了 RenderPartial 扩展方法，从当前视图目录(Views\xxx)下载入 .ascx 页面。

public static class RenderPartialExtensions  
{  
  public static void RenderPartial(this HtmlHelper htmlHelper, partialViewName, model, viewData)  
  {  
    htmlHelper.RenderPartialInternal(partialViewName, viewData, model, ViewEngines.Engines);  
  }  
}  
  
public class HtmlHelper  
{  
  internal virtual void RenderPartialInternal(string partialViewName, ViewDataDictionary viewData,   
    object model, ViewEngineCollection viewEngineCollection)  
  {  
    ...  
  
    ViewDataDictionary newViewData = null;  
  
    if (model == null)  
    {  
      if (viewData == null)  
        newViewData = new ViewDataDictionary(ViewData);  
      else  
        newViewData = new ViewDataDictionary(viewData);  
    }  
    else  
    {  
      if (viewData == null)  
        newViewData = new ViewDataDictionary(model);  
      else  
        newViewData = new ViewDataDictionary(viewData) { Model = model };  
    }  
  
    ViewContext newViewContext = new ViewContext(ViewContext, ViewContext.View,   
      newViewData, ViewContext.TempData);  
  
    IView view = FindPartialView(newViewContext, partialViewName, viewEngineCollection);  
  
    view.Render(newViewContext, ViewContext.HttpContext.Response.Output);  
  }  
  
  internal static IView FindPartialView(viewContext, partialViewName, viewEngineCollection)  
  {  
    ViewEngineResult result = viewEngineCollection.FindPartialView(viewContext, partialViewName);  
  
    if (result.View != null)  
    {  
      return result.View;  
    }  
  
    ...  
  }  
}

RenderPartialInternal 调用 FindParitialView 从视图引擎中载入 .ascx，同时将当前的环境参数传递给它。也就是说 RenderPartial 只是一种视图级别的行为，并不会再次触发 Controller Action 操作，这点要和 Controller.PartialView() 区别开来。  
  
Index.aspx

<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage<Learn.MVC.Controllers.User>" %>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
  <title>Index</title>  
</head>  
<body>  
  Name: <%= Model.Name %>; Age: <%= Model.Age %>  
  <br />  
  <% Html.RenderPartial("Part"); %>  
</body>  
</html>

Part.ascx

<%@ Control Language="C#" Inherits="System.Web.Mvc.ViewUserControl<Learn.MVC.Controllers.User>" %>  
  
<%= ViewData["message"] %>   
<br />  
<%= Model.Name %>

**2. Content**  
  
Content 用于输出(Response.Write) "静态" 片段。

protected internal virtual ContentResult Content(content, contentType, contentEncoding)  
{  
  return new ContentResult  
  {  
    Content = content,  
    ContentType = contentType,  
    ContentEncoding = contentEncoding  
  };  
}  
  
public class ContentResult : ActionResult  
{  
  public string Content { get; set; }  
      
  public override void ExecuteResult(ControllerContext context)  
  {  
    ...  
  
    HttpResponseBase response = context.HttpContext.Response;  
  
    if (!String.IsNullOrEmpty(ContentType))  
    {  
      response.ContentType = ContentType;  
    }  
    if (ContentEncoding != null)  
    {  
      response.ContentEncoding = ContentEncoding;  
    }  
    if (Content != null)  
    {  
      response.Write(Content);  
    }  
  }  
}

看看和 jQuery 的配合使用。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return View();  
  }  
  
  public ActionResult Part()  
  {  
    return Content("<a href=\"http://www.rainsts.net\">Q.yuhen</a>");  
  }  
}

Index.aspx

<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage" %>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
  <title>Index</title>  
  <script src="http://www.cnblogs.com/Scripts/jquery-1.3.1.min.js" type="text/javascript"></script>  
  <script type="text/javascript">  
    $(function()  
    {  
      $("#div1").load("/test/part");  
    });  
  </script>  
</head>  
<body>  
  <div id="div1">  
  </div>  
</body>  
</html>

