# URL 路由

在Web Form 情况下，每一个 ASPX页面既是一个文件，又是一个队请求自包含的响应。而在 MVC 情况下，请求是由控制器类中的动作方法处理的，而且与硬盘上的文件没有一对一的相互关系。

ASP.NET 平台为了处理 MVC 的 URL，采用了路由系统，它主要有两个功能：

* 考查一个输入 URL（Incoming URL），并推出该请求想要的是哪一个控制器和动作。这正是接收到一个客户端请求时，希望路由系统去做的事情。
* 生成一个输出 URL（Outgoing URL），这些 URL 是在视图渲染的 HTML 中出现的 URL，以便使用户点击这些链接时，调用一个特定的动作（此时，它又变成了输入 URL）。

## URL 模式

路由系统用一组路由来实现它的功能。这些路由共同组成了应用程序的URL架构（Schema）或方案（Scheme），这种URL架构（或方案）是应用程序能够识别并能对之作出相应的一组URL。

每一条路由都包含一个URL模式（Pattern），用它与一个输入的URL进行比较，如果该模式与这个URL匹配，那么它（URL模式）便被路由系统用来对这个URL进行处理。

URL模式主要有连个关键行为：

* URL 模式是保守的（Conservative）：只匹配与模式具有相同片段数的URL。
* URL模式是宽松的（Liberal）：如果一个URL正好具有正确的片段数，该模式就会用来为片段变量提取值，而不管这个值可能是什么。

如示例：

|  |  |  |
| --- | --- | --- |
| <http://mysite.com> | /Admin | /Index |
|  | First Segment | Second Segment |
|  | 片段1 | 片段2 |

## 简单路由的创建及注册

定义路由的文件 RouteConfig.cs 文件是在 App\_Start 文件夹中的。在其中定义的静态 RegisterRoutes 是通过 Global.asax.cs 文件进行调用的，当启动应用程序时，它建立了一些核心的 MVC 特性。

基本流程如下：

底层的 ASP.NET 平台在 MVC 第一次启动时调用Global.asax.cs —>Application\_Start() **——>** RouteConfig.cs—> RegisterRoutes()。

定义函数类似如下样子：

public static void RegisterRoutes(RouteCollection routes)

{

routes.IgnoreRoute("{resource}.axd/{\*pathInfo}");

routes.MapRoute(

name: "Default",

url: "{controller}/{action}/{id}",

defaults: new { controller = "Home", action = "Index", id = UrlParameter.Optional }

);

}

Global.asax.cs中的Application\_Start()函数调用方式如下方式：

RouteConfig.RegisterRoutes(RouteTable.Routes);

注册路由的一个方便的方法是，使用 RouteCollection 类所定义的MapRoute方法。如：

routes.MapRoute("MyRoute", "{controller}/{action}");

也可以通过创建一个新的 Route 来实现，如：

Route myRoute = new Route("{controller}/{action}", new MvcRouteHandler());

routes.Add("MyRoute", myRoute);

上述两种方式的效果是一样的。