参考文章：https://www.cnblogs.com/artech/p/inside-asp-net-core-1.html

**路由示例**

如下是一个路由示例

app.UseRouter ((Microsoft.AspNetCore.Routing.IRouteBuilder routes) => {

    // 设置默认 RouteHandler

    routes.DefaultHandler = new RouteHandler (handler);

    // 添加路由

    routes.MapRoute (

        name: "default",

        template: "{controller=Home}/{action=Index}/{id?}",

        handler);

});

**UseRouter**

UseRouter向app注册了一个路由中间件RouterMiddleware，并将router传给中间件

public static IApplicationBuilder UseRouter(this IApplicationBuilder builder, IRouter router)

{

    return builder.UseMiddleware<RouterMiddleware>(new object[] { router });

}

**RouterMiddleware如何运行**

如下，路由中间件的运行过程

public class RouterMiddleware

{

    ...

    // 中间件处理流程

    public async Task Invoke (HttpContext httpContext)

    {

        // 生成路由上下文

        var context = new RouteContext (httpContext);

        context.RouteData.Routers.Add (\_router);

        // 使用路由器分析 路由数据（RouteContext.RouteData） 并指定 请求处理器（RouteContext.Handler）

        // 注：\_router 是我们通过 RouteBuilder.Build 生成的路由

        await \_router.RouteAsync (context);

        if (context.Handler == null) {

            \_logger.RequestDidNotMatchRoutes ();

            // 如果没有处理方法，调用后续中间件

            await \_next.Invoke (httpContext);

        } else {

            httpContext.Features[typeof (IRoutingFeature)] = new RoutingFeature () {

                RouteData = context.RouteData,

            };

            // 调用请求处理方法

            await context.Handler (context.HttpContext);

        }

    }

}

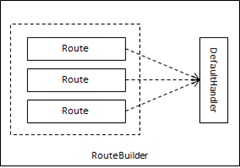
路由中间件将RouteContext传给\_router的RouteAsync

如果\_router设置了RouteContext的Handler（请求委托），那么路由中间件将会调用该请求委托

**router（路由器）从何而来**

router由IRouteBuilder生成，我们可以向IRouteBuilder注册多个Route，IRouteBuilder会将多个Route封装为一个，其原理就是循环调用注册的Route的

DefaultHandler是一个RouteHandler，后面会讲解其用途

[](http://images2015.cnblogs.com/blog/19327/201612/19327-20161222091507667-769892869.png)

**router如何分析路由**

我们的router由Route和RouteHandler组成

Route通过分析RouteContext的HttpContext属性（一般对url进行分析），并设置RouteContext的RouteData数据，如果Route觉得可以匹配当前的请求，那么Route会调用它自身引用的RouteHandler

RouteHandler会设置RouteContext的Handler

**关于UseMvc**

我们 .net core 应用在最后面会使用UseMvc，其实际上是向App注册RouterMiddleware，而UseMvc使用的RouteHandler则是...（什么来的，我忘了，好像是MvcAttributeRouteHandler），其可以根据我们匹配到的路由数据映射到控制器方法