**3. PartialView**  
  
Controller.PartialView() 和 HtmlHelper.RenderPartial() 的不同之处在于前者是再次执行 ActionInvoke 并返回一个 ActionResult 结果，后者只是使用现有的 ViewContext 显示一个视图片段。而与 Controller.Content() 的区别是 PartialView() 使用视图引擎输出一个 "动态" 的 ascx 结果。

protected internal virtual PartialViewResult PartialView(string viewName, object model)  
{  
  if (model != null)  
  {  
    ViewData.Model = model;  
  }  
  
  return new PartialViewResult  
  {  
    ViewName = viewName,  
    ViewData = ViewData,  
    TempData = TempData  
  };  
}  
  
public class PartialViewResult : ViewResultBase  
{  
  protected override ViewEngineResult FindView(ControllerContext context)  
  {  
    ViewEngineResult result = ViewEngineCollection.FindPartialView(context, ViewName);  
  
    if (result.View != null)  
    {  
      return result;  
    }  
      
    ...  
  }  
}

和 Content() 一样，我们通常将其和 jQuery 等 Ajax 框架配合使用。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return View();  
  }  
  
  public ActionResult Part()  
  {  
    ViewData["time"] = DateTime.Now;  
    var model = new User { Name = "Tom", Age = 13 };  
  
    return PartialView(model);  
  }  
}

Index.aspx

<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage" %>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
  <title>Index</title>  
  <script src="http://www.cnblogs.com/Scripts/jquery-1.3.1.min.js" type="text/javascript"></script>  
  <script type="text/javascript">  
    $(function()  
    {  
      $("#div1").load("/test/part");  
    });  
  </script>  
</head>  
<body>  
  <div id="div1">  
  </div>  
</body>  
</html>

Part.ascx

<%@ Control Language="C#" Inherits="System.Web.Mvc.ViewUserControl<Learn.MVC.Controllers.User>" %>  
  
<%= ViewData["time"] %> <br />  
<%= Model.Name %>; <%= Model.Age %>

**4. Redirect / RedirectToAction / RedirectToRoute**  
  
Controller 提供了几种方式，让我们在不同的 Action 之间进行跳转。

public class MvcApplication : System.Web.HttpApplication  
{  
  public static void RegisterRoutes(RouteCollection routes)  
  {  
    ...  
  
    routes.MapRoute  
    (  
      "Test2",  
      "Test/T2/{name}/{age}",  
      new { controller = "Test", action = "T2", name = "", age = 0 }  
    );  
  
    ...  
  }  
}

**方法1**:   
  
Redirect() 直接用 Response.Redirect() 完成 url 跳转。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return Redirect("/Test/T2/Tom/23");  
  }  
  
  public ActionResult T2(User user)  
  {  
    return Content(user.Name);  
  }  
}

相关细节：

protected internal virtual RedirectResult Redirect(string url)  
{  
  ...  
  return new RedirectResult(url);  
}  
  
public class RedirectResult : ActionResult  
{  
  public override void ExecuteResult(ControllerContext context)  
  {  
    ...  
  
    string destinationUrl = UrlHelper.Content(Url, context.HttpContext);  
    context.HttpContext.Response.Redirect(destinationUrl, false /\* endResponse \*/);  
  }  
}

**方法2**:   
  
RedirectToAction() 直接使用 Action Name 进行跳转。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return RedirectToAction("T2", new { name = "Tom", age = 23 });  
  }  
  
  public ActionResult T2(User user)  
  {  
    return Content(user.Name);  
  }  
}

如果目标 Action 不在当前 Controller 类，则可以指定目标 Controller Name。

return RedirectToAction("T2", new { controller="Test2", name = "Tom", age = 23 });

相关细节：

protected internal virtual RedirectToRouteResult RedirectToAction(string actionName,   
  string controllerName, RouteValueDictionary routeValues)  
{  
  RouteValueDictionary mergedRouteValues;  
  
  if (RouteData == null)  
  {  
    mergedRouteValues = RouteValuesHelpers.MergeRouteValues(actionName,   
      controllerName, null, routeValues, true /\* includeImplicitMvcValues \*/);  
  }  
  else  
  {  
    mergedRouteValues = RouteValuesHelpers.MergeRouteValues(actionName,   
      controllerName, RouteData.Values, routeValues, true /\* includeImplicitMvcValues \*/);  
  }  
  
  return new RedirectToRouteResult(mergedRouteValues);  
}  
  
public class RedirectToRouteResult : ActionResult  
{  
  public override void ExecuteResult(ControllerContext context)  
  {  
    ...  
  
