

技术宅毁灭世界

你必须非常努力，才能看起来毫不费力。。。

昵称

萝卜卜

园龄

8年11 月

粉丝

47

关注

88

+ 关注


2024年5月						
日	一	二	三	四	五	六
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

搜索

常用链接

我的标签
2
2

随笔档案
2020年8月 1
2019年7月 1
2018年7月 1
2018年3月 1
2017年8月 1
2016年9月 1
2016年8月 1
2016年6月 1
2016年5月 2



阅读排行榜
简面
实
发网
发面
获其单的内容一 许的

评论排行榜
浏览器的本
简面
实
就一的
发网

推荐排行榜
发网
简面
浏览器的本
就一的
到的 , ,

最新评论
发网
的不果是一 做给网, 文
大

```
45         for (; ; )
46         {
47             var cnx = httpListener.GetContext(); //获取浏览器请求上下文，串行处理，t
48             var req = cnx.Request;
49             var rep = cnx.Response;
50             rep.ContentEncoding = Encoding.UTF8;
51             rep.Headers.Add("Access-Control-Allow-Origin", ""); //允许浏览器跨域
52             rep.StatusCode = 404;
53             var ret = "pag not found";
54             //rep.ContentType = "text"; //返回内容，这里为text，ajax里面的请求datat
55             //这一部分其实就是大部分MVC网络框架里的路由部分！这里简单的发送原始文本给hanc
56             foreach (var kv in handlerMap)
57             {
58                 if(System.Text.RegularExpressions.Regex.IsMatch(req.RawUrl,kv
59                 {
60                     var data = "";
61                     if (req.HttpMethod == "GET")
62                         data = req.RawUrl; //不做任何处理，直接将原始的http请求转发到
63                     else
64                         using (var r = new StreamReader(req.InputStream, Encoding
65                         {
66                             data = r.ReadToEnd();
67                         }
68                         ret = kv.Value.Handler(data);
69                         rep.StatusCode = 200; //ok
70                         break;
71                     }
72                 }
73                 //返回处理结果给浏览器
74                 using (var w = new StreamWriter(rep.OutputStream, Encoding.UTF8))
75                 {
76                     w.WriteLine(ret);
77                 }
78             }
79         }
80
81     }
82 }
```



```
1         void Process()
2         {
3             for(; ; )
4             {
5                 var cnx = httpListener.GetContext();
6                 var req = cnx.Request;
7                 var rep = cnx.Response;
8                 rep.Headers.Add("Access-Control-Allow-Origin", "");
9                 var txt = "";
10                if (req.HttpMethod == "POST")
11                {
12                    var fp = "";
13                    using (var r = new StreamReader(req.InputStream, Encoding.UTF8)
14                    {
15                        fp = r.ReadToEnd();
16                    }
17                    var printer = req.QueryString["printer"];
18                    switch (req.Url.LocalPath)
```

```

19         {
20             case "/printZPP":
21                 printCompl.Print(fp, printer);
22                 break;
23             case "/printTJBB":
24                 printCompl.PrintBB(fp, printer);
25                 break;
26             case "/printQD":
27                 printCompl.PrintQd(fp, printer);
28                 break;
29             case "/printAll":
30                 printCompl.UniversalPrint(fp, printer);
31                 break;
32             default:
33                 //rep.StatusCode = 404;
34                 txt = "404";
35                 break;
36         }
37     }
38     else
39     {
40         switch (req.Url.LocalPath)
41         {
42             case "/GetPrinterNames":
43                 txt = printCompl.GetPrinterNames();
44                 break;
45             case "/GetPrinter":
46                 {
47                     var id = req.QueryString["id"];
48                     txt = printCompl.GetLocalPrinter(id);
49                 }
50                 break;
51             case "/SetPrinter":
52                 {
53                     var id = req.QueryString["id"];
54                     var printer = req.QueryString["printer"];
55                     printCompl.SetLocalPrinter(id, printer);
56                 }
57                 break;
58             default:
59                 txt = printCompl.GetPrinterNames();
60                 //rep.StatusCode = 404;
61                 txt = "404";
62                 break;
63         }
64     }
65
66     using (var w = new StreamWriter(rep.OutputStream, Encoding.UTF8))
67     {
68         w.WriteLine(txt);
69     }
70 }
71 }

```

```

1 //类似于MVC框架的ACTION或者CONTROLLER，可派生此接口实现其他操作
2 public interface IHttpHandler
3 {
4     string Handler(string txt); //简单的处理文本消息
5 }

```

IPrint打印接口，派生此接口执行具体打印任务，打印格式这块可以封装出一整套的处理代码（常用的

```

1 //打印接口，需要打印格式，派生此接口即可
2 public interface IPrint
3 {
4     bool Print(Graphics g); //返回值为是否还有打印需要打印的页内容，即是否打印结束
5 }

```

本 务 ， 接 或 本 务 即可 用本 处理 务

```

1 $.ajax({
2     url : "http://localhost:23333/print",
3     type : "POST",
4     contentType: "application/json;charset=utf-8",
5     data : "{ 'name': '大萝卜', 'age': 32, 'sex': true }",
6     dataType : "text",
7     success : function(result) {
8         if (result == "ok") {
9             alert("打印成功 ");
10        } else {
11            alert("打印出错: "+result )
12        }
13    },
14    error: function(msg) {
15        aler('Error: '+msg);
16    }
17 })

```

本

```

var web = new WebClient();
var txt = textBox1.Text;
byte[] bytearray = Encoding.UTF8.GetBytes(txt);
txt = Convert.ToBase64String(bytearray);
txt = System.Web.HttpUtility.UrlEncode(txt, Encoding.UTF8); //必须经过url编
web.Encoding = Encoding.UTF8;
web.Headers.Add("ContentType", "application/x-www-form-urlencoded");

try
{
    var ret = web.UploadString("http://127.0.0.1:1234", "POST", "args=" +
        textBox2.Text + Encoding.UTF8.GetString(Convert.FromBase64String(ret));
}
catch (WebException ex)
{
    var rsp = ex.Response as HttpWebResponse;
    if (rsp != null)
    {
        var code = (int)rsp.StatusCode;
    }
}

```



力量 亮园子希望

刷新 返回顶部

登录后才能查看或 立即 [登录](#) 或者 [逛逛](#) [园首](#)

【推荐】在 A P.NET 中运行 业务

【推荐】园子周边第二季 更 鼠标垫 没 g 鼠标垫

【推荐】阿里云云市场联合 园推出开 者商店 欢迎

【推荐】 力量 亮园子希望 期待您 园子



编辑推荐:

- 记 次 a . 服务器爆满 解决 程
- 因 我 次疏忽而带来 g a g . 新特性
- NET 中 轻量无负担 极 任务调度 h u Ta k
- P ha 复杂线 动画
- # 强 达式树 对象 深克隆

阅读排行:

- [记一次asp.net 8 服务器爆满的解决过程](#)
- [工业福利！用.NET快速开发物联网扫码器设备的通用扫码功能](#)
- [一款基于C#开发的通讯调试工具（支持Modbus RTU、MQTT调试）](#)
- [逆向WeChat \(二\)](#)
- [美团一面问我i++跟++i的区别是什么](#)