# Storage Table Rest API调用示例

#### 问题描述：

在使用[参考](https://docs.microsoft.com/en-us/rest/api/storageservices/query-tables)直接基于Rest API操作Azure Storage Table的过程中，[官方](https://docs.microsoft.com/en-us/rest/api/storageservices/authentication-for-the-azure-storage-services)提供了两种方式进行认证，分别为：SharedKey和ShareKeyLite。在请求Header过程中加入Date（或x-ms-date）和Authorization参数即可完成认证。

* **添加认证信息格式：**

|  |
| --- |
| x-ms-date: date\_and\_time  Authorization: scheme\_name account\_name:signature |

从Authorization信息来看，scheme\_name参数对应SharedKey或ShareKeyLite，account\_name对应存储的账户名称，都很容易获取，难点主要在signature参数的获取。下面分别基于C#给出构建两种认证方式的字符串：

* **Sharedkey签名字符串：**

|  |
| --- |
| **string** stringToSign = **string**.Format("{0}**\n**{1}**\n**{2}**\n**{3}**\n**/{4}/{5}",  request.Method,  request.Headers["Content-MD5"],  request.Headers["Content-Type"],  request.Headers["x-ms-date"],  account,  resource); |

* **SharedkeyList签名字符串：**

|  |
| --- |
| **string** stringToSign = **string**.Format("{0}**\n**/{1}/{2}",  request.Headers["x-ms-date"],  account,  resource); |

* **认证Authorization获取：**

|  |
| --- |
| **var** sharedKey = Convert.FromBase64("the\_sharedkey\_from\_the\_azure\_web");    **var** hasher = new HMACSHA256(sharedKey);    **string** signature = hasher.ComputeHash(Encoding.UTF8.GetBytes(strignToSign));  **string** authorizationHeader = **string**.Format("SharedKey {0}:{1}",  \_account,  Convert.ToBase64String(signature)); |

#### 完整测试代码：

|  |
| --- |
| using System;  using System.Globalization;  using System.IO;  using System.Net;  using System.Security.Cryptography;  using System.Text;  using System.Xml;  namespace test1  {  class Program  {  static void Main(string[] args)  {  string Resource = "Tables";  string \_Account = "yunewstoragetest";//storage account  string \_Secret = "xzWNEIf4NhrDh1jie9CEdCkGNXfKVqdnGlJ3BlB7jcoj1w829L4F/t3+oh/qPLq1jvygUdR/HTIvqjRiz9GNkA==";//key  //Create the web request  string url = "https://yunewstoragetest.table.core.chinacloudapi.cn/" + Resource;  HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(url);  request.ContentLength = 0;  request.Method = "GET";  //Add a date header to the request  request.Headers.Add("x-ms-date", DateTime.UtcNow.ToString("R", CultureInfo.InvariantCulture));  // Sign the request  string signature = "GET\n";  signature += "\n";  signature += "\n";  signature += request.Headers["x-ms-date"] + "\n";  int q = Resource.IndexOf("?");  if (q > 0) Resource = Resource.Substring(0, q);  signature += "/" + \_Account + "/" + Resource;  // Hash-based Message Authentication Code (HMAC) using SHA256 hash  HMACSHA256 hasher = new HMACSHA256(Convert.FromBase64String(\_Secret));  string authH = "SharedKey " + \_Account + ":" + Convert.ToBase64String(hasher.ComputeHash(Encoding.UTF8.GetBytes(signature)));  //// Authorization header  request.Headers.Add("Authorization", authH);  // Processing the results  using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())  {  using (StreamReader r = new StreamReader(response.GetResponseStream()))  {  string xml = r.ReadToEnd();  int ret = (int)response.StatusCode;  if (ret == 200)  {  XmlDocument doc = new XmlDocument();  doc.LoadXml(xml);  XmlNodeList nodes = doc.GetElementsByTagName("d:TableName");  Console.WriteLine("Tables in Account: " + \_Account);  foreach (XmlNode n in nodes)  {  Console.WriteLine(n.InnerText);  }  }  else  {  Console.WriteLine(xml);  }  Console.WriteLine();  }  }  Console.WriteLine("-----------------------------------------");  Console.ReadKey(true);  }  }  } |

#### 测试结果：



#### 参考链接：

Accessing Azure Tables via REST：<https://blogs.msdn.microsoft.com/rxg/2009/04/02/accessing-azure-tables-via-rest/>

Authenticating against Azure Table Storage：<http://blog.einbu.no/2009/08/authenticating-against-azure-table-storage/>

Authentication for the Azure Storage Services：<https://docs.microsoft.com/en-us/rest/api/storageservices/authentication-for-the-azure-storage-services>

Operations on Tables：<https://docs.microsoft.com/en-us/rest/api/storageservices/operations-on-tables>