ASP.NET MVC 使用视图引擎实现页面静态化

2015-05-30 十有三 0 浏览:1440 .NET技术 (/c1) ASP.NET MVC (/t22)

本文主要分享了在ASP.NET MVC中，使用视图做为静态模板，从而对网页进行静态化操作的方法。此方法需要对MVC视图引擎的相关知识有所了解，博文中会贴出一些资料以供参考，文章也会分享演示项目的下载地址。

以往常见的静态化方式是使用HTML页面作为静态化模板，但对开发人员来说将会添加额外的工作量，毕竟同时维护模板和动态页面，是十分烦人的任务。但是如果直接使用ASP.NET MVC本身的视图作为模板，由于使用了视图，开发起来将会更简单。

使用视图引擎进行静态化操作，主要使用到了两个类：ViewEngines类 (https://msdn.microsoft.com/zh-cn/library/system.web.mvc.viewengines(v=vs.118).aspx)和ViewContext类 (https://msdn.microsoft.com/zh-cn/library/system.web.mvc.viewcontext(v=vs.118).aspx)。通过ViewEngines类的FindView方法或FindPartialView方法找到要进行静态化的视图页面，然后使用ViewContext类将数据模型填充到视图模板中，获取页面内容后生成静态页面。

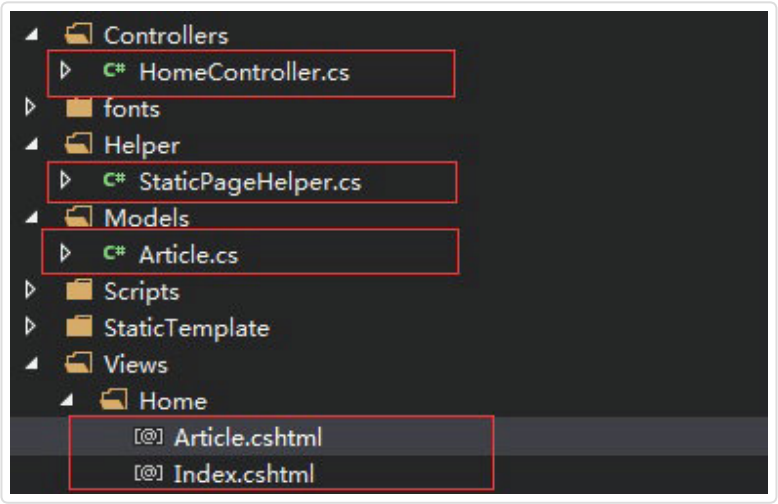
ViewContext类主要是封装与呈现视图相关的信息。

这里引用ASP.NET MVC 5 高级编程的相关资料：

1. 视图引擎是一个静态的ViewEngineCollection类型对象，可以包含所有已注册的视图引擎。
2. FindView方法迭代ViewEngineCollection中注册的视图引擎，并在每个视图引擎上调用FindView方法，并把视图名称作为参数传入。这就是ViewEngineCollection询问每个视图引擎能否渲染指定视图的方式。
3. FindPartialView方式的工作机制与FindView几乎一样，只是它关注于查找部分视图。

演示项目下载地址：ASP.NET MVC Static Operation.zip (http://pan.baidu.com/s/1dD0RR3N)（项目需要使用VS2013或者更高版本的VS编译器打开）

