**ASP.NET的基本概念**

**ASP.NET的一些基本概念**

1. **ASP.NET的基本简介：**

-> 是一个使用 HTML、CSS、JavaScript 和服务器脚本创建网页和网站的开发框架，它支持三种不同的开发模式，其中，MVC和Web Forms都是属于 **WEB应用程序** 的范畴

**Web Pages（Web 页面）、MVC（Model View Controller 模型-视图-控制器）、Web Forms（Web 窗体）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Web Pages 单页面模式** |  | **MVC 模型-视图-控制器** |  | **Web Forms 事件驱动模式** |
| 最简单的 ASP.NET 模式，提供了一种简单便捷的方式把c#代码和HTML、CSS、JavaScript、服务器代码相结合。 |  | MVC 将 Web 应用程序分成 3 个不同的组成部分：   1. 模型负责数据 2. 视图负责显示 3. 控制器负责输入 |  | 传统的 ASP.NET 事件驱动开发模式：  带有服务器控件、事件和服务器代码的网页，类似于WinForm。 |

1. **ASP.NET应用于WEB开发：**

-> 什么是WEB开发？ 把网站部署到服务器的IIS上，其他电脑的浏览器通过访问该网站的地址而向服务器发送请求，而服务器上的IIS根据浏览器发送的请求地址，并通过我们先前在IIS上设置的物理路径，找到网站文件，并投之相应计算，最后把内容返回给发送请求的浏览器(也就是浏览器上呈现出来的页面信息)，这就是WEB开发。

1. **动态网页和静态网页的区别**

**-> 动态网页：**

1. 动态网页在服务器上所呈现的是一种处理程序（\*.asp、\*.aspx、\*.ashx）
2. 区别于静态网页，动态网页是不能直接通过本地路径直接打开的，因为动态网页文件使用的是c#代码进行编写（浏览器并不能够解析c#代码），动态网页所呈现的内容都是需要通过里面的c#代码计算得出结果并动态生成的HTML网页
3. 能够与数据库进行交互
4. 因为网页内容都是动态生成的原因，网页内容会随着c#代码的运行去更新、呈现出不同的内容
5. 当用户通过浏览器发送请求至服务器的时候，服务器会运行该代码文件，并解析该文件里面的c#代码，然后进行一些内部的相关处理，最后会把处理好的结果动态拼凑展示的内容返回给浏览器，浏览器再接收服务器发送的响应内容进行渲染解析

**-> 静态网页：**

1. 静态网页在服务器上呈现的文件就是一种网页文件(\*.htm或者是\*.html)
2. 区别于动态网页，静态网页是可以通过本地文件路径直接打开，因为他是已经编写好的、写死的HTML文件，浏览器可以直接对该文件进行渲染解析并呈现给用户
3. 是不能够与数据库进行交互的
4. 因为已经写死的原因，网页内容是不能随时更改的，如果想要更改该网页内容，需要重新进行编写并上传覆盖至服务器
5. 当用户通过浏览器发送请求至服务器的时候，区别于动态网页，中间是没有代码的计算与编译这一过程的，它是直接把已经写好的HTML文件的内容直接返回给浏览器，最后浏览器再接收服务器发送的响应内容进行渲染解析
6. **WEB应用程序项目和WEB网站项目的区别**

**-> WEB网站项目：[右键解决方案] -> [新建网站] -> [ASP.NET空网站] -> [新建一个WEB网站项目]**

WEB网站项目下的每一个处理程序文件都是独立的，当我们在WEB网站项目下新建不同的处理程序文件的时候，它们都是没有命名空间的，因此，当我们请求、浏览一个已经写好的处理程序文件的时候，VS只会编译我们当前所请求、浏览的处理程序文件，并且VS会把编译出来的DLL放在一个我们系统的临时文件夹下，详情可以通过以下代码来验证：Assembly.GetExecutingAssembly().Location），通过此点也可以引出以下两点不同

1. 当我们在一个WEB网站项目下写了一个类，如果我们想在当前请求的处理程序中访问这个类是无法访问到的，因为VS只会编译我们当前所请求的处理程序文件，压根就没编译出来我们刚刚写好的那个类，但是我们把我们所写好的类放在项目文件夹的 [App\_Code] 文件夹目录下，那么当我们在请求一个处理程序的时候，VS也会自动编译该目录下的代码
2. 当我们在一个WEB网站项目下新建了不同的处理程序，当其中一个处理程序代码上出错了，那么我们去请求另一个处理程序也是可以正常执行的，因为VS只会编译我们当前所请求的处理程序文件

