## **RouteCollection类**

是一个集合类，内部维护着一个RouteBase 以路由名作为Key的字典集合， 所以我们可以给路由命名。主要的属性和方法有：

### **RouteData GetRouteData(HttpContextBase httpContext)**

　　该方法遍历集合内部的RouteBase实体，并返回第一个非Null的RouteData ，具体的RouteData实例，是由所遍历的RouteBase 通过调用 RouteBase.GetRouteData(HttpContext)方法获取的 。

返回的RouteData 中包含一个 IRouteHandler 对象，该接口的只有一个方法，GetHttpHandler，用于获取IHttpHandler 对象。

### **VirtualPathData GetVirtualPath(...)**

　　该方法具有多个重载，当您使用 ASP.NET 路由框架生成 URL 时，GetVirtualPath 方法将返回 VirtualPathData 类的一个实例。 VirtualPathData 类包含与所提供的上下文匹配路由的相关信息。

### **Route MapPageRoute(...)**

　　该方法用于向路由集合中添加路由，提供此方法是为了方便编码， 它等效于调用 Add 方法。其内部实现的主要代码为：

Route item = new Route(routeUrl, defaults, constraints, dataTokens, new PageRouteHandler(physicalFile, checkPhysicalUrlAccess));

注册使用 PageRouteHandler 类创建的 Route 对象。

　　在Asp.net Mvc 方法中，并没有利用该方法向集合中添加路由而是通过了了 MapRoute()方法 这是一个扩展方法，定义在了 RouteCollectionExtensions 类中。

### **static Route MapRoute(this RouteCollection routes, ......)**

　　这是一个扩展方法，并且具有多个重载，目的是方便编码，用于向RouteCollection中 添加路由。其内部主要是 生成了一个 以 MvcRouteHandler 为 路由处理的Route 并将其加入到集合中。

Route route = new Route(url, new MvcRouteHandler()) {

Defaults = new RouteValueDictionary(defaults),

Constraints = new RouteValueDictionary(constraints),

DataTokens = new RouteValueDictionary()

};