接下来看下演示项目的大概结构：



静态化帮助类StaticPageHelper：

```
/*
 * 该类需要引用下列的命名空间
 * using System;
 * using System.Web;
 * using System.Text;
 * using System.IO;
 * using System.Web.Mvc;
 */

/// <summary>
/// 创建时间：2013-11-06
/// 创建者：shiyousan.com
/// 描述：静态页面生成帮助类
/// </summary>
public class StaticPageHelper
{
    /// <summary>
    /// 根据View视图生成静态页面
    /// </summary>
    /// <param name="strStaticPageAbsolutePath">存放静态页面所在绝对路径</param>
    /// <param name="context">ControllerContext</param>
    /// <param name="strViewName">视图名称</param>
    /// <param name="strMasterName">模板视图名称</param>
    /// <param name="model">参数实体模型</param>
    /// <param name="strMessage">返回信息</param>
    /// <param name="isPartial">是否分布视图</param>
    /// <returns>生成成功返回true,失败false</returns>
    public static bool GenerateStaticPage(string strStaticPageAbsolutePath, Con
trollerContext context, string strViewName, string strMasterName, object model,
 out string strMessage, bool isPartial = false)
    {
        bool isSuccess = false;
        try
        {
            //创建存放静态页面目录
            if (!Directory.Exists(Path.GetDirectoryName(strStaticPageAbsolutePa
th)))
            {
                Directory.CreateDirectory(Path.GetDirectoryName(strStaticPageAb
solutePath));
            }
            //删除已有的静态页面
            if (File.Exists(strStaticPageAbsolutePath))
            {
                File.Delete(strStaticPageAbsolutePath);
            }

            ViewEngineResult result = null;
            if (isPartial)
            {
                result = ViewEngines.Engines.FindPartialView(context, strViewNa
me);
            }
            else
            {
                result = ViewEngines.Engines.FindView(context, strViewName, str
MasterName);
```

```
            }

            if (model != null)
            {
                context.Controller.ViewData.Model = model;
            }

            /*
             * 设置临时数据字典作为静态化标识
             * 可以在视图上使用TempData["IsStatic"]来控制某些元素显示。
             */
            if (!context.Controller.TempData.ContainsKey("IsStatic"))
            {
                context.Controller.TempData.Add("IsStatic", true);
            }

            if (result.View != null)
            {
                using (var sw = new StringWriter())
                {
                    string strResultHtml = string.Empty;
                    //填充数据模型到视图中，并获取完整的页面
                    ViewContext viewContext = new ViewContext(context, result.V
iew, context.Controller.ViewData, context.Controller.TempData, sw);
                    result.View.Render(viewContext, sw);
                    strResultHtml = sw.ToString();
                    //通过IO操作将页面内容生成静态页面
                    File.WriteAllText(strStaticPageAbsolutePath, strResultHtm
l);
                    strMessage = string.Format("生成静态页面成功！存放路径：{0}", st
rStaticPageAbsolutePath);
                    isSuccess = true;
                }
            }
            else
            {
                isSuccess = false;
                strMessage = "生成静态页面失败！未找到视图！";
            }

        }
        catch (IOException ex)
        {
            strMessage = ex.Message;
            isSuccess = false;
        }
        catch (Exception ex)
        {
            strMessage = ex.Message;
            isSuccess = false;
        }
        return isSuccess;
    }
}
```

该类可帮助我们迅速的使用视图引擎进行静态化操作。

文章模型类：

```
public class Article
{
    /// <summary>
    /// 文章标题
    /// </summary>
    public string Title { get; set; }
    /// <summary>
    /// 文章内容
    /// </summary>
    public string Content { get; set; }
    /// <summary>
    /// 文章作者
    /// </summary>
    public string Authur { get; set; }
    /// <summary>
    /// 发布日期
    /// </summary>
    public DateTime InputDate { get; set; }
}
```