    string destinationUrl = UrlHelper.GenerateUrl(RouteName, null /\* actionName \*/,   
      null /\* controllerName \*/, RouteValues, Routes, context.RequestContext,   
      false /\* includeImplicitMvcValues \*/);  
  
    ...  
    context.HttpContext.Response.Redirect(destinationUrl, false /\* endResponse \*/);  
  }  
}

可以看到 RedirectToRouteResult.ExecuteResult 中使用 Route 相关信息拼接成目标 Url 后进行跳转。  
  
**方法3**:  
  
RedirectToRoute() 则是直接用 MapRoute 时定义的 Route Name 进行跳转。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return RedirectToRoute("Test2", new { name = "Tom", age = 23 });  
  }  
}

相关细节：

protected internal virtual RedirectToRouteResult RedirectToRoute(string routeName, RouteValueDictionary routeValues)  
{  
  return new RedirectToRouteResult(routeName, RouteValuesHelpers.GetRouteValues(routeValues));  
}

执行过程和 RedirectToAction() 相同。  
  
**5. Json**  
  
Json() 在编写 Ajax 时非常有用，可以将 Entity 等对象序列化成 JSON 格式供 Javascript 使用。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return View();  
  }  
  
  public ActionResult GetUser(string name)  
  {  
    var user = new User { Name = name, Age = 23 };  
    return Json(user);  
  }  
}

Index.aspx

<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage" %>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
  <title>Index</title>  
  <script src="http://www.cnblogs.com/Scripts/jquery-1.3.1.min.js" type="text/javascript"></script>  
  <script type="text/javascript">  
    $(function()  
    {  
      $("#btnTest").click(function()  
      {  
        $.getJSON  
        (  
          "/Test/GetUser",   
          { name: "Tom" },   
          function(json)  
          {  
            alert(json.Name + ";" + json.Age);  
          }  
        );  
      });  
    });  
  </script>  
</head>  
<body>  
  <input type="button" id="btnTest" value="Test" />  
</body>  
</html>

很好用，不是吗？看看相关细节。

protected internal virtual JsonResult Json(object data, string contentType, Encoding contentEncoding)  
{  
  return new JsonResult  
  {  
    Data = data,  
    ContentType = contentType,  
    ContentEncoding = contentEncoding  
  };  
}  
  
public class JsonResult : ActionResult  
{  
  public override void ExecuteResult(ControllerContext context)  
  {  
    ...  
  
    if (Data != null)  
    {  
      JavaScriptSerializer serializer = new JavaScriptSerializer();  
      response.Write(serializer.Serialize(Data));  
    }  
  }  
}

使用 System.Web.Script.Serialization.JavaScriptSerializer 完成 JSON 序列化操作，也就是说我们还可以用 ScriptIgnoreAttribute 排除某些属性。  
  
**6. Javascript**  
  
某 些时候，我们需要根据一些逻辑判断来载入执行不同的 Javascript 代码。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return View();  
  }  
  
  public ActionResult GetJs(int id)  
  {  
    switch (id)  
    {  
      case 1:  
        return JavaScript("alert('Hello, C#!');");        
      case 2:  
        return JavaScript("alert('Hello, MVC!');");        
      default:  
        return null;  
    }  
  }  
}

Index.aspx

<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage" %>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
  <title>Index</title>  
  <script src="http://www.cnblogs.com/Scripts/jquery-1.3.1.min.js" type="text/javascript"></script>  
  <script type="text/javascript">  
    $(function()  
    {  
      $("#btnTest").click(function()  
      {  
        var id = $("#txtId").val();  
        $.getScript("/Test/GetJs/" + id);  
      });  
    });  
  </script>  
</head>  
<body>  
  <input type="text" id="txtId" value="1" />  
  <input type="button" id="btnTest" value="Test" />  
</body>  
</html>

只是这种做法，似乎将 View 和 Controller 的耦合加大了…… 还不如直接用 Javascript 来处理这些。

protected internal virtual JavaScriptResult JavaScript(string script)  
{  
  return new JavaScriptResult { Script = script };  
}  
  
public class JavaScriptResult : ActionResult  
{  
  public override void ExecuteResult(ControllerContext context)  
  {  
    ...  
  
