

IHandler 是 ASP.NET 处理实际操作的接口。在 MSDN 是这样定义的：使用自定义的 HTTP 处理程序同步处理 HTTP Web 请求而实现的协定。（注意：这里写的很清楚是同步 HTTP 请求如果是异步的话就要使用 IHttpAsyncHandler 接口程序了）。他包含一个 属性 IsReusable 用于获取当前 IHttpHandler 实例是否可用一般设置为 True. 一个方法 ProcessRequest(HttpContext context) 进行实际的操作过程。

自定义 IHttpHandler

- 1) 定义一个实现了 IHttpHandler 接口的类。
- 2) 在 Web.config 文件中注册这个类
- 3) 执行相应的 HttpRequest

下面是一个图片防盗链接的实例：

```

public class LinkProtectHandler: IHttpHandler
{
    #region IHttpHandler 成员

    public bool IsReusable
    {
        get { return true; }
    }

    public void ProcessRequest(HttpContext context)
    {
        //具体执行的操作
        //获取文件服务器端物理路径
        string fileName = context.Server.MapPath(context.Request.FilePath);
        //如果UrlReferrer为空则显示一张默认的禁止盗链的图片
        if (context.Request.UrlReferrer.Host==null)
        {
            context.Response.ContentType = "image/JPEG";
            context.Response.WriteFile("/LinkProtect.jpg");
        }
        else
        {
            //如果UrlReferrer中不包含自己站点的主机域名，则显示一张默认的禁止盗链图片
            if (context.Request.UrlReferrer.Host.IndexOf("mydomain.com")>0)
            {
                context.Response.ContentType = "image/JPEG";
                context.Response.WriteFile(fileName);
            }
            else
            {
                context.Response.ContentType = "image/JPEG";
                context.Response.WriteFile("/LinkProtect.jpg");
            }
        }
    }

    #endregion
}

```

<httpHandlers>

<remove verb="*" path="*.asmx"/>

<add verb="*" path="*.asmx" validate="false" type="System.Web.Script.Services.ScriptHandlerFactory, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35"/>

```

        <add verb="*" path="*.jpeg" validate="false" type="WebApplication3.LinkProt
ectHandler,WebApplication3"/>
    </httpHandlers>

```

自定义 IHttpHandlerFactory

IHttpHandlerFactory 在 MSDN 中是这样定义的：定义类工厂为创建新的 IHttpHandler 对象而必须实现的协定。他包含 2 个接口方法，GetHandler():返回实现 IHttp 接口的类的实例。ReleaseHandler():始工厂可以重复使用现有的处理程序实例。这个接口就是可以把很多实现 IHttpHandler 接口的方法放在 Factory 中在 Web.config 注册时主要注册这个类即可，在使用的时候交由 Factory 进行判断到底使用哪个 IHttpHandler 实例。

```

public class CustomerHttpHandlerFactory:IHttpHandlerFactory
{
    #region IHttpHandlerFactory 成员

    public IHttpHandler GetHandler(HttpContext context, string requestType, string url, string pathTranslated)
    {
        //获取文件服务器物理路径
        string path = context.Request.PhysicalPath;
        //获取文件名后缀名
        string exstention = Path.GetExtension(path);
        //进行判断，交由相应的处理程序
        if (exstention==".rss")
        {
            return new RssHandler();
        }
        else if (exstention==".atmo")
        {
            return new ATMOHandler();
        }
        else
        {
            return null;
        }
    }

    public void ReleaseHandler(IHttpHandler handler)
    {
    }

    #endregion
}

```

在 Web.Config 里面进行如下定义：

```

<httpHandlers>
    <remove verb="*" path="*.asmx"/>
    <add verb="*" path="*.rss,*.atom" validate="false" type="WebApplication3.Cu
stomerHttpHandlerFactory,WebApplication3"/>
</httpHandlers>

```

IHttpAsyncHandler 异步接口

IHttpAsyncHandler 在 MSDN 中是这样描述的：定义 HTTP 异步处理程序对象必须实现的协定。他包含 3 个方法：

BeginProcessRequest：启动对 HTTP 处理程序的异步调用，EndProcessRequest：进程结束时提供异步处理 End 方法。ProcessRequest：通过实现 IHttpHandler 接口的自定义 HttpHandler 启用 HTTP Web 请求的处理。（继承自 IHttpHandler。）和一个 IsReusable 属性。