**-> WEB应用程序项目：[右键解决方案] -> [新建项目] -> [ASP.NET WEB应用程序] -> [新建一个WEB应用程序项目]**

一个完整的WEB应用程序项目才属于一个程序，当我们在WEB应用程序项目下新建不同的处理程序文件的时候，它们都是拥有属于自己项目的命名空间的，因此，当我们请求、浏览一个已经写好的处理程序文件的时候，VS会统一编译当前WEB应用程序项目下的所有文件，并把生成出来的DLL放在当前项目文件夹目录的bin文件夹下，通过此点也可以引出以下两点不同

1. 区别于WEB网站项目，当我们在一个WEB应用程序项目下写了一个类，如果我们想在当前所请求的处理程序中访问这个类是可以正确访问到的，因为VS在编译生成的时候会把当前WEB应用成项目下的所有文件都统一编译进去
2. 当我们在一个WEB应用程序项目下存在不同的处理程序，其中一个处理程序在代码上出现了错误，那么其他处理程序文件也无法正确的请求、浏览，因为要VS编译的时候会统一进行编译，所以不能存在一个文件出错

**IIS：Internet Information Services**

1. **IIS的简介**

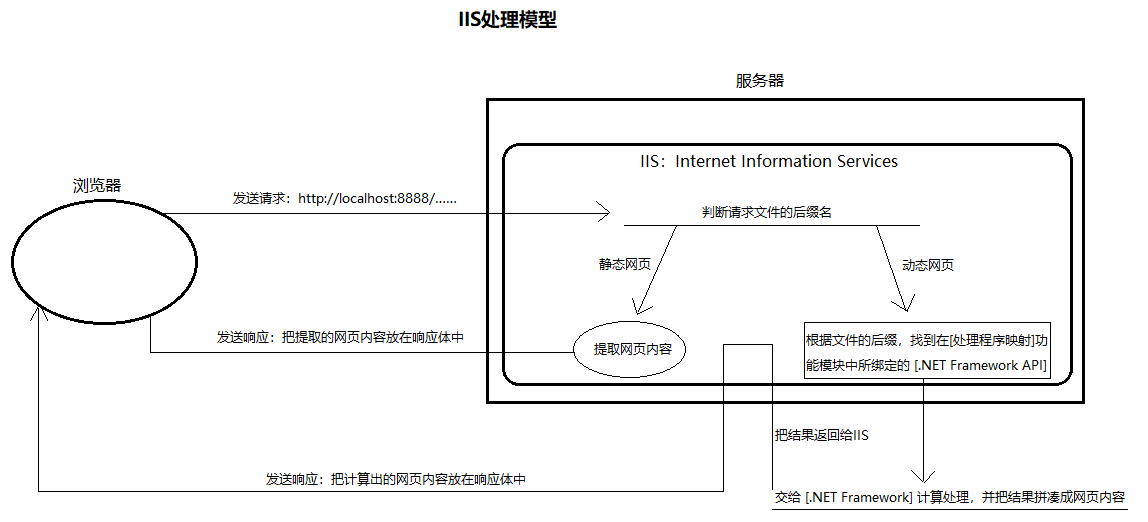
-> IIS：Internet Information Services 互联网信息服务

-> IIS是微软提供的一种服务器部署配置软件，**ASP.NET开发与它的关系密切相连，通过IIS我们才能够使用ASP.NET技术快速的完成网站的部署与服务器的相关配置**

