使用EventHubClient连接IoT中心的正确连接字符串

Azure IoT中心支持两种通信方式，一个是设备到云（以下简称D2C），另一个是云到设备（以下简称C2D）。对于这两种通信，都会有相应的发送端和消费端。微软也提供了相应的SDK以供开发者方便使用。以.NET为例（其他语言SDK的源代码分享在此[Github库](https://github.com/Azure/azure-iot-sdks)中）

* D2C消息发送端：[DeviceClient](https://github.com/Azure/azure-iot-sdk-csharp/blob/1f5a259eee0d178db42d2bf866b80c9846adebba/device/Microsoft.Azure.Devices.Client/DeviceClient.cs), [Nuget](https://www.nuget.org/packages/Microsoft.Azure.Devices.Client/), namespace: Microsoft.Azure.Devices.Client

|  |
| --- |
| var deviceConnectionString = "HostName=<iothub\_host\_name>;DeviceId=<device\_id>;SharedAccessKey=<device\_key>"  var deviceClient = DeviceClient.CreateFromConnectionString(deviceConnectionString); |

* D2C消息消费端：EventHubClient, [Nuget](https://www.nuget.org/packages/WindowsAzure.ServiceBus/), namespace: Microsoft.ServiceBus.Messaging

|  |
| --- |
| var eventHubCompatibleConnStr = “Endpoint=<eventhub\_compatible\_endpoint>;SharedAccessKeyName=<iothub\_sas\_policy\_name>;SharedAccessKey=<iothub\_sas\_policy\_key>;EntityPath=<eventhub\_compatible\_name>"  var eventHubClient = EventHubClient.CreateFromConnectionString(eventHubCompatibleConnStr); |

* C2D消息发送端：[ServiceClient](https://github.com/Azure/azure-iot-sdk-csharp/blob/1f5a259eee0d178db42d2bf866b80c9846adebba/service/Microsoft.Azure.Devices/ServiceClient.cs), [Nuget](https://www.nuget.org/packages/Microsoft.Azure.Devices/), namespace: Microsoft.Azure.Devices

|  |
| --- |
| var iotHubConnectionString = “HostName=<iothub\_host\_name>;SharedAccessKeyName=<iothub\_sas\_policy\_name>;SharedAccessKey=<iothub\_sas\_policy\_key>;”  var serviceClient = ServiceClient.CreateFromConnectionString(iotHubConnectionString, <transport\_type>); |

* C2D消息消费端：[DeviceClient](https://github.com/Azure/azure-iot-sdk-csharp/blob/1f5a259eee0d178db42d2bf866b80c9846adebba/device/Microsoft.Azure.Devices.Client/DeviceClient.cs), [Nuget](https://www.nuget.org/packages/Microsoft.Azure.Devices.Client/), namespace: Microsoft.Azure.Devices.Client

|  |
| --- |
| var deviceConnectionString = "HostName=<iothub\_host\_name>;DeviceId=<device\_id>;SharedAccessKey=<device\_key>"  var deviceClient = DeviceClient.CreateFromConnectionString(deviceConnectionString); |

可以看到其中消费D2C（设备到云）消息是比较特殊的，它不是调用IoT中心的SDK，而是使用服务总线SDK里的EventHubClient。这是因为IoT 中心公开[事件中心](https://www.azure.cn/documentation/services/event-hubs/)兼容的终结点来接收设备到云的消息 。这样就可以重用现有的事件中心所支持的方式来消费IoT中心的D2C消息，包括使用EventHubClient直接读取，使用事件处理程序主机（EventProcessorHost），或者使用Apache Storm等等。好处很明显，但是有时候也会产生混淆，**经常容易出错的地方就是使用IoT中心的连接字符串来创建EventHubClient**。这样的使用是会产生问题的，比如连接不上或者连接容易断开等等。**正确的使用方式是使用事件中心兼容的连接字符串来创建EventHubClient**。所以在IoT中心的管理门户里面提供了相应的事件中心兼容的名称和端点信息，以供构建正确的连接字符串。

