## ASP.NET MVC使用HTML模板进行静态化操作

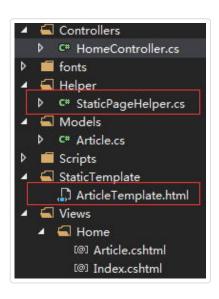
**iii** 2015-05-30 **1** 十有三 **1** 0 **3** 浏览:812 **ii** .NET技术 (/c1) **3** ASP.NET MVC (/t22)

使用html模板加上自定义占位符是常见的页面静态化方法,本文将分享在asp.net mvc项目中使用静态模板进行静态化的具体代码。其实具体的实现原理还是蛮简单的,这种方法需要编写对应的html模板页,模板中需要进行数据替换的地方定义好自定义占位符,最后在生成HTML静态页面的时候,将自定义占位符替换成要显示的数据。

比如模板页面中,定义了一个@template\_placeholder\_articlecontent占位符(表示文章的内容),我们就可以使用string.Replace方法将占位符标识替换成所需的文章内容,或者使用正则表达式进行替换。将数据填充到模板后,通过IO操作将网页内容保存为html页面。

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)

先看下演示项目的代码结构:



其中最主要的两个文件分别是ArticleTemplate.html(静态模板)和StaticPageHelper.cs(静态化帮助类)。另外还有Article模型对象,Home控制器和对应的视图(主要控制和显示操作),下面会仔细说明。

StaticPageHelper代码:

```
†
†
```

```
/// <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>
   /// 生成静态页面
   /// </summary>
   /// <param name="strStaticPageAbsolutePath">生成静态页面的名称</param>
   /// <param name="strStaticPageRelativeDirectory">静态页面所在相对目录</param>
   /// <param name="strStaticPageBasicPath">静态页面所在基础路径</param>
   /// <param name="strTemplateAbsolutePath">静态页面模板文件路径</param>
   /// <param name="arrPlaceholder">占位符数组</param>
   /// <param name="arrReplaceContent">要替换的内容数组</param>
   public static bool GenerateStaticPage(string strStaticPageAbsolutePath, str
ing strStaticPageRelativeDirectory, string strStaticPageBasicPath, string strTe
mplateAbsolutePath, string[] arrPlaceholder, string[] arrReplaceContent, out st
ring strMessage)
    {
       string strStaticPageBaseDirectory = string.Format("{0}\\{1}", strStatic
PageBasicPath.TrimEnd('\\'), strStaticPageRelativeDirectory.TrimEnd('\\'));
       return GenerateStaticPage(strStaticPageAbsolutePath, strStaticPageBaseD
irectory, strTemplateAbsolutePath, arrPlaceholder, arrReplaceContent, out strMe
ssage);
   }
   /// <summary>
   /// 生成静态页面
   /// </summary>
   /// <param name="strStaticPageAbsolutePath">生成静态页面的名称</param>
   /// <param name="strStaticPageDirectory">静态页面所在基础目录</param>
   /// <param name="strTemplateAbsolutePath">静态页面模板文件路径</param>
   /// <param name="arrPlaceholder">占位符数组</param>
   /// <param name="arrReplaceContent">要替换的内容数组</param>
   public static bool GenerateStaticPage(string strStaticPageAbsolutePath, str
ing strStaticPageBaseDirectory, string strTemplateAbsolutePath, string[] arrPla
ceholder, string[] arrReplaceContent, out string strMessage)
    {
       //获取静态页面完整路径
       string strStaticPageFullPath = string.Format("{0}\\{1}", strStaticPageB
aseDirectory, strStaticPageAbsolutePath);
       return GenerateStaticPage(strStaticPageAbsolutePath, strTemplateAbsolut
ePath, arrPlaceholder, arrReplaceContent, out strMessage);
   }
   /// <summary>
   /// 根据静态的HTML页面模板生成静态页面
   /// </summary>
   /// <param name="strStaticPageAbsolutePath">存放静态页面所在绝对路径</param>
   /// <param name="strTemplateAbsolutePath">静态模板页面的绝对路径</param>
   /// <param name="arrPlaceholder">占位符数组</param>
   /// <param name="arrReplaceContent">要替换的内容数组</param>
   /// <param name="strMessage">返回信息</param>
   /// <returns>生成成功返回true,失败false</returns>
   public static bool GenerateStaticPage(string strStaticPageAbsolutePath, str
ing strTemplateAbsolutePath, string[] arrPlaceholder, string[] arrReplaceConten
```

```
†
†
```

```
t, out string strMessage)
    {
       bool isSuccess = false;
       try
       {
           //生成存放静态页面目录
           if (!Directory.Exists(Path.GetDirectoryName(strStaticPageAbsolutePa
th)))
           {
               Directory.CreateDirectory(Path.GetDirectoryName(strStaticPageAb
solutePath));
           //验证占位符
           if (arrPlaceholder.Length != arrReplaceContent.Length)
               strMessage = string.Format("生成静态页面失败,占位符数组个数和替换内容
个数不相等! 存放路径为: {0}", strStaticPageAbsolutePath);
               return false;
           }
           //生成存放静态页面文件
           if (File.Exists(strStaticPageAbsolutePath))
           {
               File.Delete(strStaticPageAbsolutePath);
           //获取模板HTML
           StringBuilder strHtml = new StringBuilder();
           strHtml.Append(File.ReadAllText(strTemplateAbsolutePath, Encoding.U
TF8));
           //替换模板占位符,获取要生成的静态页面HTML
           for (int i = 0; i < arrPlaceholder.Length; <math>i++)
               strHtml.Replace(arrPlaceholder[i], arrReplaceContent[i]);
           }
           //生成静态页面
           File.WriteAllText(strStaticPageAbsolutePath, strHtml.ToString());
           strMessage = string.Format("生成静态页面成功! 存放路径: {0}", strStaticP
ageAbsolutePath);
           isSuccess = true;
       catch (IOException ex)
           strMessage = ex.Message;
           isSuccess = false;
       catch (Exception ex)
           strMessage = ex.Message;
           isSuccess = false;
       return isSuccess;
   }
}
```