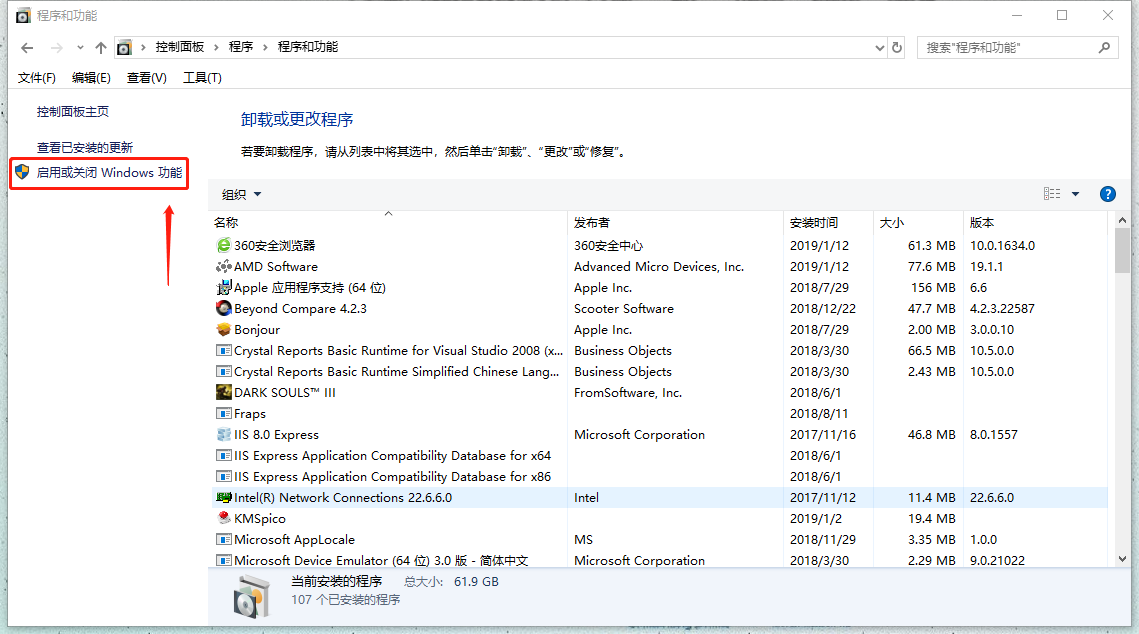
-> 在开发测试阶段，我们是不需要在自己电脑上部署IIS的，因为我们在使用VS进行调试的时候，VS会自动帮我启动一个 [IIS Express] 的程序，该程序会自动的帮我们把所调试的网站临时部署至IIS上，以便于我们调试开发

1. **IIS处理模型流程：**

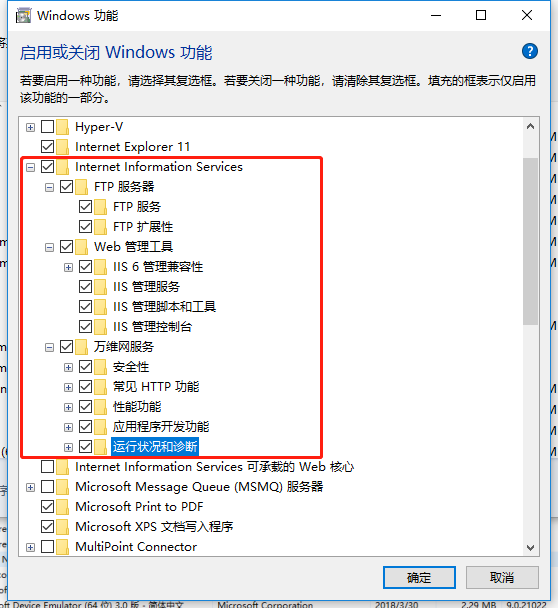
-> 当浏览器发送了一个请求过来时，IIS会找到请求报文中URL所指定的请求文件的物理路径，然后根据该请求文件的后缀名，IIS在内部会做一个判断，如果请求的页面是一个静态网页（\*.htm或者是\*.html），IIS则会读取里面的HTML内容，并把成功读取到的内容放在HTTP响应报文的响应体中，并发送回给客户端（浏览器），而如果请求的是一个动态页面/处理程序（\*.asp/\*.aspx/\*.ashx），IIS就根据指定处理文件的后缀名，在 [处理程序映射功能模块] 中找到对应注册、绑定的 [.Net Framework API]，然后交给 .Net FrameWork 去执行该处理程序文件中的代码**(ashx一般处理程序会执行里头的[ProcessRequest] 函数)**，最后 .Net Framwork 会把处理程序所处理的结果反交给IIS，IIS再把刚刚接收到的结果、内容放在HTTP响应报文中的响应体中，并发送回给客户端（浏览器）