    HttpResponseBase response = context.HttpContext.Response;  
    response.ContentType = "application/x-javascript";  
  
    if (Script != null)  
    {  
      response.Write(Script);  
    }  
  }  
}

**7. File (Download / Upload)**  
  
File() 提供了 Download 功能。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return View();  
  }  
  
  public ActionResult Download(int id)  
  {  
    var filename = String.Format("~/Content/Download/{0}.rar", id);  
    var fileDownloadName = String.Format("{0}.rar", id);  
  
    return File(filename, "application/octet-stream", fileDownloadName);  
  }  
}

当我们在浏览器请求 "/Test/Download/1" 是就会打开下载窗口，同时给出了保存文件名。

protected internal virtual FileContentResult File(byte[] fileContents, contentType, fileDownloadName)  
{  
  return new FileContentResult(fileContents, contentType) { FileDownloadName = fileDownloadName };  
}  
  
public abstract class FileResult : ActionResult  
{  
  public override void ExecuteResult(ControllerContext context)  
  {  
    ...  
    HttpResponseBase response = context.HttpContext.Response;  
    response.ContentType = ContentType;  
    ...  
    WriteFile(response);  
  }  
  
  protected abstract void WriteFile(HttpResponseBase response);  
}  
  
public class FileContentResult : FileResult  
{  
  protected override void WriteFile(HttpResponseBase response)  
  {  
    response.OutputStream.Write(FileContents, 0, FileContents.Length);  
  }  
}

文件上传是另一个常用的 Web 应用。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return View();  
  }  
  
  public ActionResult Upload(HttpPostedFileBase file)  
  {  
    var filename = Server.MapPath("~/Content/Upload/" + Path.GetFileName(file.FileName));  
  
    file.SaveAs(filename);  
    return null;  
  }  
}

Index.aspx

<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage" %>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
  <title>Index</title>  
</head>  
<body>  
  <form action="/Test/Upload" enctype="multipart/form-data" method="post">  
    <input type="file" name="file" />  
    <input type="submit" name="upload" />  
  </form>  
</body>  
</html>

MVC 提供了一个 HttpPostedFileBaseModelBinder 将 Request.Files 的信息直接映射给 Action 同名参数。

public class HttpPostedFileBaseModelBinder : IModelBinder  
{  
  public object BindModel(ControllerContext controllerContext, ModelBindingContext bindingContext)  
  {  
    ...  
  
    HttpPostedFileBase theFile = controllerContext.HttpContext.Request.Files[bindingContext.ModelName];  
  
    // case 1: there was no <input type="file" ... /> element in the post  
    if (theFile == null)  
    {  
      return null;  
    }  
  
    // case 2: there was an <input type="file" ... /> element in the post, but it was left blank  
    if (theFile.ContentLength == 0 && String.IsNullOrEmpty(theFile.FileName))  
    {  
      return null;  
    }  
  
    // case 3: the file was posted  
    return theFile;  
  }  
}

看看一次上传多个文件的演示。

public class TestController : Controller  
{  
  public ActionResult Index()  
  {  
    return View();  
  }  
  
  public ActionResult Upload(HttpPostedFileBase file1, HttpPostedFileBase file2)  
  {  
    var html = String.Format("{0}:{1}<br />{2}:{3}",   
      file1.FileName, file1.InputStream.Length,  
      file2.FileName, file2.InputStream.Length);  
  
    return Content(html);  
  }  
}

Index.aspx

<%@ Page Language="C#" Inherits="System.Web.Mvc.ViewPage" %>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
  <title>Index</title>  
</head>  
<body>  
  <form action="/Test/Upload" enctype="multipart/form-data" method="post">  
    <input type="file" name="file1" />  
    <input type="file" name="file2" />  
    <input type="submit" name="upload" />  
  </form>  
</body>  
</html>

