

IoT Datahub Java SDK

by Dasudian

Contents

1 IoT Datahub Java SDK	1
2 更新日志	2
3 第三方依赖	2
4 弃用列表	2
5 继承关系索引	2
5.1 类继承关系	2
6 类索引	2
6.1 类列表	2
7 类说明	3
7.1 ActionCallback类 参考	3
7.1.1 成员函数说明	3
7.2 DataHubClient.Builder类 参考	3
7.2.1 构造及析构函数说明	4
7.2.2 成员函数说明	4
7.3 Constants类 参考	6
7.3.1 类成员变量说明	6
7.4 DataHubClient类 参考	6
7.4.1 成员函数说明	7
7.5 Message类 参考	9
7.5.1 构造及析构函数说明	9
7.5.2 成员函数说明	9
7.6 ServiceException类 参考	10
7.6.1 成员函数说明	10
7.6.2 类成员变量说明	10
索引	13

1 IoT Datahub Java SDK

SDK提供了一个简单的基于MQTT协议与服务器交互的方法

SDK基于MQTT协议,传输实时的消息到大数点IoT云服务器

你可以收集设备上的数据发送到云上;也可以订阅某个topic,来接收云服务器推送的消息

如何使用SDK:

- 1: 创建一个客户端实例
- 2: 如果想接收消息,那么就订阅某个topic
- 3: 或者发送消息到服务器
- 4: 退出时,销毁该客户端

2 更新日志

Author	Date	Version	Note
Kevin Liang	6/27/2017	2.2.0	connect() 中 waitForCompletion() 传入超时时间, 解决程序可能一直阻塞问题
Eden Wang	4/21/2017	2.1.0	暴露 cleansession 和 订阅时的 qos 选项
Jack Liu	3/13/2017	2.0.0	客户端全面升级, 版本更新为 2.0.0
Jack Liu	2/28/2017	1.1.1	修改了上传图片的 API
Jack Liu	2/18/2017	1.1.0	1. 添加了上传图片的 api 2. 修改了发送消息函数和接收消息函数的参数类型

3 第三方依赖

依赖

org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.1.0

org.hamcrest:hamcrest-core:1.3

4 弃用列表

成员 [DataHubClient.publish](#) (String topic, [Message](#) msg, int QoS)

由于异步发送消息的数目有限制, 该方法不建议继续使用, 我们将在后期的版本废弃该方法。

5 继承关系索引

5.1 类继承关系

此继承关系列表按字典顺序粗略的排序:

ActionCallback	3
DataHubClient.Builder	3
Constants	6
DataHubClient	6
Exception	
ServiceException	10
Message	9

6 类索引

6.1 类列表

这里列出了所有类、结构、联合以及接口定义等, 并附带简要说明:

ActionCallback	3
DataHubClient.Builder	3

Constants	6
DataHubClient	6
Message	9
ServiceException	10

7 类说明

7.1 ActionCallback类 参考

Public 成员函数

- void [onMessageReceived](#) (String topic, byte[] payload)
- void [onConnectionStatusChanged](#) (boolean isConnected)

7.1.1 成员函数说明

7.1.1.1 onConnectionStatusChanged()

```
void onConnectionStatusChanged (
    boolean isConnected )
```

SDK与服务器的连接状态改变

参数

<i>isConnected</i>	true:连接成功 ; false:连接丢失
--------------------	------------------------

7.1.1.2 onMessageReceived()

```
void onMessageReceived (
    String topic,
    byte [] payload )
```

接收到发布的消息

参数

<i>topic</i>	发布的topic
<i>payload</i>	发布的内容

该类的文档由以下文件生成:

- ActionCallback.java

7.2 DataHubClient.Builder类 参考

Public 成员函数

- [Builder](#) (String instanceId, String instanceKey, String clientName, String clientId) throws ServiceException

- [Builder setCallback](#) ([ActionCallback](#) callback)
- [Builder setServerURL](#) (String serverURL)
- [Builder setDebug](#) (boolean debug)
- [Builder setCleanSession](#) (boolean cleansession)
- [DataHubClient build](#) ()

7.2.1 构造及析构函数说明

7.2.1.1 Builder()

```
Builder (
    String instanceId,
    String instanceKey,
    String clientName,
    String clientId ) throws ServiceException
```

