逆心

船停在港湾是很安全的,但那不是造船的目的!

博客园:: 首页:: 新随笔:: 联系:: 订阅 XML :: 管理 492 Posts:: 0 Stories:: 450 Comments:: 0 Trackbacks

WebConfig配置文件详解

```
今天看到博客园一位朋友整理的一个WebConfig配置文件详解,觉得不错,转载一下:
<?xml version="1.0"?>
<!--注意:除了手动编辑此文件以外,您还可以使用 Web 管理工具来配置应用程序的设置。可以使用 Visual
Studio 中的"网站"->"Asp.Net 配置"选项。
设置和注释的完整列表在 machine.config.comments 中,该文件通常位于
"Windows"Microsoft.Net"Framework"v2.x"Config 中。-->
<!--Webconfig文件是一个xml文件, configuration是xml文件的根节点,由于xml文件的根节点只能有一
个,所以Webconfig的所有配置都是在这个节点内进行的。-->
<configuration>
 <!--指定配置节和命名空间声明。clear:移除对继承的节和节组的所有引用,只允许由当前 section 和
sectionGroup 元素添加的节和节组。remove:移除对继承的节和节组的引用。
section:定义配置节处理程序与配置元素之间的关联。sectionGroup:定义配置节处理程序与配置节之间的关
联。-->
<configSections>
   <sectionGroup name="system.web.extensions"</pre>
type="System.Web.Configuration.SystemWebExtensionsSectionGroup,
System. Web. Extensions, Version=1.0.61025.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35">
     <sectionGroup name="scripting"</pre>
type="System.Web.Configuration.ScriptingSectionGroup, System.Web.Extensions,
Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35">
       <section name="scriptResourceHandler"</pre>
type="System.Web.Configuration.ScriptingScriptResourceHandlerSection,
System. Web. Extensions, Version=1.0.61025.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" requirePermission="false"
allowDefinition="MachineToApplication"/>
     </sectionGroup>
   </sectionGroup>
```

type="Intelligencia.UrlRewriter.Configuration.RewriterConfigurationSectionHandler,

Intelligencia.UrlRewriter" />

<section name="rewriter"</pre>

```
</configSections>
<!--appSettings是应用程序设置,可以定义应用程序的全局常量设置等信息-->
    <appSettings>
<add key="1" value="1" />
<add key="gao" value="weipeng" />
</appSettings>
<!--连接字符串设置-->
<connectionStrings>
   <add name="ConnString" connectionString="Data Source=GAO; Initial</pre>
Catalog=HBWXDate; User ID=sa; password=sa"></add>
   <add name="111" connectionString="11111" />
</connectionStrings>
<!--指定应用子配置设置的资源,并锁定配置设置,以防止它们被子配置文件重写。page指定应用包含的配置
设置的资源.allowOverride是否允许配置文件的重写,提高配置文件的安全性-->
<location path="Default.aspx" allowOverride="false">
   <!--控制asp.net运行时的行为-->
<system.web>
   <!--identity控制web应用程序的身份验证标识.-->
   <identity impersonate="false" />
   <!--标识特定于页的配置设置(如是否启用会话状态、视图状态,是否检测用户的输入等)。<pages>可以
在计算机、站点、应用程序和子目录级别声明.
   这里的几个属性的意思是默认主页为Index,主题是Default,不检测用户在浏览器输入的内容中是否存在潜
在的危险数据(注:该项默认是检测,如果你使用了不检测,一要对用户的输入进行编码或验证),在从客户端回
发页时将检查加密的视图状态,以验证视图状态是否已在客户端被篡改。(注:该项默认是不验证)禁用
ViewState-->
```

