**路由定义**

将某个url映射到某个处理方法

**传统路由**

传统路由也被叫做全局路由，如下：

app.UseMvc (routes => {

    routes.MapRoute (

        name: "default",

        template: "{controller}/{action=Index}/{id?}");

});

在UseMvc中我们通过routes.MapRoute注册了一个传统路由，当url满足该模板是，会调用我们的控制器方法

**属性路由**

属性路由应用特性进行定义

[Route ("api/[controller]")]

public class SampleDataController : Controller

{

    [HttpGet ("[action]")]

    public IEnumerable<WeatherForecast> WeatherForecasts () { }

}

**路由参数**

我们可以指定url中包含路由参数

    [HttpGet ("[action]/{id}")]

    public IEnumerable<WeatherForecast> WeatherForecasts () { }

可选参数

    [HttpGet ("[action]/{id?}")]

**参数约束**

我们可以约束参数的类型等

    [HttpGet ("[action]/{id:int}")]

    public IEnumerable<WeatherForecast> WeatherForecasts () { }

约束表

| 约束 | 示例 | 匹配项示例 | 说明 |
| --- | --- | --- | --- |
| int | {id:int} | 123456789、-123456789 | 匹配任何整数 |
| bool | {active:bool} | true、FALSE | 匹配 true或 false（区分大小写） |
| datetime | {dob:datetime} | 2016-12-31、2016-12-31 7:32pm | 匹配有效的 DateTime 值（位于固定区域性中 - 查看警告） |
| decimal | {price:decimal} | 49.99、-1,000.01 | 匹配有效的 decimal值（位于固定区域性中 - 查看警告） |
| double | {weight:double} | 1.234、-1,001.01e8 | 匹配有效的 double值（位于固定区域性中 - 查看警告） |
| float | {weight:float} | 1.234、-1,001.01e8 | 匹配有效的 float 值（位于固定区域性中 - 查看警告） |
| guid | {id:guid} | CD2C1638-1638-72D5-1638-DEADBEEF1638、{CD2C1638-1638-72D5-1638-DEADBEEF1638} | 匹配有效的 Guid 值 |
| long | {ticks:long} | 123456789、-123456789 | 匹配有效的 long 值 |
| minlength(value) | {username:minlength(4)} | Rick | 字符串必须至少为 4 个字符 |
| maxlength(value) | {filename:maxlength(8)} | Richard | 字符串不得超过 8 个字符 |
| length(length) | {filename:length(12)} | somefile.txt | 字符串必须正好为 12 个字符 |
| length(min,max) | {filename:length(8,16)} | somefile.txt | 字符串必须至少为 8 个字符，且不得超过 16 个字符 |
| min(value) | {age:min(18)} | 19 | 整数值必须至少为 18 |
| max(value) | {age:max(120)} | 91 | 整数值不得超过 120 |
| range(min,max) | {age:range(18,120)} | 91 | 整数值必须至少为 18，且不得超过 120 |
| alpha | {name:alpha} | Rick | 字符串必须由一个或多个字母字符（a-z，区分大小写）组成 |
| regex(expression) | {ssn:regex(^\\d{{3}}-\\d{{2}}-\\d{{4}}$)} | 123-45-6789 | 字符串必须匹配正则表达式（参见有关定义正则表达式的提示） |
| required | {name:required} | Rick | 用于强制在 URL 生成过程中存在非参数值 |

**自定义路由约束**

以IntRouteConstraint（内置的路由约束）为例

app.UseMvc (routes => {

    routes.MapRoute (

        name: "default",

        // id 可有可无

        template: "{controller}/{action}/{id?}",

        defaults : new { controller = "Home", action = "Index" },

        // 如果url中有id（如：/Home/Index/1）,则 id 必须是Int

        constraints : new { id = new IntRouteConstraint () });

});

自定义路由必须实现IRouteConstraint接口

public class IntRouteConstraint : IRouteConstraint, IParameterPolicy

{

    // 是否满足约束

    public bool Match (

        HttpContext httpContext,

        IRouter route,

        string routeKey,                // 约束的key，也就是 constraints : new { id = new IntRouteConstraint () } 中的 id

        RouteValueDictionary values,    // 解析到的路由的值

        RouteDirection routeDirection)

    {

        ...

        object value;

        if(values.TryGetValue(routeKey, out value) && value != null)

        {

            // 判断 value 是否是 int 类型，是返回 true

        }

        return false;

    }

}