****

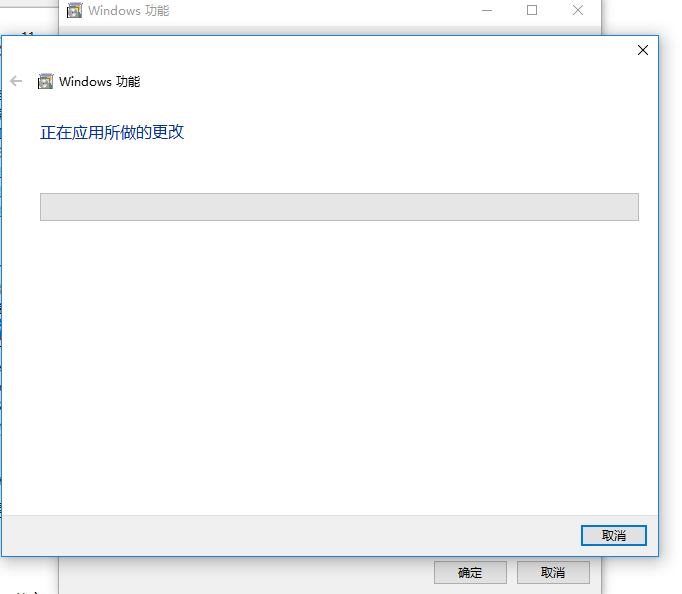
1. **如何安装IIS：**
2. 在Windows控制面板中找到添加与删除程序，并点击 [启用或关闭Windows功能]



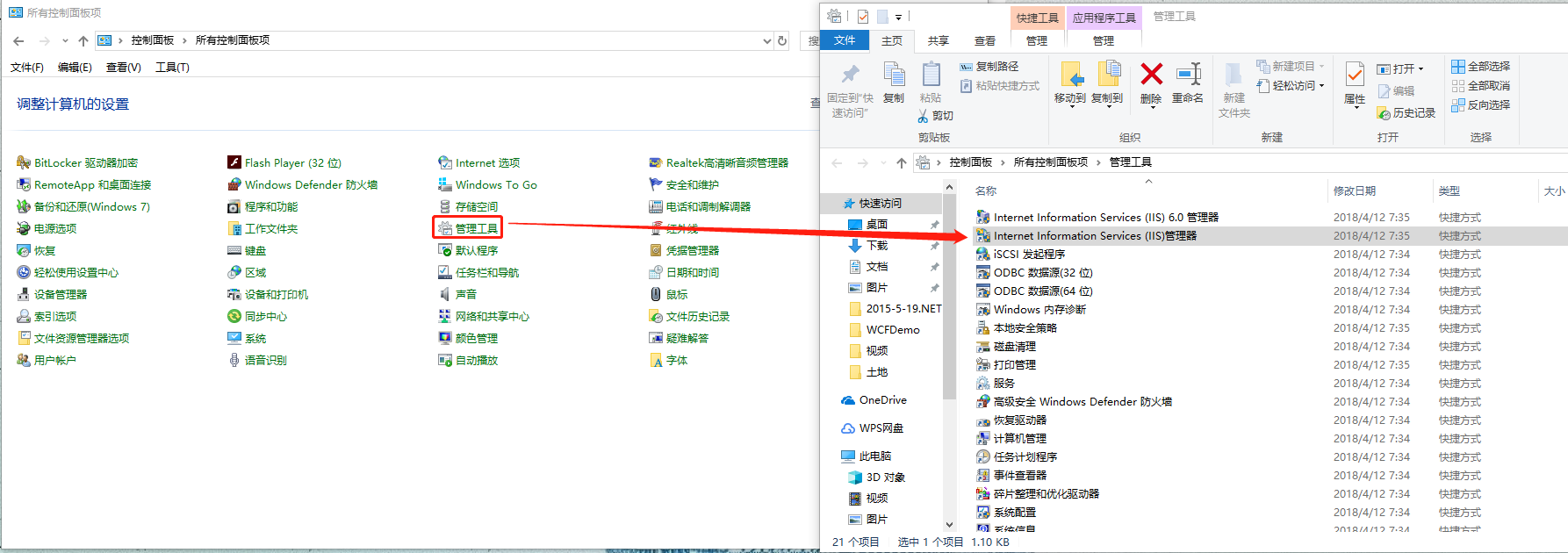
1. 找到 [Internet Information Services] 选项，并**选中所有的功能**