<pages masterPageFile="Index" theme="Default" buffer="true"</pre>

```
enableViewStateMac="true" validateRequest="false" enableViewState="false">
     <!--controls 元素定义标记前缀所在的 register 指令和命名空间的集合-->
     <controls></controls>
     <!--将在程序集预编译期间使用的导入指令的集合-->
     <namespaces></namespaces>
   </pages>
   <!--默认错误页设置,mode:具有On,Off,RemoteOnly 3种状态。On表示始终显示自定义的信息; Off表
示始终显示详细的asp.net错误信息; RemoteOnly表示只对不在本地Web服务器上运行的用户显示自定义信
息.defaultRedirect:用于出现错误时重定向的URL地址-->
   <customErrors defaultRedirect="Err.html" mode="RemoteOnly">
     <!--特殊代码编号的错误从定向文件-->
     <error statusCode="403" redirect="NoAccess.htm" />
     <error statusCode="404" redirect="FileNotFound.htm" />
   </customErrors>
   <!--配置调试和跟踪:下面配置的意思是启动调试(默认),捕获跟踪信息,要缓存的跟踪请求个数(15),跟踪
结果的排列顺序-->
   <trace enabled="true" localOnly="false" pageOutput="true" requestLimit="15"</pre>
traceMode="SortByCategory"/>
   <!-- 设置 compilation debug="true" 将调试符号插入已编译的页面中。但由于这会影响性能,因此
只在开发过程中将此值设置为 true。设置默认的开发语言C#。batch是否支持批处理-->
   <compilation debug="true" defaultLanguage="c#" batch="false">
     <assemblies>
      <!--加的程序集引用,每添加一个程序集,就表示你的应用程序已经依赖了一个程序集,你就可以在你
的应用程序中使用了-->
      <add assembly="System.Web.Extensions, Version=1.0.61025.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35"/>
      <add assembly="System.Design, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A"/>
```

```
<add assembly="System.Web.Extensions.Design, Version=1.0.61025.0,</pre>
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
        <add assembly="System.Windows.Forms, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Web, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A"/>
        <add assembly="System, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Xml, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Drawing, Version=2.0.0.0, Culture=neutral,</pre>
PublicKeyToken=B03F5F7F11D50A3A"/>
        <add assembly="System.Data, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
        <add assembly="System.Web.Services, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=B03F5F7F11D50A3A"/>
        <add assembly="System.Configuration, Version=2.0.0.0, Culture=neutral,
PublicKevToken=B03F5F7F11D50A3A"/>
      </assemblies>
      <!--定义用于编译自定义资源文件的生成提供程序的集合。-->
      <buildProviders>
        <!--->
        <add extension=".aspx" type="System.Web.Compilation.PageBuildProvider"/>
        <add extension=".ascx"</pre>
type="System.Web.Compilation.UserControlBuildProvider"/>
        <add extension=".master"</pre>
type="System.Web.Compilation.MasterPageBuildProvider"/>
        <add extension=".asmx"</pre>
type="System.Web.Compilation.WebServiceBuildProvider"/>
        <add extension=".ashx"</pre>
type="System.Web.Compilation.WebHandlerBuildProvider"/>
        <add extension=".soap"
type="System.Web.Compilation.WebServiceBuildProvider"/>
        <add extension=".resx" type="System.Web.Compilation.ResXBuildProvider"/>
```

```
<add extension=".resources"</pre>
type="System.Web.Compilation.ResourcesBuildProvider"/>
       <add extension=".wsdl" type="System.Web.Compilation.WsdlBuildProvider"/>