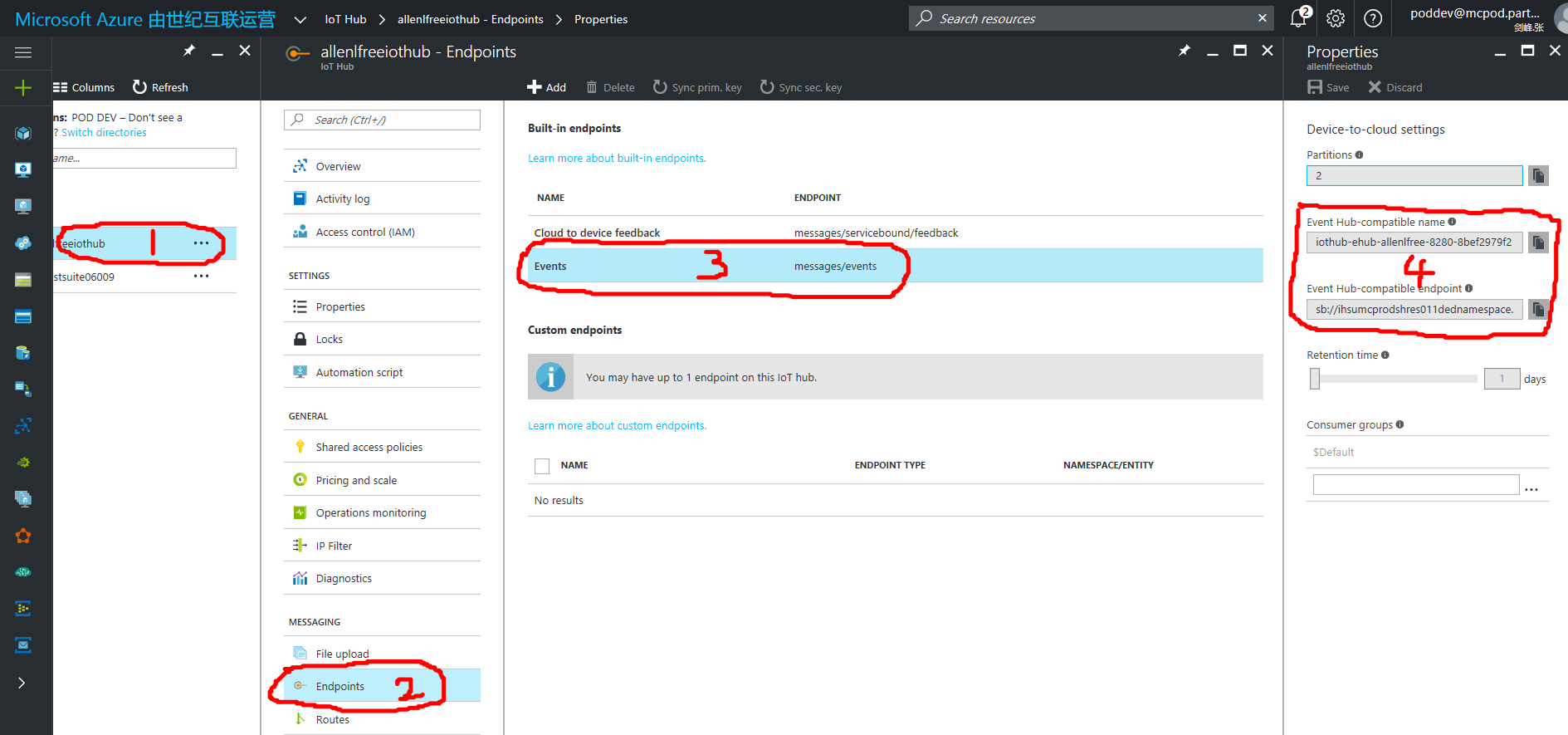
**事件中心兼容的连接字符串格式**：

Endpoint=<eventhub\_compatible\_endpoint>;SharedAccessKeyName=<iothub\_sas\_policy\_name>;SharedAccessKey=<iothub\_sas\_policy\_key>;EntityPath=<eventhub\_compatible\_name>

**在IoT中心的管理门户上获取相应值**：

|  |  |
| --- | --- |
| eventhub\_compatible\_endpoint | {Your IoT Hub} -> Endpoints -> Built-in endpoints -> Events -> Event Hub-compatible endpoint |
| iothub\_sas\_policy name and key | {Your IoT Hub} -> Shared access policies -> {any policy with Device connect permission} |
| eventhub\_compatible\_name | {Your IoT Hub} -> Endpoints -> Built-in endpoints -> Events -> Event Hub-compatible name |

**截图**：



**代码示例**：

var eventHubCompatibleConnStr = “Endpoint=<eventhub\_compatible\_endpoint>;SharedAccessKeyName=<iothub\_sas\_policy\_name>;SharedAccessKey=<iothub\_sas\_policy\_key>;EntityPath=<eventhub\_compatible\_name>"

var eventHubClient = EventHubClient.CreateFromConnectionString(eventHubCompatibleConnStr);