我们上边所看到的Action都是return View();我们可以看作这个返回值用于解析一个aspx文件。而它的返回类型是ActionResult如

public ActionResult Index()  
 {  
 return View();  
 }

除了View（）之外那我们这里还能用于返回什么值呢？

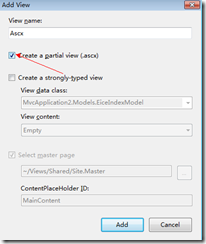
**一、ascx页面**

场景：要返回代码片断，比如Ajax返回一个子页

我们先新建一个Action

public ActionResult Ascx()  
 {  
 return PartialView();  
 }

我们下面再建一个View，仍然是在Action中点右键，AddView。

[](http://kb.cnblogs.com/images/cnblogs_com/chsword/WindowsLiveWriter/ASP.NETMVCActionResult_52D/image_2.png)注意图中勾选。

于是新建了一个ascx页，我们将之少做改写一下

<%@ Control Language="C#" Inherits="System.Web.Mvc.ViewUserControl" %>  
<div>  
得到一个DIV  
</div>

运行，得到页面

[image](http://kb.cnblogs.com/images/cnblogs_com/chsword/WindowsLiveWriter/ASP.NETMVCActionResult_52D/image_4.png)

**二、返回文本**

除了上述情况，有时我们还会仅返回一段文本。

此时我们可以使用以下Action形式：

public ActionResult Text(){  
 return Content("这是一段文本");  
 }

**三、返回Json**

有时我们在调用Ajax时还会要求返回对象为Json序列化的结果，如：

public ActionResult ShowJson()  
 {  
 var m = new EiceIndexModel  
 {  
 Name = "邹健",  
 Sex = true  
 };  
 return Json(m);  
 }

返回文本：

{"Name":"邹健","Sex":true}

**四、输出JS文件**

大多时候js文件都是静态的，但有时js文件可能也要动态生成这时我们可以这样输出

public ActionResult Js()  
 {  
 return JavaScript("var x=0;");  
 }

我们访问之，得到一个正常页面但其Content-Type：application/x-javascript; charset=utf-8

**五、页面跳转**

1.跳转到Url

public ActionResult rdurl()  
 {  
 return Redirect("http://www.baidu.com");  
 }

2.跳转到Action

public ActionResult rdaction()  
 {  
 return RedirectToAction("Index","Eice");  
 }

3.跳转到Routing规则

public ActionResult rdrouting()  
 {  
 return RedirectToRoute("Default",//Route名  
 new{  
 Controller = "Eice",  
 Action = "Index"  
 });  
 }

**六、显示文件**

public ActionResult fn()  
 {  
 return File(  
 "/Content/site.css"//文件路径  
 , "text/css"//文件类型  
 );  
 }

[http://pic.cnblogs.com/face/u107620.jpg](http://home.cnblogs.com/u/stalwart/)

[风黑月高kable](http://home.cnblogs.com/u/stalwart/)  
[关注 - 0](http://home.cnblogs.com/u/stalwart/followees/)  
[粉丝 - 2](http://home.cnblogs.com/u/stalwart/followers/)

[关注博主](javascript:void(0);)

0

0

0

(请您对文章做出评价)

[«](http://www.cnblogs.com/stalwart/archive/2010/06/25/1764872.html) 上一篇：[如何使用ASP.NET MVC的Filter](http://www.cnblogs.com/stalwart/archive/2010/06/25/1764872.html)  
[»](http://www.cnblogs.com/stalwart/archive/2010/06/27/1766091.html) 下一篇：[ASP.NET MVC 1.0 Result 几何？](http://www.cnblogs.com/stalwart/archive/2010/06/27/1766091.html)

posted @ 2010-06-25 14:52 [风黑月高kable](http://www.cnblogs.com/stalwart/) 阅读(430) [评论(0)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html#commentform) [编辑](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html) [收藏](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html) 所属分类: [ASP.NET MVC](http://www.cnblogs.com/stalwart/category/242712.html)

http://www.cnblogs.com/stalwart/aggbug/1765172.html?type=1&webview=1

[刷新评论列表](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html" \l "##)  [刷新页面](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html###)  [返回页首](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html#top)