       <add extension=".xsd" type="System.Web.Compilation.XsdBuildProvider"/>
       <add extension=".rdlc" type="Microsoft.Reporting.RdlBuildProvider,
Microsoft.ReportViewer.Common, Version=8.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"/>
     </buildProviders>
   </compilation>
     <!--通过 <authentication> 节可以配置 ASP.NET 使用的 安全身份验证模式,以标识传入的用
户。Windows: 使用IIS验证方式, Forms: 使用基于窗体的验证方式, Passport: 采用Passport cookie验证
模式, None: 不采用任何验证方式-->
   <authentication mode="Forms">
     <!--Name: 指定完成身份验证的Http cookie的名称.LoginUrl: 如果未通过验证或超时后重定向的
页面URL,一般为登录页面,让用户重新登录。Protection:指定 cookie数据的保护方式.
     可设置为:All表示加密数据,并进行有效性验证两种方式,None表示不保护Cookie,Encryption表示
对Cookie内容进行加密, validation表示对Cookie内容进行有效性验证, TimeOut: 指定Cookie的失效时
间. 超时后要重新登录。-->
     <forms name=".ASPXUSERDEMO" loginUrl="Login.aspx" protection="All"</pre>
timeout="30"/>
   </authentication>
   <!--控制对 URL 资源的客户端访问(如允许匿名用户访问)。此元素可以在任何级别(计算机、站点、应
用程序、子目录或页)上声明。必需与<authentication> 节配合使用。此处的意思是对匿名用户不进行身份验
证。拒绝用户weipeng-->
   <authorization>
     <allow users="*"/>
     <deny users="weipeng"/>
     <allow users="aa" roles="aa" />
   </authorization>
   <!--站点全球化设置,requestEncoding: 它用来检查每一个发来请求的编码.responseEncoding: 用
于检查发回的响应内容编码.fileEncoding:用于检查aspx,asax等文件解析的默认编码,默认的编码是utf-8--
   <globalization requestEncoding="gb2312" responseEncoding="gb2312"</pre>
```

```
fileEncoding="gb2312" />
   <!--会话状态设置。mode: 分为off,Inproc,StateServer,SqlServer几种状态 mode = InProc 存
储在进程中特点: 具有最佳的性能,速度最快,但不能跨多台服务器存储共享.mode = "StateServer" 存储在
状态服务器中特点: 当需要跨服务器维护用户会话信息时,使用此方法。但是信息存储在状态服务器上,一旦状态
服务器出现故障,信息将丢失. mode="SqlServer" 存储在sql server中特点:工作负载会变大,但信息不会
丢失
   stateConnectionString:指定asp.net应用程序存储远程会话状态的服务器名,默认为本机。
sqlConnectionString: 当用会话状态数据库时,在这里设置连接字符串。Cookieless:设置为flase时,表示
使用cookie会话状态来标识客户.timeout表示会话超时时间。-->
   <sessionState mode="InProc" stateConnectionString="tcpip=127.0.0.1:42424"</pre>
sqlConnectionString="data source=127.0.0.1;Trusted Connection=yes" cookieless="false"
timeout="20"></sessionState>
   <!--为 ASP.NET 应用程序配置页的视图状态设置。设置要存储在页历史记录中的项数。-->
   <sessionPageState historySize="9"/>
   <!--配置asp.net http运行库的设置。可以在计算机,站点,应用程序和子目录级别声明
   允许最多的请求个数100,最长允许执行请求时间为80秒,控制用户上传文件的大小,默认是4M。
useFullyQualifiedRedirectUrl客户端重定向不需要被自动转换为完全限定格式。-->
   <httpRuntime appRequestQueueLimit="100" executionTimeout="80"</pre>
maxRequestLength="40960" useFullyQualifiedRedirectUrl="false"/>
   <!--httpModules在一个应用程序内配置 HTTP 模块。-->
   <httpModules>
     <add name="OutputCache" type="System.Web.Caching.OutputCacheModule" />
     <add name="Session" type="System.Web.SessionState.SessionStateModule" />
     <add name="WindowsAuthentication"
type="System.Web.Security.WindowsAuthenticationModule" />
     <add name="FormsAuthentication"</pre>
type="System.Web.Security.FormsAuthenticationModule" />
     <add name="PassportAuthentication"
type="System.Web.Security.PassportAuthenticationModule" />
     <add name="RoleManager" type="System.Web.Security.RoleManagerModule" />
     <add name="UrlAuthorization" type="System.Web.Security.UrlAuthorizationModule"</pre>
/>
     <add name="FileAuthorization"
type="System.Web.Security.FileAuthorizationModule" />