控制器代码：

```
/// <summary>
/// 使用视图引擎静态化
/// </summary>
/// <returns></returns>
[HttpPost]
public ActionResult UseViewEngineStatic()
{
    string strMessage = string.Empty;
    Article entity = GetArticleModel(true);

    //保存静态页面的绝对路径
    string strStaticPageAbsolutePath = GetStaticPageAbsolutePath();
    //生成静态页面,其中的Article是视图名称
    StaticPageHelper.GenerateStaticPage(strStaticPageAbsolutePath, ControllerCo
ntext, "Article", null, entity, out strMessage);

    return Content("使用视图引擎实现页面静态化-----"+strMessage);
}
/// <summary>
/// 文章内容展示页
/// </summary>
/// <returns></returns>
public ActionResult Article()
{
    Article entity = GetArticleModel();
    return View(entity);
}

#region 自定义方法
/// <summary>
/// 获取文章数据模型，一般是要从数据库查询
/// </summary>
/// <param name="isViewEngine">是否使用视图引擎</param>
/// <returns></returns>
private Article GetArticleModel(bool isViewEngine=false)
{
    Article entity = new Article();
    entity.Title = "ASP.NET MVC静态化DEMO";
    if (isViewEngine)
    {
        entity.Title += "----------------使用视图引擎";
    }
    else
    {
        entity.Title += "----------------使用HTML模板";
    }
    entity.Authur = "十有三";
    entity.InputDate = DateTime.Now;
    entity.Content = "<p>分别演示了使用HTML模板和视图引擎这两种方式生成静态页面。</p>";
    return entity;
}

/// <summary>
/// 获取保存静态页面绝对路径
/// </summary>
/// <returns></returns>
private string GetStaticPageAbsolutePath()
```

```
{
    //静态页面名称
    string strStaticPageName = string.Format("{0}.html", DateTime.Now.Ticks.ToS
tring());
    //静态页面相对路径
    string strStaticPageRelativePath = string.Format("article\\{0}\\{1}", DateT
ime.Now.ToString("yyyy/MM").Replace('/', '\\'), strStaticPageName);
    //静态页面完整路径
    string strStaticPageAbsolutePath = AppDomain.CurrentDomain.BaseDirectory +
 strStaticPageRelativePath;
    return strStaticPageAbsolutePath;
}

#endregion
```

UseViewEngineStatic操作方法主要调用静态帮助类进行静态化操作，并返回操作结果信息。

Article操作方法主要是获取文章数据模型，选择文章视图呈现页面到浏览器上。

操作界面视图代码（Home/Index.cshtml）：

```
@Ajax.ActionLink("使用视图引擎实现页面静态化", "UseViewEngineStatic", "Home", null,
new AjaxOptions() { HttpMethod = "POST", OnSuccess = "OnSuccess" }, new { @clas
s = "btn btn-primary btn-lg" })
<div id="divMessage">
</div>
@section scripts{
    <script type="text/javascript">
        function OnSuccess(message) {
            var msg = "<div class=\"alert alert-info\" role=\"alert\">"+message
+"</div>";
            $("#divMessage").prepend(msg);
        }
    </script>
}
```

主要是向Home控制器下的UseViewEngineStatic操作方法发送一个AJAX请求，从而进行静态化操作。

PS:由于使用了Ajax.ActionLink，记得引用jquery.unobtrusive-ajax.js

文章视图模板（/Home/Article.cshtml，就是显示文章内容的视图页）：

```
@model ASP.NET_MVC_Static_Operation.Models.Article
@{
    Layout = null;
}

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>静态模板页面</title>
</head>
<body>
    <h1>@Model.Title</h1>
    <hr />
    <div>文章作者：@Model.Authur | 发布时间： @Model.InputDate</div>
    <hr />
    <div>@Html.Raw(Model.Content)</div>
</body>
</html>
```

文章的视图页面，也是要进行静态化的模板。

上面的例子都是使用Razor视图引擎，asp.net mvc 3以来Razor已经成为默认的视图引擎。

作者：十有三 **(http://shiyousan.com)**

出处：**http://shiyousan.com/post/635685973774173122 (http://shiyousan.com/post/635685973774173122)**

上一篇: SQL Server Express不支持维护计划功能 (/post/635679187188692480)

下一篇: ASP.NET MVC使用HTML模板进行静态化操作 (/post/635686037177640853)

发表评论

*昵称:

(必填)

邮箱:

(可选)：用于评论回复通知。

网址:

(可选)：方便交流回访

*评论:

(必填)：最多可输入1000个字。

记住昵称

发表评论

关于十有三

这是一个程序猿的独立博客，主要分享与编程技术有关的内容，包括.NET、数据库、WEB前端、网站开发与建设、各种开发工具和插件等。 这里也是博主记录工作经验和生活感悟的地方。

PS:博主是一名.NET程序员，也是一名草根站长，目前主要从事ASP.NET网站开发工作，更多信息请点击关于本站 (/Home/About)。

文章分类

.NET技术 97 (/c1) 数据库 22 (/c2) Web前端 19 (/c3) 网站建设 27 (/c4)

玄学哲学 2 (/c5) 操作系统与应用 42 (/c7) 程序猿日常 19 (/c8)

开发工具 4 (/c9) 其他随笔 7 (/c6)

文章标签

ASP.NET (/t11) ASP.NET MVC (/t22) C# (/t23) CSS (/t34) HTML (/t14)

Javascript (/t13) MongoDB (/t33) SQL (/t27) SQL Server (/t5) Visual Studio (/t17)

Windows系统 (/t10) 版本控制系统 (/t32) 插件工具 (/t21) 道学 (/t20) 佛经 (/t7)

服务器 (/t29) 搞笑娱乐 (/t6) 好文分享 (/t30) 软件应用 (/t31) 生活知识 (/t28)

诗词 (/t8) 手机问题 (/t25) 随笔 (/t24) 网络知识 (/t26) 网站设计优化 (/t16)

网站维护 (/t4) 养生保健 (/t12) 异常处理 (/t15) 游戏攻略 (/t18)

最新评论

- 是使用PHP开发的网站吗？虽然我对PHP不熟悉，但 (/post/63562991038336867 5#cm2…
- 你好，我设置了URL重写（实现了用www.123. (/post/63562991038336867 5#cm209)
- 你好，我设置了URL重写（实现了用www.123. (/post/63562991038336867 5#cm208)
- 哟西，就是这样解决的~ (/post/63539856879980682 7#cm207)
- 说的方法反反复复凤飞飞反复反复 (/post/63538815510028738 0#cm206)
- 我新装了2012也是找不到，原来是这样 (/post/63567918718869248 0#cm205)
- 蛋疼的问题啊，同样更新不动，用vpn也没用 (/post/63575675938363105 1#cm204)
- 有用 good (/post/63538302586100458 5#cm203)
- 网址式？估计要网上找找了，我目前还没接触过类似的 (/post/63562991038336867 5#cm…
- 嗯嗯 感激！ 刚刚我也找到window._bd_s (/post/63562991038336867 5#cm201)
- 百度分享插件如果要与AJAX结合使用，建议看下看下 (/post/63562991038336867 5#cm…
- 找到问题了` 根据您的配置，更多里的按钮没有dat (/post/63562991038336867 5#cm198)
- 这个是范例的BUG，已经修复代码了，在打开范例看下 (/post/63562991038336867 5#cm..
- 您好，看了您的文章和demo，发现微信分享的url (/post/63562991038336867 5#cm196)
- 111 (/post/63597809938139192 3#cm195)

推荐文章

ASP.NET MVC使用HTML模板进行静...
十有三,十有三的个人博客,独立博客,
ASP.NET MVC 页面静态化操作的思路
思考ASP.NET网站静态化的利与弊
此请求的 URL 的长度超过配置的 ma...
[ASP.NET MVC]标签
ASP.NET MVC 表单提交数组和泛型...
ASP.NET MVC全局异常处理和捕获的..
ASP.NET MVC 5 学习笔记：使用Ha...
ASP.NET MVC中注册Global.asax的...

百度推荐

站点统计

文章总数：238
评论总数：105
浏览总数：357888
本月文章：0

友情链接

月光清城 (http://guoms.com) Wil的博客 (http://www.94will.com) Passingwind的博客 (http://blog.wuliping.cn)