发表评论 [博客园首页](http://www.cnblogs.com)  [IT新闻](http://news.cnblogs.com)  [闪存](http://home.cnblogs.com/ing/)  [招聘](http://job.cnblogs.com/)  [博问](http://space.cnblogs.com/q/)

昵称：

主页：

评论内容： [写博客、攒园豆、赢大奖](http://zt.cnblogs.com/blogswarriors/)

引用粗体链接缩进图片代码



[不改了](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html###)  [注销](http://passport.cnblogs.com/logout.aspx?ReturnUrl=http%3a%2f%2fwww.cnblogs.com%2fstalwart%2farchive%2f2010%2f06%2f25%2f1765172.html%3flogout%3d1%23commentform)  [订阅回复](http://www.cnblogs.com/UI/Controls/#%23%23)

[使用Ctrl+Enter键快速提交]

0 1765172 MVC Controller与ActionResult的返回值

[IT新闻](http://news.cnblogs.com/):  
· [从“非诚勿扰”看淘宝算法效果测试](http://news.cnblogs.com/n/84481/)  
· [谷歌推出iPhone版谷歌纵横应用](http://news.cnblogs.com/n/84480/)  
· [分析称Facebook挑战谷歌互联网霸主地位](http://news.cnblogs.com/n/84479/)  
· [Google被指责在搜索结果中优先显示其内容](http://news.cnblogs.com/n/84478/)  
· [Twitter发布2010年热门话题排行榜](http://news.cnblogs.com/n/84477/)  
[更多IT新闻...](http://news.cnblogs.com/)

**[知识库最新文章](http://kb.cnblogs.com/" \o "程序员知识库" \t "_blank)**:

[关系型数据库性能优化总结](http://kb.cnblogs.com/page/84200/" \t "_blank)  
[VS2010 调试利器](http://kb.cnblogs.com/page/84180/" \t "_blank)  
[CSS3圆角详解](http://kb.cnblogs.com/page/84160/" \t "_blank)  
[系统架构技能之设计模式—组合模式](http://kb.cnblogs.com/page/84146/" \t "_blank)  
[系统架构技能之设计模式—外观模式](http://kb.cnblogs.com/page/84141/" \t "_blank)

网站导航：  
[博客园首页](http://www.cnblogs.com/" \t "_blank)  [IT新闻](http://news.cnblogs.com/" \t "_blank)  [个人主页](http://home.cnblogs.com/" \t "_blank)  [闪存](http://home.cnblogs.com/ing/" \t "_blank)  [程序员招聘](http://job.cnblogs.com/" \t "_blank)  [社区](http://space.cnblogs.com/" \t "_blank)  [博问](http://space.cnblogs.com/q/" \t "_blank)

[](http://www.china-pub.com/static07/0908/zh_pod_090825.asp)  
[China-pub 计算机图书网上专卖店！6.5万品种2-8折！](http://www.china-pub.com/itbook/" \t "_blank)  
[China-Pub 计算机绝版图书按需印刷服务](http://www.china-pub.com/static07/0901/zh_jueba_090121.asp" \t "_blank)

链接：[切换模板](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html?switchskin=1" \l "skinlist)

