

# ASP.NET MVC 页面静态化操作的思路

本文主要讲述了在asp.net mvc中，页面静态化的几种思路和方法。对于网站来说，生成纯html静态页面除了有利于seo外，还可以减轻网站的负载能力和提高网站性能。在asp.net mvc中，视图的Razor引擎和控制器为我们静态化提供了新的思路和方法，我们可以利用视图引擎和重写过滤器进行静态操作。

先说下本人目前已知的静态化思路，大概有4种方法：

## 一、使用html模板搭配自定义占位符标识

这是最常见的方法，这种方法需要先编写一个html模板页，模板中定义好需要替换的占位符标识，在需要生成HTML页面的时候，将自定义的占位符标识替换成要展现的内容。比如模板html页面中，有个@template\_placeholder\_articlecontent标识（表示文章的内容），我们就可以使用string.Replace方法将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。

最后将生成的静态网页内容通过IO保存为html页面，具体的实现步骤可以看下这篇文章：[ASP.NET MVC使用HTML模板进行静态化操作](#)

## 二、使用ASP.NET MVC的视图引擎生成静态HTML页面

这种方式是我比较推荐的方法，因为可以使用视图文件作为静态模板，还可搭配布局页来使用，这样动态界面和静态界面都是同一套模板，减少了额外编写静态模板的时间！并且在静态化时，填充数据也十分简单，依旧是使用模型数据！

大概说下思路，首先使用ViewEngines类找到要静态化的View视图，然后通过ViewContext类将数据模型填充到视图中，再使用ViewContext.View.Render(ViewContext viewContext, TextWriter writer)方法将获取视图页面的输出流，最后通过File.WriteAllText(string path, string contents)方法将页面内容写入到磁盘中。

我另外写了一篇文章来详细的说明：[ASP.NET MVC 使用视图引擎实现页面静态化](#)。毕竟这种方式需要涉及到多个MVC特性的使用，文章中会分享一个已经写好的帮助类，可以很容易的进行调用和操作。

PS:外国猿类同胞的思路：[Rendering ASP.NET MVC Views to String](#)

## 三、ASP.NET MVC中通过重写操作和结果过滤器实现静态化

我们可以通过重写过滤器来实现一个自定义的静态操作过滤器。通过继承IResultFilter接口，利用HttpResponse.Filter将页面内容输出到流中，最后通过IO操作生成静态页面。

另外利用Response.Filter属性本身也是一种生成静态页的方法，网上也是有很多相关资料。

## 四、直接在后台编写和生成HTML页面的代码

本人十分不推荐使用这种方法，这种方法是直接在服务端用字符串拼接html代码，最后保存为.html后缀的静态网页。通过拼接字符串，对于维护、优化、代码查看来说，都是非常不友好的，有可能会坑了以后维护代码的兄弟。毕竟一大堆的StringBuilder.Append方法是和很不雅观的，而且也十分不灵活，需要使用大量的转义符号！

## 五、使用XML+XSL生成HTML

这里我建议看这篇文章：[使用XML及XSL生成简单HTML](#)

这种方式主要是将数据保存成XML，通过与XSL模板进行转换后（相当于数据填充到模板中），形成纯HTML页面字符串，最后进行IO操作保存并生成静态页面，这也算是另外一种方式的静态化。

另外听说可以使用自定义基架模板，比如MVC中默认的T4模板（Text Template Transformation Toolkit），具体我也不太了解，有兴趣的朋友可以看下。

这里额外的说下生成静态页面的时机，一般是根据业务需求来决定，有条件也可以编写一个服务程序，在服务器上自动定时生成纯静态的HTML页面。当然也可以在添加、删除、修改页面等操作后再进行静态化处理。

作者：[十有三](#) 出处：<http://shiyousan.com/post/635686634025340822> 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。