该类主要用于帮助我们快速简单的进行静态化操作,GenerateStaticPage方法先使用File.Rea dAllText方法获取HTML模板的字符串,然后通过使用循环将数据的内容数组与占位符数组进行匹配并填充替换,完成将占位符替换成数据的操作:

```
//替换模板占位符,获取要生成的静态页面HTML
for (int i = 0; i < arrPlaceholder.Length; i++)
{
    strHtml.Replace(arrPlaceholder[i], arrReplaceContent[i]);
}</pre>
```

arrPlaceholder参数表示占位符数组,arrReplaceContent表示要替换的内容数组,这两个数组的长度一定要一致,因为是一一对应的!使用循环对这两个数字进行匹配和替换操作,这样的好处是不会限制模板上面自定义占位符的个数,可以随着需求而添加。

最后将数据替换完的页面通过File.WriteAllText写入到磁盘中,到了这一步整个静态化操作就完成了。

#### ArticleTemplate.html代码:

HTML的静态模板页面,根据个人不同的需求添加自定义占位符。

根据上面的代码,我们在视图上调用UseHtmlTemplateStatic操作方法进行静态化:



```
†
†
```

```
/// <summary>
/// 使用HTML模板静态化
/// </summarv>
/// <returns></returns>
[HttpPost]
public ActionResult UseHtmlTemplateStatic()
{
   string strMessage = string.Empty;
   //静态页面模板路径
   string strTemplateFullPath = string.Format("{0}StaticTemplate/{1}", AppDoma
in.CurrentDomain.BaseDirectory, "ArticleTemplate.html");
   //保存静态页面的绝对路径
   string strStaticPageAbsolutePath = GetStaticPageAbsolutePath();
   //获取模板占位符数组
   string[] arrPlaceholder = new string[4];
   arrPlaceholder[0] = "@template_placeholder_title";
   arrPlaceholder[1] = "@template_placeholder_author";
   arrPlaceholder[2] = "@template_placeholder_date";
   arrPlaceholder[3] = "@template_placeholder_content";
   //获取填充到模板中的占位符(自定义标识)所对应的数据数组
   Article entity = GetArticleModel();
   string[] arrReplaceContent = new string[4];
   arrReplaceContent[0] = entity.Title;
   arrReplaceContent[1] = entity.Authur;
   arrReplaceContent[2] = entity.InputDate.ToString("yyyy/MM/dd");
   arrReplaceContent[3] = entity.Content;
   StaticPageHelper.GenerateStaticPage(strStaticPageAbsolutePath, strTemplateF
ullPath, arrPlaceholder, arrReplaceContent, out strMessage);
   return Content("使用HTML模板生成静态页面----" + strMessage);
}
#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 = "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。";
   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;
}
```