**相关搜索:**  
[ASP.NET MVC](http://zzk.cnblogs.com/s?w=ASP.NET+MVC" \t "_blank)

**最简洁阅读版式：**  
[MVC Controller与ActionResult的返回值](http://archive.cnblogs.com/a/1765172/)

**公告**

[粉丝 - 2](http://home.cnblogs.com/u/stalwart/followers/)  
[关注 - 0](http://home.cnblogs.com/u/stalwart/followees/)  
[我的主页](http://home.cnblogs.com/u/stalwart/" \t "_blank)  [个人资料](http://home.cnblogs.com/u/stalwart/detail/" \t "_blank)  
[我的闪存](http://home.cnblogs.com/stalwart/ing/" \t "_blank)  [发短消息](http://space.cnblogs.com/msg/send/%e9%a3%8e%e9%bb%91%e6%9c%88%e9%ab%98kable" \t "_blank)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | [<](javascript:__doPostBack('Calendar1$entryCal','V3773')) | 2010年6月 | [>](javascript:__doPostBack('Calendar1$entryCal','V3834')) | | | | | | | |
| **日** | **一** | **二** | **三** | **四** | **五** | **六** |
| 30 | 31 | 1 | [2](http://www.cnblogs.com/stalwart/archive/2010/6/2.html) | [3](http://www.cnblogs.com/stalwart/archive/2010/6/3.html) | 4 | [5](http://www.cnblogs.com/stalwart/archive/2010/6/5.html) |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | [21](http://www.cnblogs.com/stalwart/archive/2010/6/21.html) | 22 | 23 | [24](http://www.cnblogs.com/stalwart/archive/2010/6/24.html) | [25](http://www.cnblogs.com/stalwart/archive/2010/6/25.html) | 26 |
| [27](http://www.cnblogs.com/stalwart/archive/2010/6/27.html) | 28 | 29 | 30 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |

**搜索**





**常用链接**

* [我的随笔](http://www.cnblogs.com/stalwart/MyPosts.html)
* [我的空间](http://home.cnblogs.com/stalwart/)
* [我的短信](http://space.cnblogs.com/msg/recent)
* [我的评论](http://www.cnblogs.com/stalwart/MyComments.html)
* [更多链接](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [我的参与](http://www.cnblogs.com/stalwart/OtherPosts.html" \o "我发表过评论的随笔)
* [我的新闻](http://www.cnblogs.com/stalwart/MyNews.html)
* [最新评论](http://www.cnblogs.com/stalwart/RecentComments.html)
* [我的标签](http://www.cnblogs.com/stalwart/tag/)

**随笔分类**

* [ASP.NET MVC(23)](http://www.cnblogs.com/stalwart/category/242712.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [C#(9)](http://www.cnblogs.com/stalwart/category/245051.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [JQuery+JS(13)](http://www.cnblogs.com/stalwart/category/245605.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [Linux(1)](http://www.cnblogs.com/stalwart/category/264407.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [MAC OS(1)](http://www.cnblogs.com/stalwart/category/265524.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [MySql](http://www.cnblogs.com/stalwart/category/266016.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [Ruby on Rails(17)](http://www.cnblogs.com/stalwart/category/263641.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [SQL SERVER(4)](http://www.cnblogs.com/stalwart/category/246829.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [WebDesign(6)](http://www.cnblogs.com/stalwart/category/245977.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [随笔(1)](http://www.cnblogs.com/stalwart/category/251145.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [系统(1)](http://www.cnblogs.com/stalwart/category/245112.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)

**随笔档案**

* [2010年12月 (7)](http://www.cnblogs.com/stalwart/archive/2010/12.html)
* [2010年11月 (7)](http://www.cnblogs.com/stalwart/archive/2010/11.html)
* [2010年10月 (17)](http://www.cnblogs.com/stalwart/archive/2010/10.html)
* [2010年9月 (7)](http://www.cnblogs.com/stalwart/archive/2010/09.html)
* [2010年8月 (2)](http://www.cnblogs.com/stalwart/archive/2010/08.html)
* [2010年7月 (9)](http://www.cnblogs.com/stalwart/archive/2010/07.html)
* [2010年6月 (14)](http://www.cnblogs.com/stalwart/archive/2010/06.html)
* [2010年5月 (10)](http://www.cnblogs.com/stalwart/archive/2010/05.html)
* [2010年4月 (1)](http://www.cnblogs.com/stalwart/archive/2010/04.html)
* [2010年3月 (1)](http://www.cnblogs.com/stalwart/archive/2010/03.html)
* [2010年1月 (2)](http://www.cnblogs.com/stalwart/archive/2010/01.html)

**文章分类**

* [ASP.NET(5)](http://www.cnblogs.com/stalwart/category/227255.html) [(rss)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)

**MySql**

* [MySql5.1在线手册](http://imysql.cn/docs/MySQL_51_zh/mysql_51.html)

**Ruby On Rails**

* [Rails 版本变迁](http://www.javaeye.com/topic/724718)
* [Rails3](http://ihower.tw/rails3/)

**博友**

* [山贼](http://thief-world.blogbus.com/)
* 苏州UCD领航者!

