

ALOOH MQTT SDK Guideline for Android

2015.05.08

CRESPRIT Inc.

MQTT SDK Guideline for Android	Date modified(Version)
	05/08/2015

Table of Contents

1.	OVERVIEW.....	3
2.	배포파일	4
3.	APIS.....	5
2.1	ALOOH CLOUD API	5
2.1.1	AloohCloud_LogIn.....	5
2.1.2	AloohCloud_GetDeviceList	5
2.1.3	AloohCloud_GetKeyInfo.....	5
2.1.4	AloohCloud_GetSensorList	6
2.2	ALOOH MQTT API.....	7
2.2.1	AloohMessage_SetPublishData	7
2.2.2	AloohMessage_Publish	7
2.2.3	AloohMessage_ResetPublishData()	8
2.2.4	AloohMessage_Subscribe	8
2.2.5	AloohMessage_Disconnect()	9
4.	APIS WORK FLOW	10

MQTT SDK Guideline for Android	Date modified(Version)
	05/08/2015

1. Overview

본 문서는 Android를 이용하여 ALOOH MQTT 메시지 서버와 MQTT Application간의 센서데이터를 MQTT 메시지 형태로 전달 및 수신하기 위해 사용되는 ALOOH MQTT SDK에 대한 가이드입니다.

SDK에서 제공하는 API는 크게 ALOOH Cloud 서버와 통신하는 API(API Prefix : AloohCloud_)와 ALOOH MQTT 메시지 서버와 통신하는 API(API Prefix : AloohMessage_)로 나뉘어 집니다.

ALOOH Cloud API는 ALOOH Cloud 서버에 로그인하여 사용자 보유장치에 대한 정보를 가져오거나 사용자 장치에 등록되어 있는 센서에 대한 정보를 가져옵니다.

ALOOH MQTT API는 MQTT 메시지를 Publish하거나 Subscribe 합니다.

SDK에서 전달하는 모든 MQTT 메시지는 JSON 타입입니다.

MQTT SDK Guideline for Android	Date modified(Version)
	05/08/2015

2. 배포파일

ALOOH MQTT용 안드로이드 App 개발을 위해 아래의 파일이 배포됩니다. 안드로이드 App 개발을 위해서는 Eclipse를 기준으로 프로젝트의 libs 디렉토리 안에 위치시켜야 합니다. jar파일은 크게 하나의 클래스로 구성되어 있으며 해당 클래스는 디바이스와 인증키에 대한 정보를 구성하는 각각의 inner 클래스로 구성되어 있습니다.

- **mqttSDK.jar** : MQTT Engine 및 ALOOH Cloud API, ALOOH MQTT API가 포함된 jar 파일

- MQTTSDK.class

- MQTTSDK.DeviceInfo.class(InnerClass)

Method	Description
getDeviceName	디바이스명
getTopicId	Topic ID
getActivationCode	Activation Code
getProductName	프로젝트 명
getAuid	AUID(디바이스 고유 ID)

- MQTTSDK.KeyInfo.class(InnerClass)

API	Description
getTopicId	Topic ID
getKeyLabel	Key Label
getAccessKey	Publish, Subscribe를 하기 위한 ALOOH MQTT 서버 인증키

3. APIs

※ 제공되는 대부분의 API는 네트워크를 이용하기 때문에 ANR을 방지하기 위해서 메인쓰레드(UI)이외 별도 쓰레드에서 호출하여야 합니다.

2.1 ALOOH CLOUD API

2.1.1 AloohCloud_LogIn

Prototype	static String AloohCloud_LogIn(String _email, String _passwd)	
Description	ALOOH Cloud 서버로 로그인을 통해 로그인 키를 얻습니다.	
Parameters	String _email	Alooh cloud 서버 계정
	String _passwd	해당 계정에 대한 패스워드
Return	String	로그인 키
Example	String Loginkey = MQTTSdk.AloohCloud_LogIn(m_pId, m_pPasswd);	

2.1.2 AloohCloud_GetDeviceList

Prototype	static ArrayList<DeviceInfo> AloohCloud_GetDeviceList(String _loginKey)	
Description	사용자가 보유한 모든 장치에 대한 정보를 가져옵니다.	
Parameters	String _loginKey	로그인을 통해 전달되는 Alooh cloud 서버 계정에 대한 로그인 키
Return	ArrayList<DeviceInfo>	장치정보 리스트
Example	<pre>ArrayList<DeviceInfo> deviceList = null; deviceList = MQTTSdk.AloohCloud_GetDeviceList(m_pLoginKey);</pre>	

2.1.3 AloohCloud_GetKeyInfo

Prototype	static ArrayList<KeyInfo> AloohCloud_GetKeyInfo(String _aid)	
Description	<p>Argument _aid를 가지는 장치의 키 정보들을 가져옵니다.</p> <p>_aid는 AloohCloud_GetDeviceList() API를 통해 얻을 수 있습니다.</p> <p>KeyInfo의 AccessKey는 publish시 ALOOH MQTT 서버에 대한 장치의 인증키로서 사용되며 ALOOH Cloud 서버에서 추가할 수 있으며 이로 인해</p>	

MQTT SDK Guideline for Android	Date modified(Version)
	05/08/2015

	서버로부터 2개 이상의 KeyInfo가 전달 될 수도 있습니다.	
Parameters	String _auid	장치 AUID
Return	ArrayList<KeyInfo>	장치에 대한 키 정보 리스트
Example	<pre>ArrayList<KeyInfo> keyInfoList = null; keyInfoList = MQTTSDK.AloohCloud_GetKeyInfo(deviceList.getAuid());</pre>	