     <add name="AnonymousIdentification"
```

```
type="System.Web.Security.AnonymousIdentificationModule" />
     <!--自定义的URL重写,type基本上就是dll名-->
     <add name="UrlRewriter" type="Intelligencia.UrlRewriter.RewriterHttpModule,
Intelligencia.UrlRewriter" />
     <add name="Profile" type="System.Web.Profile.ProfileModule" />
   </httpModules>
   <!--http:// Handlers用于根据用户请求的URL和HTTP谓词将用户的请求交给相应的处理程序。可以在配置级
别的任何层次配置此节点,也就是说可以针对某个特定目录下指定的特殊文件进行特殊处理。
   add:指定映射到处理程序的谓词/路径。clear:移除当前已配置或已继承的所有处理程序映射。remove:移
除映射到处理程序的谓词/路径。remove 指令必须与前一个 add 指令的谓词/路径组合完全匹配。该指令不支
持通配符。-->
   <httpHandlers>
     <remove verb="*" path="*.asmx"/>
     <add verb="*" path="*.asmx" validate="false"</pre>
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions,
Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
     <add verb="*" path="* AppService.axd" validate="false"</pre>
type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions,
Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"/>
     <add verb="GET,HEAD" path="ScriptResource.axd"</pre>
type="System.Web.Handlers.ScriptResourceHandler, System.Web.Extensions,
Version=1.0.61025.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
validate="false"/>
     <add verb="POST,GET" path="ajaxpro/*.ashx" type="AjaxPro.AjaxHandlerFactory,
AjaxPro.2"/>
   </httpHandlers>
   <!--为 Web 应用程序使用的 Cookie 配置属性。domain:设置 Cookie 域名。httpOnlyCookies:在
Internet Explorer 6 SP1 中启用 HttpOnlyCookies Cookie 的输出。默认值为 false。requireSSL:
获取一个指示是否需要安全套接字层 (SSL) 通信的值.-->
   <httpCookies httpOnlyCookies="false" requireSSL="false"/>
   <!--控制 ASP.NET Web 服务及其客户端的行为。protocols:指定传输协议,ASP.NET 可使用这些传输
协议来解密 HTTP-->
   <webServices>
     otocols>
       < add/>
```

```
</protocols>
   </webServices>
   <!--为 Web 应用程序配置缓存设置。cache:定义全局应用程序缓存设置。outputCache:指定应用程序
范围的输出缓存设置。outputCacheSettings:指定可以应用于应用程序中页的输出缓存设置。
sqlCacheDependency:为 ASP.NET 应用程序配置 SQL 缓存依赖项。-->
   <caching>
     <cache disableMemoryCollection = "false" disableExpiration = "false"</pre>
privateBytesLimit = "0" percentagePhysicalMemoryUsedLimit = "90" privateBytesPollTime
= "00:02:00"/>
     <!--设计需要以这种方式缓存的页时,您需要向该页添加以下指令: <%@ OutputCache
CacheProfile="ServerOnly" %>-->
     <outputCacheSettings>
       <outputCacheProfiles>
         <add name="ServerOnly" duration="60" varyByCustom="browser"</pre>
location="Server" />
       </outputCacheProfiles>
     </outputCacheSettings>
   </caching>
    </system.web>
</location>
<!--网络设置, authenticationModules: 指定用于对 Internet 请求进行身份验证的模块。
connectionManagement: 指定与 Internet 宿主的连接的最大数目。defaultProxy: 配置超文本传输协议
(HTTP) 代理服务器。
mailSettings: 配置简单邮件传输协议 (SMTP) 邮件发送选项。requestCaching: 控制网络请求的缓存机
制。settings: 配置 System.Net 的基本网络选项。-->
<system.net>
<!--配置SMTP电子邮件设置-->
<mailSettings>
  <smtp from="weipeng">
   <network host="Gao" password="" userName="" />
  </smtp>
```

```
</mailSettings>
  <!--禁用所有缓存-->
  <requestCaching disableAllCaching="true"></requestCaching>
  <!--指定代理地址,并对本地访问和 contoso.com 跳过代理。-->
  <defaultProxy>
    bypassonlocal="True"/>
    <br/>bypasslist>
      <add address="[a-z]+".contoso".com" />
    </bypasslist>
  </defaultProxy>
</system.net>