**技术链接**

* [scottgu博客](http://blog.joycode.com/scottgu/)
* .NET 开发主管scottgu博客
* [吕国宁](http://lvguoning.com/)

**最新评论[http://www.cnblogs.com/images/xml.gif](http://www.cnblogs.com/stalwart/CommentsRSS.aspx)**

[1. Re:Rails Model验证之强大](http://www.cnblogs.com/stalwart/archive/2010/10/29/1864279.html" \l "1946301)

不错，的确很方便 (版主)

[2. Re:ASP.NET MVC 2.0 Html.RenderPartial & Html.RenderAction](http://www.cnblogs.com/stalwart/archive/2010/08/24/1807698.html" \l "1920664)

是的就如上面的图画的一样。。因为有时候PartialView要处理数据的，在实际应用中RanderAction用的比较多！ (华网)

[3. Re:ASP.NET MVC 2.0 Html.RenderPartial & Html.RenderAction](http://www.cnblogs.com/stalwart/archive/2010/08/24/1807698.html" \l "1910442)

使用RenderAction调用action,然后action返回数据给对应的ascx页面? 是这样的顺序码? (香炮)

[4. Re:C# 可空类型和空接合运算符(??)](http://www.cnblogs.com/stalwart/archive/2010/07/08/1773841.html" \l "1866891)

童鞋， 知道这个方法不？ int? x1 = GetNullableType(); 我在用的时候怎么都找不到这个东东。如果知道，帮忙回复下我哦： (类型安全的心)

**阅读排行榜**

* [1. MVC Controller与ActionResult的返回值(430)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1765172.html)
* [2. Win7环境下安装Mac OS双系统及Iphone开发SDK(419)](http://www.cnblogs.com/stalwart/archive/2010/10/15/1852450.html)
* [3. jQuery getJSON()函数及getScript()函数(289)](http://www.cnblogs.com/stalwart/archive/2010/07/15/1778481.html)
* [4. 如何使用jQuery向asp.net Mvc传递复杂json数据-ModelBinder篇(288)](http://www.cnblogs.com/stalwart/archive/2010/07/17/1779305.html)
* [5. 理解ASP.NET MVC Framework Action Filters（翻的）(282)](http://www.cnblogs.com/stalwart/archive/2010/06/25/1764870.html)

**评论排行榜**

* [1. ASP.NET MVC 2.0 Html.RenderPartial & Html.RenderAction(2)](http://www.cnblogs.com/stalwart/archive/2010/08/24/1807698.html)
* [2. C# 可空类型和空接合运算符(??)(1)](http://www.cnblogs.com/stalwart/archive/2010/07/08/1773841.html)
* [3. Rails Model验证之强大(1)](http://www.cnblogs.com/stalwart/archive/2010/10/29/1864279.html)
* [4. Rails 2.0.2用scaffold来生成简单实例(0)](http://www.cnblogs.com/stalwart/archive/2010/10/26/1861180.html)
* [5. Rails安装(0)](http://www.cnblogs.com/stalwart/archive/2010/10/24/1859615.html)

**60天内阅读排行**

* [1. Win7环境下安装Mac OS双系统及Iphone开发SDK(419)](http://www.cnblogs.com/stalwart/archive/2010/10/15/1852450.html)
* [2. ASP.NET MVC 3 Beta初体验之实用的WebMail【附示例下载】(185)](http://www.cnblogs.com/stalwart/archive/2010/11/16/1878528.html)
* [3. 利用google翻译解决URL可读性(107)](http://www.cnblogs.com/stalwart/archive/2010/11/15/1877767.html)
* [4. Ruby Gems（1）–简要介绍和ruby on rails安装(94)](http://www.cnblogs.com/stalwart/archive/2010/10/24/1859594.html)
* [5. Rails Model验证之强大(73)](http://www.cnblogs.com/stalwart/archive/2010/10/29/1864279.html)

Copyright ©2010 风黑月高kable

窗体底端