链滴

阿里云开放搜索（Open Search）C# 版签名代码 ·Signature 生成算法

作者：LyZane

原文链接：https://ld246.com/article/1493083553805

来源网站：链滴

根据官方文档的描述，踩了若干个坑，把源码贴出来，做个分享。

包含签名过程的代码：

```
private static IResponsePack RequestAliyun(string apiPath, dynamic data, string method = "G
T")
{
    var dic = new RouteValueDictionary(data);
    dic["Version"] = "v2";
    dic["AccessKeyId"] = AccessKeyId;

    dic["SignatureMethod"] = "HMAC-SHA1";
    dic["Timestamp"] = DateTime.UtcNow.ToString("yyyy-MM-ddTHH:mm:ssZ");

    dic["SignatureVersion"] = "1.0";
    dic["SignatureNonce"] = Guid.NewGuid().ToString();

    //开始生成Signature
    string[] array = dic.OrderBy(a => a.Key, StringComparer.Ordinal).Select(a => PercentEncod
(a.Key) + "=" + PercentEncode(a.Value.ToString())).ToArray();
    string dataStr = string.Join("&", array);
    string signStr = method + "&" + PercentEncode("/") + "&" + PercentEncode(dataStr);

    HMACSHA1 myhmacsha1 = new HMACSHA1(Encoding.UTF8.GetBytes(AccessKeySecret +
&"));
    byte[] byteArray = Encoding.UTF8.GetBytes(signStr);
    MemoryStream stream = new MemoryStream(byteArray);
    string signature = Convert.ToBase64String(myhmacsha1.ComputeHash(stream));

    dic["Signature"] = signature;

    //这里的WebBrowser对象是我自己封装的一个WebClient类。
    //在NuGet中可以下载到：https://www.nuget.org/packages/Zane.Common.WebBrowser
    //源码：https://github.com/LyZane/Zane.Common/tree/master/Zane.Common.WebBrowse

    WebBrowser browser = new WebBrowser(new RequestConfig() { Method = method });
    return browser.DownloadJson(new Uri(APIHost, apiPath), dic);
}

private static string PercentEncode(string value)
{
    return UpperCaseUrlEncode(value)
        .Replace("+", "%20")
        .Replace("*", "%2A")
        .Replace("%7E", "~");
}

private static string UpperCaseUrlEncode(string s)
{
    char[] temp = HttpUtility.UrlEncode(s).ToCharArray();
    for (int i = 0; i < temp.Length - 2; i++)
    {
        if (temp[i] == '%')
        {
```

```
            temp[i + 1] = char.ToUpper(temp[i + 1]);
            temp[i + 2] = char.ToUpper(temp[i + 2]);
        }
    }
    return new string(temp);
}
```

调用，以使用索引进行搜索为例：

```
var response = RequestAliyun("search", new
{
    query = BuildQueryString(where),
    index_name = IndexName
});
```

在此过程中遇到的最大坑就是：使用C#的 HttpUtility.UrlEncode() 编码得到的形如 %xx%xx%xx%xx 的内容是小写的，Java的是大写的。

后来也是在博客园查到一篇帖子提示到了这点，然后才成功。