1. 当点击确定后，系统就能帮我们自动的安装IIS了



1. 我们通过在Windows控制面板中的管理工具能够准确地找到我们刚刚安装好的IIS



1. IIS安装的注意事项

-> 建议安装顺序为：IIS -> .Net Framework

-> 有时候当一个请求过来的时候，而处理程序无法正确执行的原因就是因为这个导致的，因为IIS其实就是一个服务器软件，它是不认识我们所写的c#代码的，需要IIS根据指定处理文件的后缀名，在 [处理程序映射功能模块] 中找到对应注册、绑定的 [.Net Framework API]，然后通过 .Net FrameWork 才能够执行处理程序中的c#代码。而有部分的系统当我们的安装顺序为 .Net Framework -> IIS ，IIS中的 [处理程序映射功能] 里面就没有对应处理程序文件后缀所注册的 [.Net Framework API]，这时候处理程序就无法成功的执行了

-> 如果还是发生了以上的这种情况，可以参考以下网站来解决：<https://www.jb51.net/article/84668.htm>，<https://blog.csdn.net/MadeInJack/article/details/2275468>

1. **如何使用IIS部署网站**
2. 应用程序池 微信截图_20190114230347：

-> 应用程序池是将一个或多个应用程序链接到一个或多个工作进程集合的配置

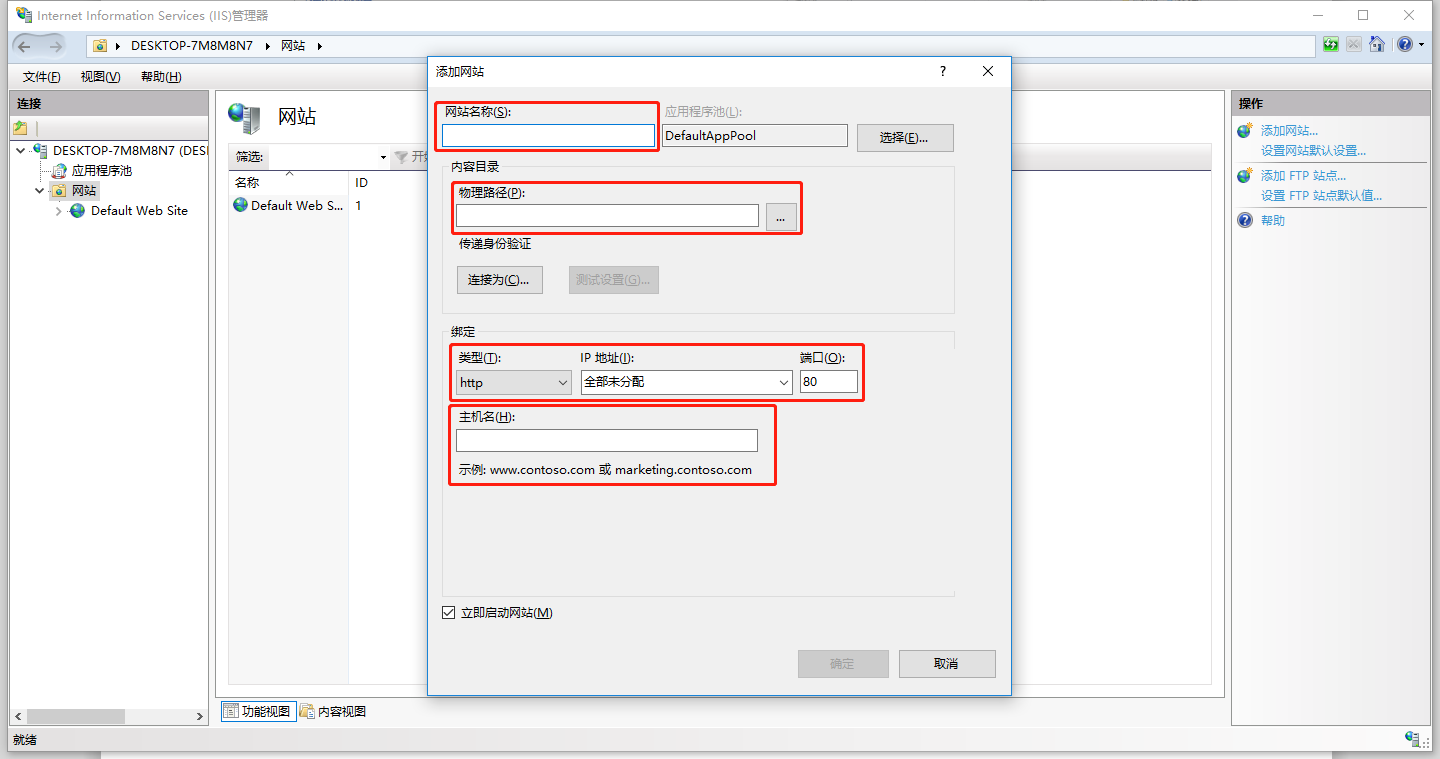
-> 每个应用程序池都对应着一个工作者进程：IIS 6.0 : w3wp.exe / IIS 5.0 : aspnet\_wp.exe

