## **UrlRoutingModule 源码**

UrlRoutingModule 位于 System.web.dll 文件中，利用Reflector 可以查看到其源码：

IMG_256

[IMG_257](http://www.cnblogs.com/keyindex/archive/2012/08/11/javascript:void(0);)

1 [TypeForwardedFrom("System.Web.Routing, Version=3.5.0.0, Culture=Neutral, PublicKeyToken=31bf3856ad364e35")]

2 public class UrlRoutingModule : IHttpModule

3 {

4 // Fields

5 private static readonly object \_contextKey = new object();

6 private static readonly object \_requestDataKey = new object();

7 private RouteCollection \_routeCollection;

8

9 // Methods

10 protected virtual void Dispose()

11 {

12 }

13

14 protected virtual void Init(HttpApplication application)

15 {

16 if (application.Context.Items[\_contextKey] == null)

17 {

18 application.Context.Items[\_contextKey] = \_contextKey;

//在HttpApplication 中添加PostResolveRequestCache 事件

19 application.PostResolveRequestCache += new EventHandler(this.OnApplicationPostResolveRequestCache);

20 }

21 }

22

23 private void OnApplicationPostResolveRequestCache(object sender, EventArgs e)

24 {

25 HttpContextBase context = new HttpContextWrapper(((HttpApplication) sender).Context);

26 this.PostResolveRequestCache(context);

27 }

28

29 [Obsolete("This method is obsolete. Override the Init method to use the PostMapRequestHandler event.")]

30 public virtual void PostMapRequestHandler(HttpContextBase context)

31 {

32 }

33

34 public virtual void PostResolveRequestCache(HttpContextBase context)

35 {

//根据HttpContextBase 获得路由数据

36 RouteData routeData = this.RouteCollection.GetRouteData(context);

37 if (routeData != null)

38 {

//获得路由句柄

39 IRouteHandler routeHandler = routeData.RouteHandler;

40 if (routeHandler == null)

41 {

42 throw new InvalidOperationException(string.Format(CultureInfo.CurrentUICulture, SR.GetString("UrlRoutingModule\_NoRouteHandler"), new object[0]));

43 }

44 if (!(routeHandler is StopRoutingHandler))

45 {

//生成请求上下文

46 RequestContext requestContext = new RequestContext(context, routeData);

//填写Request结构

47 context.Request.RequestContext = requestContext;

//根据请求上下文获得httpHandler

48 IHttpHandler httpHandler = routeHandler.GetHttpHandler(requestContext);

49 if (httpHandler == null)

50 {

51 throw new InvalidOperationException(string.Format(CultureInfo.CurrentUICulture, SR.GetString("UrlRoutingModule\_NoHttpHandler"), new object[] { routeHandler.GetType() }));

52 }

//如果httpHandler 是失败的句柄

53 if (httpHandler is UrlAuthFailureHandler)

54 {

55 if (!FormsAuthenticationModule.FormsAuthRequired)

56 {

57 throw new HttpException(0x191, SR.GetString("Assess\_Denied\_Description3"));

58 }

59 UrlAuthorizationModule.ReportUrlAuthorizationFailure(HttpContext.Current, this);

60 }

61 else

62 {

//执行httpHandler的程序

63 context.RemapHandler(httpHandler);

64 }

65 }

66 }

67 }

68

69 void IHttpModule.Dispose()

70 {

71 this.Dispose();

72 }

73

74 void IHttpModule.Init(HttpApplication application)

75 {

76 this.Init(application);

77 }

78

79 // Properties

80 public RouteCollection RouteCollection

81 {

82 get

83 {

84 if (this.\_routeCollection == null)

85 {

86 this.\_routeCollection = RouteTable.Routes;

87 }

88 return this.\_routeCollection;

89 }

90 set

91 {

92 this.\_routeCollection = value;

93 }

94 }

95 }

UrlHttpModule 实现了 IHttpModule 接口

UrlHttpModule 在 init 方法中注册了PostResolveRequestCache 事件的处理程序。

## **PostResolveRequestCache**

    该事件在完成缓存解析并投递时触发。

在 PostResolveRequestCache 之前分别触发的事件有：  
引发 BeginRequest 事件。  
引发 AuthenticateRequest 事件。  
引发 PostAuthenticateRequest 事件。  
引发 AuthorizeRequest 事件。  
引发 PostAuthorizeRequest 事件。  
引发 ResolveRequestCache 事件。

核心逻辑代码：

1 //获取路由信息

2 RouteData routeData = this.RouteCollection.GetRouteData(context);

3 IRouteHandler routeHandler = routeData.RouteHandler;

4 //构建请求上下文

5 RequestContext requestContext = new RequestContext(context, routeData);

6 context.Request.RequestContext = requestContext;

7 IHttpHandler httpHandler = routeHandler.GetHttpHandler(requestContext);

8 //将MvcHandler 实例 映射到管线中（通常我们是利用web.config 进行配置的，但是MvcHandler 没有默认无参构造函数，所以直接通过向其传递一个实例

9 //进行映射）

10 context.RemapHandler(httpHandler);