通过该Builder构建DataHubClient实例。

参数

<i>instanceId</i>	用于大数点验证用户，保证客户端与服务器间的安全通信。 demo中的instanceId仅可以用于测试大数点IoT DataHub功能使用，如果您想正式使用大数点IoT服务，请联系大数点客服获取私有的instanceId。
<i>instanceKey</i>	用于大数点验证用户，保证客户端与服务器间的安全通信。 demo中的instanceKey仅可以用于测试大数点IoT DataHub功能使用，如果您想正式使用大数点IoT服务，请联系大数点客服获取私有的instanceKey。
<i>clientName</i>	客户端名字，可以填写任意的utf-8字符。 如果你有第三账号系统，并想将自己的账号系统与大数点服务器同步，那么你可以使用第三方账号的名字、昵称。 如果没有自己的账号系统，或者对该客户端名字不关心，可以使用随机的名字，但是不能填null。
<i>clientId</i>	客户端id，用于服务器唯一标记一个客户端，服务器通过该id向客户端推送消息； 注意：不同的客户端的id必须不同，如果有两个客户端有相同的id，服务器会关掉其中的一个客户端的连接。 可以使用设备的mac地址，或者第三方账号系统的id（比如qq号，微信号）。 如果没有自己的账号系统，则可以随机生成一个不会重复的客户端id。 或者自己指定客户端的id，只要能保证不同客户端id不同即可。

异常

ServiceException	有参数为null或长度为0时抛出异常。
----------------------------------	---------------------

7.2.2 成员函数说明

7.2.2.1 build()

```
DataHubClient build ( )
```

获取到DataHubClient实例。

返回

DataHubClient实例。

7.2.2.2 setCallback()

```
Builder setCallback (
    AsyncCallback callback )
```

设置回调函数，用于接收消息和监听SDK与服务器的连接状态。

参数

callback	回调函数。
----------	-------

返回

Builder对象。

7.2.2.3 setCleanSession()

```
Builder setCleanSession (
    boolean cleansession )
```

设置是否清除会话，默认为false

参数

cleansession	cleansession为false, 当客户端断线或下线后, 保存客户端订阅的topic和发送给客户端的所有消息. cleansession为true, 当客户端断线或下线后, 不保留客户端订阅的topic和发送给客户端的任何消息.
--------------	--

7.2.2.4 setDebug()

```
Builder setDebug (
    boolean debug )
```

设置是否打开调试功能，默认为false。

参数

debug	true:打开调试；false:关闭调试。
-------	-----------------------

返回

Builder对象。

7.2.2.5 setServerURL()

```
Builder setServerURL (
    String serverURL )
```

服务器地址，如果不设置，则默认使用大数点公有云测试服务器。

参数

<code>serverURL</code>	服务器的地址，加密连接方式tcp://host:port，非加密连接方式ssl://host:port; 其中ssl和tcp都必须为小写，host可以使用域名或ip地址，port表示对应的端口。
------------------------	---

返回

Builder对象。

该类的文档由以下文件生成:

- DataHubClient.java

7.3 Constants类 参考

静态 **Public** 属性

- static final String **COPYRIGHT** = "(c) 2016 Dasudian Technologies Ltd."
- static final int **CONNECTION_TIMEOUT** = 10000
- static final int **PING_TIME_S** = 20
- static final String **DEFAULT_MQTT_URL** = "tcp://try.iotdatahub.net:1883"

7.3.1 类成员变量说明

7.3.1.1 CONNECTION_TIMEOUT

```
final int CONNECTION_TIMEOUT = 10000 [static]
```

连接超时常量（单位毫秒）

7.3.1.2 DEFAULT_MQTT_URL

```
final String DEFAULT_MQTT_URL = "tcp://try.iotdatahub.net:1883" [static]
```

默认的服务器地址

7.3.1.3 PING_TIME_S

```
final int PING_TIME_S = 20 [static]
```

ping包发送间隔时间（单位秒）

该类的文档由以下文件生成:

- Constants.java

7.4 DataHubClient类 参考

类

- class **Builder**

Public 成员函数

- synchronized void [publish](#) (String topic, [Message](#) msg, int QoS) throws ServiceException
- synchronized void [sendRequest](#) (String topic, [Message](#) msg, int QoS, long timeout) throws ServiceException
- synchronized void [subscribe](#) (String topic, int QoS, long timeout) throws ServiceException
- synchronized void [unsubscribe](#) (String topic, long timeout) throws ServiceException
- synchronized void [destroy](#) ()

Protected 成员函数

- **DataHubClient** ([Builder](#) builder)

7.4.1 成员函数说明

7.4.1.1 [destroy\(\)](#)

synchronized void [destroy](#) ()
销毁客户端，并断开与服务器的连接。

7.4.1.2 [publish\(\)](#)

synchronized void [publish](#) (
 String *topic*,
 [Message](#) *msg*,
 int *QoS*) throws [ServiceException](#)
异步发送消息，SDK根据QoS设置来发送消息，无法知道消息发送成功或失败。

弃用 由于异步发送消息的数目有限制,该方法不建议继续使用,我们将在后期的版本废弃该方法.

参数

<i>topic</i>	主题名。
<i>msg</i>	消息内容，长度必须小于512k。
<i>QoS</i>	服务质量。

异常

ServiceException	失败时抛出异常。
----------------------------------	----------

7.4.1.3 [sendRequest\(\)](#)

synchronized void [sendRequest](#) (
 String *topic*,
 [Message](#) *msg*,
 int *QoS*,
 long *timeout*) throws [ServiceException](#)
同步发送消息，该方法会阻塞的等待消息发送完成，或者超时返回。

参数

<i>topic</i>	主题名。
<i>msg</i>	消息内容，长度必须小于512k。
<i>QoS</i>	服务质量。
<i>timeout</i>	超时时间，单位s，必须大于0。表示该函数最多阻塞多长时间。对于普通的文本消息，建议超时时间为10s。 注意：该函数超时返回不代表消息发送失败，仅表示在指定时间内没有接收到服务器的应答。

异常

ServiceException	失败时抛出异常。
----------------------------------	----------

7.4.1.4 subscribe()

```
synchronized void subscribe (  
    String topic,  
    int QoS,  
    long timeout ) throws ServiceException
```

订阅一个主题，该方法会阻塞的等待消息发送完成，或者超时返回。

参数

<i>topic</i>	主题名。
<i>QoS</i>	服务质量。
<i>timeout</i>	超时时间，单位s，必须大于0。表示该函数最多阻塞多长时间。对于普通的文本消息，建议超时时间为10s。 注意：该函数超时返回不代表消息发送失败，仅表示在指定时间内没有接收到服务器的应答。

异常

ServiceException	失败时抛出异常。
----------------------------------	----------

7.4.1.5 unsubscribe()

```
synchronized void unsubscribe (  
    String topic,  
    long timeout ) throws ServiceException
```

取消订阅一个主题，该方法会阻塞的等待消息发送完成，或则超时返回。

参数

<i>topic</i>	主题名。
<i>timeout</i>	超时时间，单位s，必须大于0。表示该函数最多阻塞多长时间。对于普通的文本消息，建议超时时间为10s。 注意：该函数超时返回不代表消息发送失败，仅表示在指定时间内没有接收到服务器的应答。

异常

ServiceException	失败时抛出异常。
----------------------------------	----------

该类的文档由以下文件生成:

- DataHubClient.java

7.5 Message类 参考

Public 成员函数

- [Message](#) (byte[] payload)
- byte [] [getPayload](#) ()
- void [setPayload](#) (byte[] payload)

7.5.1 构造及析构造函数说明

7.5.1.1 Message()

[Message](#) (
byte [] payload)

构造函数

参数

<i>payload</i>	消息的内容
----------------	-------

7.5.2 成员函数说明

7.5.2.1 getPayload()

byte [] getPayload ()

获取消息内容

返回

消息内容

7.5.2.2 setPayload()

void setPayload (
byte [] payload)

设置消息内容

参数

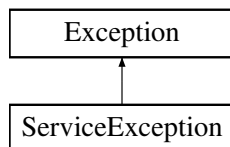
<i>payload</i>	消息内容
----------------	------

该类的文档由以下文件生成:

- Message.java

7.6 ServiceException 类 参考

类 ServiceException 继承关系图:



Public 成员函数

- **ServiceException** (int code)
- **ServiceException** (int code, String message)
- **ServiceException** (int code, Throwable t, String message)
- int [getCode](#) ()

静态 Public 属性

- static final int [ERROR_NONE](#) = 0
- static final int [ERROR_ILLEGAL_PARAMETERS](#) = -1
- static final int [ERROR_DISCONNECTED](#) = -2
- static final int [ERROR_UNACCEPT_PROTOCOL_VERSION](#) = -3
- static final int [ERROR_IDENTIFIER_REJECTED](#) = -4
- static final int [ERROR_SERVER_UNAVAILABLE](#) = -5
- static final int [ERROR_BAD_USERNAME_OR_PASSWD](#) = -6
- static final int [ERROR_UNAUTHORIZED](#) = -7
- static final int [ERROR_AUTHORIZED_SERVER_UNAVAILABLE](#) = -8
- static final int [ERROR_OPERATION_FAILURE](#) = -9
- static final int [ERROR_MESSAGE_TOO_BIG](#) = -10
- static final int [ERROR_NETWORK_UNREACHABLE](#) = -11
- static final int [ERROR_TIMEOUT](#) = -12
- static final int [ERROR_IO](#) = -300

7.6.1 成员函数说明

7.6.1.1 getCode()

```
int getCode ( )
```

获取错误码

返回

错误码

7.6.2 类成员变量说明

7.6.2.1 ERROR_AUTHORIZED_SERVER_UNAVAILABLE

```
final int ERROR_AUTHORIZED_SERVER_UNAVAILABLE = -8 [static]
```

认证服务器不可用

7.6.2.2 ERROR_BAD_USERNAME_OR_PASSWD

```
final int ERROR_BAD_USERNAME_OR_PASSWD = -6 [static]
```

错误的用户名和密码

7.6.2.3 ERROR_DISCONNECTED

```
final int ERROR_DISCONNECTED = -2 [static]
```

没有与服务器连接

7.6.2.4 ERROR_IDENTIFIER_REJECTED

```
final int ERROR_IDENTIFIER_REJECTED = -4 [static]
```

标识符已拒绝

7.6.2.5 ERROR_ILLEGAL_PARAMETERS

```
final int ERROR_ILLEGAL_PARAMETERS = -1 [static]
```

非法参数

7.6.2.6 ERROR_IO

```
final int ERROR_IO = -300 [static]
```

IO错误

7.6.2.7 ERROR_MESSAGE_TOO_BIG

```
final int ERROR_MESSAGE_TOO_BIG = -10 [static]
```

消息太大

7.6.2.8 ERROR_NETWORK_UNREACHABLE

```
final int ERROR_NETWORK_UNREACHABLE = -11 [static]
```

网络不可达

7.6.2.9 ERROR_NONE

```
final int ERROR_NONE = 0 [static]
```

没有错误

7.6.2.10 ERROR_OPERATION_FAILURE

```
final int ERROR_OPERATION_FAILURE = -9 [static]
```

操作失败

7.6.2.11 ERROR_SERVER_UNAVAILABLE

```
final int ERROR_SERVER_UNAVAILABLE = -5 [static]
```

服务器不可用

7.6.2.12 ERROR_TIMEOUT

```
final int ERROR_TIMEOUT = -12 [static]
```

超时

7.6.2.13 ERROR_UNACCEPT_PROTOCOL_VERSION

```
final int ERROR_UNACCEPT_PROTOCOL_VERSION = -3 [static]
```

协议版本不支持

7.6.2.14 ERROR_UNAUTHORIZED

```
final int ERROR_UNAUTHORIZED = -7 [static]
```

未认证

该类的文档由以下文件生成:

- ServiceException.java

Index

ActionCallback, 3

build

com::dasudian::iot::sdk::DataHubClient::Builder, 4

Builder

com::dasudian::iot::sdk::DataHubClient::Builder, 4

CONNECTION_TIMEOUT

com::dasudian::iot::sdk::Constants, 6

com::dasudian::iot::sdk::ActionCallback

onConnectionStatusChanged, 3

onMessageReceived, 3

com::dasudian::iot::sdk::Constants

CONNECTION_TIMEOUT, 6

DEFAULT_MQTT_URL, 6

PING_TIME_S, 6

com::dasudian::iot::sdk::DataHubClient

destroy, 7

publish, 7

sendRequest, 7

subscribe, 8

unsubscribe, 8

com::dasudian::iot::sdk::DataHubClient::Builder

build, 4

Builder, 4

setCallback, 5

setCleanSession, 5

setDebug, 5

setServerURL, 5

com::dasudian::iot::sdk::Message

getPayload, 9

Message, 9

setPayload, 9

com::dasudian::iot::sdk::ServiceException

ERROR_AUTHORIZED_SERVER_UNAVAILABLE, 10

ERROR_BAD_USERNAME_OR_PASSWD, 11

ERROR_DISCONNECTED, 11

ERROR_IDENTIFIER_REJECTED, 11

ERROR_ILLEGAL_PARAMETERS, 11

ERROR_IO, 11

ERROR_MESSAGE_TOO_BIG, 11

ERROR_NETWORK_UNREACHABLE, 11

ERROR_NONE, 11

ERROR_OPERATION_FAILURE, 11

ERROR_SERVER_UNAVAILABLE, 11

ERROR_TIMEOUT, 11

ERROR_UNACCEPT_PROTOCOL_VERSION, 11

ERROR_UNAUTHORIZED, 12

getCode, 10

Constants, 6

DEFAULT_MQTT_URL

com::dasudian::iot::sdk::Constants, 6

DataHubClient, 6

DataHubClient.Builder, 3

destroy

com::dasudian::iot::sdk::DataHubClient, 7

ERROR_AUTHORIZED_SERVER_UNAVAILABLE

com::dasudian::iot::sdk::ServiceException, 10

ERROR_BAD_USERNAME_OR_PASSWD

com::dasudian::iot::sdk::ServiceException, 11

ERROR_DISCONNECTED

com::dasudian::iot::sdk::ServiceException, 11

ERROR_IDENTIFIER_REJECTED

com::dasudian::iot::sdk::ServiceException, 11

ERROR_ILLEGAL_PARAMETERS

com::dasudian::iot::sdk::ServiceException, 11

ERROR_IO

com::dasudian::iot::sdk::ServiceException, 11

ERROR_MESSAGE_TOO_BIG

com::dasudian::iot::sdk::ServiceException, 11

ERROR_NETWORK_UNREACHABLE

com::dasudian::iot::sdk::ServiceException, 11

ERROR_NONE

com::dasudian::iot::sdk::ServiceException, 11

ERROR_OPERATION_FAILURE

com::dasudian::iot::sdk::ServiceException, 11

ERROR_SERVER_UNAVAILABLE

com::dasudian::iot::sdk::ServiceException, 11

ERROR_TIMEOUT

com::dasudian::iot::sdk::ServiceException, 11

ERROR_UNACCEPT_PROTOCOL_VERSION

com::dasudian::iot::sdk::ServiceException, 11

ERROR_UNAUTHORIZED

com::dasudian::iot::sdk::ServiceException, 12

getCode

com::dasudian::iot::sdk::ServiceException, 10

getPayload

com::dasudian::iot::sdk::Message, 9

Message, 9

com::dasudian::iot::sdk::Message, 9

onConnectionStatusChanged

com::dasudian::iot::sdk::ActionCallback, 3

onMessageReceived

com::dasudian::iot::sdk::ActionCallback, 3

PING_TIME_S

com::dasudian::iot::sdk::Constants, 6

publish

com::dasudian::iot::sdk::DataHubClient, 7

sendRequest

com::dasudian::iot::sdk::DataHubClient, 7

ServiceException, 10

setCallback

com::dasudian::iot::sdk::DataHubClient::Builder, 5

setCleanSession

com::dasudian::iot::sdk::DataHubClient::Builder, 5

setDebug
 com::dasudian::iot::sdk::DataHubClient::Builder, [5](#)
setPayload
 com::dasudian::iot::sdk::Message, [9](#)
setServerURL
 com::dasudian::iot::sdk::DataHubClient::Builder, [5](#)
subscribe
 com::dasudian::iot::sdk::DataHubClient, [8](#)
unsubscribe
 com::dasudian::iot::sdk::DataHubClient, [8](#)