可以看到我们传入了模板的路径、存放静态页面的路径,占位符数组和对应的数据数组,最后成功的生成了静态页面。更加具体的代码可以下载演示项目 (http://pan.baidu.com/s/1jGDxlAi)

作者: 十有三 (http://shiyousan.com)

出处: http://shiyousan.com/post/635686037177640853 (http://shiyousan.com/post/635686037177640853)

欢迎转载本文,本文版权归作者所有,转载请声明出处或保留此段声明。**^\_^**请尊重他人劳动成果,共建美好的网络环境。

▲ 上一篇: ASP.NET MVC 使用视图引擎 实现页面静态化 (/post/63568597377417 3122) ▼ 下一篇: ASP.NET MVC 页面静态化操作的思路 (/post/635686634025340822)

## 发表评论

### 

(必填)

#### ₩邮箱:

(可选): 用于评论回复通知。

#### **多** 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

### 它 发表评论

# **6** 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.NE T、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 98 (/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

- 遇到同样的问题 (/post/635756759383631051#cm214)
- ^.^ 谢谢支持 (/post/635646254870261696#cm213)
- ₱ 相见恨晚啊,好网站,已经收藏了,继续支持,我也是. (/post/635646254870261696#c...
- 是使用PHP开发的网站吗?虽然我对PHP不熟悉,但 (/post/635629910383368675#cm2...
- **■** 你好,我设置了URL重写(实现了用www.123. (/post/635629910383368675#cm209)
- **呷**你好,我设置了URL重写(实现了用www.123. (/post/635629910383368675#cm208)
- 🗭 哟西,就是这样解决的~ (/post/635398568799806827#cm207)
- ₩ 说的方法反反复复凤飞飞反复反复 (/post/635388155100287380#cm206)
- ➡ 我新装了2012也是找不到,原来是这样 (/post/635679187188692480#cm205)
- ➡ 蛋疼的问题啊,同样更新不动,用vpn也没用 (/post/635756759383631051#cm204)
- **呷** 有用 good (/post/635383025861004585#cm203)
- 网址式?估计要网上找找了,我目前还没接触过类似的 (/post/635629910383368675#cm...
- 嗯嗯 感激! 刚刚我也找到window.\_bd\_s (/post/635629910383368675#cm201)
- ➡ 百度分享插件如果要与AJAX结合使用,建议看下看下 (/post/635629910383368675#cm...
- **■** 找到问题了`根据您的配置,更多里的按钮没有dat (/post/635629910383368675#cm198)

## **分** 推荐文章

■ ASP.NET MVC 使用视图引擎实现页...

- ■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
- ASP.NET MVC 出现"System.Web.M...
- ASP.NET MVC 5中的FilterConfig类
- C#如何根据输入的行数打印出由\*星...
- [ASP.NET MVC]标签
- ASP.NET MVC出现XML5632仅允许...
- ASP.NET MVC实现IExceptionFilter接..
- ASP.NET MVC 表单提交数组和泛型...

百度推荐

## ■■站点统计

▶ 文章总数: 239

▶ 评论总数: 108

● 浏览总数: 373738

■ 本月文章: 1



### **⑤** 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)



(http://creativecommons.org/licenses/by-sa/4.0/)本站作品采用知识共享署名-相同方式共享 4.0

国际许可协议 (http://creativecommons.org/licenses/by-sa/4.0/)进行许可。

闽ICP备15003702号 (http://www.miitbeian.gov.cn/)

(http://www.cnzz.com/stat/website.php?web\_id=1000466525)