Azure IoT Hub – Csharp Device Client SDK

|  |  |
| --- | --- |
| Feature Area | Azure IoT |
| Doc Owner | ED |

Change History

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 09/28/2015 | AR | Ported document to Public repo with CSharp Device Client SDK Content Only |

Contents

[1 Connection Strings 2](#_Toc431218674)

[2 .NET 2](#_Toc431218675)

[2.1 SDK 2](#_Toc431218676)

[2.1.1 DeviceClient 2](#_Toc431218677)

[2.1.1.1 DeviceClient class 2](#_Toc431218678)

[2.1.1.1.1 DeviceClient Constructor 2](#_Toc431218679)

[2.1.1.1.2 DeviceClient Settings 3](#_Toc431218680)

[2.1.1.1.3 DeviceClient methods 3](#_Toc431218681)

# Connection Strings

Connection strings are used as container of all the information required by a client to successfully connect and authenticate to an IoT Hub. Connection Strings are property bags serialized in the following way:

Key1=value1;key2=value2;key3=value3

Currently the possible properties are:

|  |  |  |
| --- | --- | --- |
| Property name |  | Description |
| HostName | required | Endpoint of iot hub without protocol.  E.g. xxxxx.azure-devices.net |
| DeviceId | Required only for device connection string | This is URL-encoded when serialized in the connection string. |
| CredentialScope | Required if DeviceId is present. | Device or IoTHub. Specifies if this connection string can authenticate at the hub level (i.e. registry, service, or as any device), or device level (as an individual device). |
| SharedAccessKeyName | Required if SharedAccessSignature is not specified. | The name of the Shared Access policy. |
| SharedAccessKey | Required if SharedAccessKeyName is specified | The key value in base64url |
| SharedAccessSignature | Required if SharedAccessKeyName is not specified. | The url encoded token. |

# .NET

## SDK

### DeviceClient

The **DeviceClient** class uses the same **Message** class as the **ServiceClient** class to represent a message.

#### DeviceClient class

##### DeviceClient Constructor

|  |  |
| --- | --- |
| Method signature | public static DeviceClient CreateFromConnectionString(string connectionString);  public static DeviceClient CreateFromConnectionString(string connectionString, DeviceClient.TransportType transportType);  public static DeviceClient Create(string hostname, AuthenticationMethod authenticationMethod);  public static DeviceClient Create(string hostname, AuthenticationMethod authenticationMethod, DeviceClient.TransportType transportType) |
| Inputs | connectionString: a device-scoped connection string granting **DeviceConnect** permissions  authenticationMethod: A device-scoped authentication mthod.  transportType: AMQP or HTTP  hostname: hostname of the iot hub |
| Outputs | A **DeviceClient** instance |
| Exceptions | ArgumentException – Bad connection string, bad authenticationMethod (e.g. not device-scoped) |

##### DeviceClient Settings

The **DeviceClient** class exposes the following settings as a property.

|  |  |  |
| --- | --- | --- |
| Property name | Type | Description |
| Settings.OperationTimeout | Timespan | Timeout for all communications to the service. |
| Settings.TransportType | TransportType | Read-only.  The transport protocol to be used by the client.  Possible values: “Http1”,”Amqp”. |

##### DeviceClient methods

###### OpenAsync/CloseAsync

**DeviceClient** instances should be opened and closed properly to avoid resource leaks.

|  |  |
| --- | --- |
| Method signature | public Task OpenAsync();  public Task CloseAsync(); |
| Inputs | -- |
| Outputs | -- |
| Exceptions | ArgumentException – Bad connection string  UnauthorizedException –  HubNotFound – if IoT Hub does not exist  OperationThrottledException – Service requires exponential back-off.  TimeoutException – Service did not respond in time |

###### SendEventAsync

Use to send telemetry events.

|  |  |
| --- | --- |
| Method signature | public abstract Task SendEventAsync(Message message); |
| Inputs | The **Message** instance representing the event to send. No special properties need to be set. |
| Outputs | -- |
| Exceptions | ArgumentException – Bad connection string  UnauthorizedException –  HubNotFound – if IoT Hub does not exist  DeviceMessageTooLargeException-  OperationThrottledException – Service requires exponential back-off.  TimeoutException – Service did not respond in time |

###### SendEventBatchAsync

Use to send telemetry events.

|  |  |
| --- | --- |
| Method signature | public Task SendEventBatchAsync(IEnumerable<Message> messages); |
| Inputs | The **Message** instances representing the event to send. No special properties need to be set. |
| Outputs | -- |
| Exceptions | ArgumentException – Bad connection string  UnauthorizedException –  HubNotFound – if IoT Hub does not exist  DeviceMessageTooLargeException-  OperationThrottledException – Service requires exponential back-off.  TimeoutException – Service did not respond in time |

Note:

###### ReceiveAsync/AbandonAsync/RejectAsync/CompleteAsync

Use the **ReceiveAsync** method to receive C2D messages.

|  |  |
| --- | --- |
| Method signature | public Task<Message> ReceiveAsync();  public Task<Message> ReceiveAsync(TimeSpan timeout); |
| Inputs | Timeout: the amount of time to wait before returning null. Default value if not specified.  Default: 1 minute. |
| Outputs | The received **Message** instance, null if a timeout occurs. |
| Exceptions | UnauthorizedException –  HubNotFound – if IoT Hub does not exist  OperationThrottledException – Service requires exponential back-off.  TimeoutException – Service did not respond in time |

Received messages should be completed or abandoned using the **AbandonAsync, RejectAsync** or **CompleteAsync** methods.

|  |  |
| --- | --- |
| Method signature | public Task AbandonAsync(string lockToken);  public Task RejectAsync(string lockToken);  public Task CompleteAsync(string lockToken);  Convenience methods to not expose the lockToken property.  public Task AbandonAsync(Message message);  public Task RejectAsync(Message message);  public Task CompleteAsync(Message message); |
| Inputs | lockToken: a property of the received message  message: the received message. |
| Outputs | -- |
| Exceptions | UnauthorizedException –  DeviceMessageLockLostException – Depending on the underlying protocol (e.g. in AMQP if the connection is lost and regained) the client can lose the lock token and not be able to complete/reject. This timeout has the same effect as if the message was abandoned.  HubNotFound – if IoT Hub does not exist  OperationThrottledException – Service requires exponential back-off.  TimeoutException – Service did not respond in time |

**Note**: an abandoned message will be re-enqueued in the per-device queue, and the **DeviceClient** instance will receive it again. A rejected message will be deleted from the queue and not received again by the device.