-> 应用程序池的概念出现在 IIS 6.0 当中，而IIS 5.0中所使用的还是应用程序域(Application Domain)的处理机制

-> 一个资源文件请求至服务器时，HTTP.SYS会根据IIS中的Metabase查看基于该Request的Resource属于哪个Application Pool，如果不存在则创建之，当应用程序池创建的时候，由WAS(Web Administrative service)根据IIS在Metabase中所维护的Application Pool和Worker Process的Mapping，找到所对应的Worker Process并监听它，如果Woker Process不存在，则创建之，在工作者进程在进行初始化的过程当会通过相对应的(分析资源文件的后缀然后在 [处理程序映射模块(ISAPI Extension Mapping)] 中获知当前所请求资源所指定的ISAPI Extension，并告知给工作者进程)ISAPI Extension(以 \*.aspx 请求资源为例，所指定的ISAPI Extension默认为 [aspnet\_isapi.dll])去初始化完成CLR，为Web Application构建一个托管运行的环境，需要注意的是，CLR在初始化的过程当中会为当前Web Application构建一个应用程序域，简而言之，应用程序池承载着一个工作者进程，而工作者进程当中承载着一个或多个应用程序域，应用程序池通过进程之间的边界分隔机制来达成不同的工作者进程之间互不共享信息的局面，而应用程序域在一个工作者进程当中也各自有用自己的实例，透过这点也能够达成一个互不影响的机制。由此可以得出结论，如果一个工作者进程挂了，那么在这个工作者进程下的所有Web Application都会随之受到影响，但是当前系统中的其他工作者进程却不会受到崩溃的工作者进程的影响



1. 网站微信截图_20190114230403：
2. [右键网站] -> [添加网站] 弹出 [添加网站] 的配置窗口，我们就可以正式开始在IIS上部署一个网站



1. 以下解释上图所框住的名词

-> 网站名称：设置我们所部署在IIS上的网站的名称

-> 物理路径：网站文件所指定的文件夹，便于IIS在我们所指定的物理路径中寻找相关文件，关于物理路径我们最好 不要直接指定开发文件的目录，正确的做法是在VS中右键我们所需要发布的Web Application，然后点击发布到指 定目录下，然后我们在让物理路径指向我们所指定的发布目录下即可，为什么要这么做？如果我们直接在IIS中指定网 站的物理路径为我们所开发文件的目录下的话，我们会发现一些Web Application中的 \*.cs 文件是直接存在于文件 目录下的，也就是说别人可以直接通过 \*.cs 文件来查看我们网站的源代码，而在通过在VS中点击发布Web Application后，相应的cs 文件都会被编译成为了dll文件并且这些dll文件VS都是通过【Release】模式进行编译 的，性能会比debug模式高很多

-> 绑定类型（协议）：浏览器和服务器之间传递数据的一种要求、规则、规定

**拓展：除了http协议应用于网站之外，还有众多的协议供我们选择，其中smtp协议是属于发送邮件的协议**

-> IP地址：服务器的IP地址，类似于一个生活中的住址(不包含门牌号)，我们可以通过IP地址去找到服务器

**拓展：192.168.xx.xx 网段的IP地址只能够在局域网中访问到，如果想要互联网能够识别的IP地址，需要我们去购买**

-> 端口号：类似于一个门牌号，我们只有通过地址（IP地址）和门牌号（端口号），才能正确的访问到我们 部署 在IIS上的网站

-> 主机名：网站的域名，域名是唯一的，一个域名只能够绑定一个IP地址

**拓展：当我们在浏览器中输入网站的域名，例如：[www.baidu.com，当我们进入该域名的网站的时候，浏览器最终还是会把](http://www.baidu.com，当我们进入该域名的网站的时候，浏览器最终还是会把) 该域 名解析成一个网络能够访问的IP地址**