<!--该节替换在 httpHandlers 和 httpModules 节中添加的与 AJAX 相关的 HTTP 处理程序和模块。该
节使 IIS 7.0 在集成模式下运行时可使用这些处理程序和模块。在iis7.0 下运行 ASP.NET AJAX 需要
system.webServer
节。对早期版本的 IIS 来说则不需要此节。 -->
<system.webServer>
   <validation validateIntegratedModeConfiguration="false"/>
   <modules>
     <add name="ScriptModule" preCondition="integratedMode"</pre>
type="System.Web.Handlers.ScriptModule, System.Web.Extensions, Version=3.5.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>
   </modules>
   <handlers>
     <remove name="WebServiceHandlerFactory-Integrated"/>
     <add name="ScriptHandlerFactory" verb="*" path="*.asmx"</pre>
preCondition="integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory,
System. Web. Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
     <add name="ScriptHandlerFactoryAppServices" verb="*" path="* AppService.axd"</pre>
preCondition="integratedMode" type="System.Web.Script.Services.ScriptHandlerFactory,
System. Web. Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
```

```
<add name="ScriptResource" preCondition="integratedMode" verb="GET,HEAD"</pre>
path="ScriptResource.axd" type="System.Web.Handlers.ScriptResourceHandler,
System.Web.Extensions, Version=3.5.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35"/>
   </handlers>
</system.webServer>
<!--ASP.NET AJAX 中配置 ASP.NET 服务-->
 <system.web.extensions>
   <!--配置 JSON 序列化-->
   <scripting>
     <webServices>
       <jsonSerialization maxJsonLength="5000"/>
     </webServices>
   </scripting>
</system.web.extensions>
<!--对WCF的相关配置-->
<system.serviceModel>
   <services>
     <service name="WCFStudent.WCFStudentText"</pre>
behaviorConfiguration="ServiceBehavior">
       <!-- Service Endpoints -->
       <endpoint address="" binding="wsHttpBinding"</pre>
contract="WCFStudent.IStuServiceContract">
         <!-- 部署时,应删除或替换下列标识元素,以反映在其下运行部署服务的标识。删除之后, WCF 将
自动推导相应标识。-->
         <identity>
           <dns value="localhost"/>
         </identity>
       </endpoint>
```

```
<endpoint address="mex" binding="mexHttpBinding"</pre>
contract="IMetadataExchange"/>
     </service>
   </services>
   <behaviors>
     <serviceBehaviors>
       <behavior name="ServiceBehavior">
         <!-- 为避免泄漏元数据信息,请在部署前将以下值设置为 false 并删除上面的元数据终结点 --
         <serviceMetadata httpGetEnabled="true"/>
         <!-- 要接收故障异常详细信息以进行调试,请将以下值设置为 true。在部署前设置为 false 以
避免泄漏异常信息-->
         <serviceDebug includeExceptionDetailInFaults="false"/>
       </behavior>
     </serviceBehaviors>
   </behaviors>
 </system.serviceModel>
 <!--URL重定向-->
 <rewriter>
   <rewrite url="~/user/u(.+).aspx" to="~/user/index.aspx?r=$1" />
   <rewrite url="~/ask/q(.+).aspx" to="~/home/ask/content.aspx?id=$1" />
   <rewrite url="~/blog/b(.+).aspx" to="~/home/blog/article.aspx?r=$1" />
   <rewrite url="~/news/n(.+).aspx" to="~/home/news/content.aspx?nid=$1" />
   <rewrite url="~/default.aspx" to="~/home/ram/net.aspx" />
</rewriter>
</configuration>
```

文章出处: http://www.cnblogs.com/gaoweipeng/archive/2009/05/17/1458762.html

posted on 2013-04-16 23:49 逆心 阅读(36130) 评论(4) 编辑 收藏

Copyright @ 逆心 Powered by: .Text and ASP.NET Theme by: .NET Monster 小家居