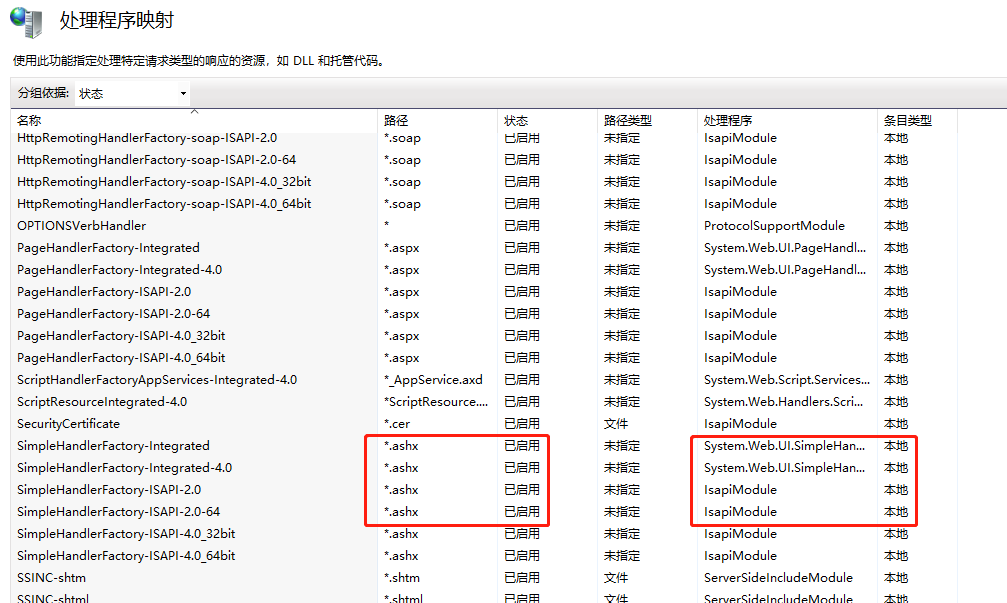
1. 通过以上操作配置后，我们就成功的把网站部署到IIS上了，这时候我们只需要在浏览器中按照以下格式输入就 能够正常的访问网站了：协议类型://地址(本地主机的话可以使用localhost或127.0.0.1) :端口号/网站文件名……
2. **IIS一些常用功能介绍**
3. 默认文档 微信截图_20190228203730：用于配置所部署网站的首页



1. 处理程序映射 ：配置请求资源文件与ISAPI Extension(ISAPI Extension Mapping)、HttpHandler之间的映射关系

-> 当一个资源请求至服务器的时候，工作者进程会分析当前请求资源文件的后缀并根据在该模块中所对应注册的ISAPI Extension Mapping找到ISAPI Extension（以\*.aspx 请求资源为例，所指定的ISAPI Extension默认为 [aspnet\_isapi.dll])，并通过这个 ISAPI Extension 去加载CLR，构建 .NETFramework Runtime

-> 除了上面所说的关于ISAPI Extention的Mapping，一个Resource Request至服务器的HttpRuntime的时候，同样会根据这里的Mapping去构建相应的HttpHandler去处理后续的请求



**HttpHandler**

1. **一般处理程序：\*.ashx**

-> 简介

1. ASP.NET中，以.ashx文件后缀作为结尾的一般处理程序是一种简单、高效的Web Handler，它能够让我们更加方便、 有效的对HTTP请求的数据进行解析处理或返回，例如生成动态图片、生成JSON/XML格式的数据并返回........等
2. 一般处理程序所展现的性能是优于WebForm(较为消耗性能)和MVC的(介于两者之间)，因为一般处理程序它省去了 控件和事件的解析以及页面处理的过程
3. 一般处理程序文件(.ashx)是[.net 2.0]新加的文件类型，其实在[.net 1.0下]已经可用，但是当时微软并没有对其进行公 开提供
4. **Web窗体：\*.aspx**

-> 关于WebForm的概述会在另一边文档体现，[请点击这里](../WebForm/ASP.NET WebForm.docx)