2.1.4 AloohCloud_GetSensorList

Prototype	static ArrayList<String> AloohCloud_GetSensorList(String _topicId)	
Description	argument _topicId에 대응하는 센서의 이름들을 가져옵니다. _topicId 는 AloohCloud_GetDeviceList() API를 통해 얻을 수 있습니다.	
Parameters	String _topicId	장치의 topicId
	String _accessKey	ALOOH MQTT 서버에 대한 인증키
Return	ArrayList<String>	장치에 등록된 센서 이름들
Example	<pre>ArrayList<KeyInfo> KeyList = MQTTSDK.AloohCloud_GetKeyInfos(deviceInfo.getAuid()); ArrayList<String> sensorList = MQTTSDK.AloohCloud_GetSensorList(deviceInfo.getTopicId, KeyList.get(0).getAccessKey());</pre>	

MQTT SDK Guideline for Android	Date modified(Version)
	05/08/2015

2.2 ALOOH MQTT API

2.2.1 AloohMessage_SetPublishData

Prototype	static void AloohMessage_SetPublishData(String _sensorName, String _value, String _time)	
Description	publish하기 위한 데이터들을 추가합니다. 해당 API를 통해서 하나의 센서에 대한 복수의 데이터를 동시에 보낼 수 있으며, 또한 복수의 센서에 대한 데이터를 동시에 보낼 수도 있습니다.	
Parameters	_sensorName	GetSensorList() API를 통해 얻은 센서명
	_value	전달하기 위한 데이터
	_time	데이터 추출 시간(UTC형식)
Return	None	
Example	<pre>SimpleDateFormat formatUTC = new SimpleDateFormat("yyyy-MM-dd HH:mm:ssZ"); formatUTC.setTimeZone(TimeZone.getTimeZone("UTC")); String time = formatUTC.format(new Date()); MQTTSDK.AloohMessage_SetPublishData(SensorName, value, time);</pre>	

2.2.2 AloohMessage_Publish

Prototype	static void AloohMessage_Publish(String _topicId, String _accessKey)	
Description	<p>AloohMessage_SetPublishData() API를 통해 설정한 데이터를 주어진 topicId로 publish 합니다. 이를 통해 동일 topicId로 subscribe를 하고 있는 여타의 모든 클라이언트들은 이 데이터를 수신하게 됩니다.</p> <p>topicId는 AloohCloud_GetDeviceInfo()API를 통해 얻을 수 있습니다.</p> <p>API를 사용하여 MQTT 메시지를 publish하기 위해서는 먼저 AloohMessage_SetPublishData() API를 이용하여 전달하려고 하는 데이터를 설정하여야만 합니다.</p> <p>accessKey는 AloohCloud_GetKeyInfo()API를 통해 얻을 수 있습니다.</p>	
Parameters	_topicId	publish하려고 하는 TopicId

MQTT SDK Guideline for Android	Date modified(Version)
	05/08/2015

	<code>_accessKey</code>	ALOOH MQTT 서버에 대한 인증키
Return	None	
Example	<pre>SimpleDateFormat formatUTC = new SimpleDateFormat("yyyy-MM-dd HH:mm:ssZ"); formatUTC.setTimeZone(TimeZone.getTimeZone("UTC")); String time = formatUTC.format(new Date()); MQTTSDK.AloohMessage_SetPublishData("temperature_Sensor","26.7",time.toString()); KeyInfoList = MQTTSDK.AloohCloud_GetKeyInfos(deviceInfo.getAuid()); MQTTSDK.AloohMessage_Publish(deviceInfo.getTopicId(), KeyInfoList.get(0).getAccessKey());</pre>	

2.2.3 AloohMessage_ResetPublishData()

Prototype	<code>static void AloohMessage_ResetPublishData()</code>
Description	AloohMessage_SetPublishData() API를 통해 설정된 모든 Publish데이터를 초기화 시킵니다.
Parameters	None
Return	None
Example	<code>MQTTSDK.AloohMessage_ResetPublishData();</code>

2.2.4 AloohMessage_Subscribe

Prototype	<code>static void AloohMessage_Subscribe(String _topicId, String _accessKey, ISubscribeListener _listener)</code>	
Description	입력된 topicID에 대해 subscribe한다. 여타의 MQTT Client가 동일 topicID로 publish를 수행하면 해당 MQTT 엔진은 _listener로 등록한 콜백함수를 메시지와 함께 호출한다.	
Parameters	<code>_topicId</code>	subscribe하려고 하는 topicId
	<code>_accessKey</code>	ALOOH MQTT 서버에 대한 인증키

MQTT SDK Guideline for Android	Date modified(Version)
	05/08/2015

	_listener	메시지가 수신될 때 호출되는 콜백 리스너
Return	None	
Example	<pre> ArrayList<KeyInfo> KeyList = null; KeyList = MQTTSDK.AloohCloud_GetKeyInfos(deviceInfo.getAuid()); MQTTSDK.AloohMessage_Subscribe(deviceInfo.getTopicId(), KeyList.get(0).getAccessKey(), listener); </pre>	

2.2.5 AloohMessage_Disconnect()

Prototype	static void AloohMessage_Disconnect()
Description	AloohMqtt 서버와의 연결을 해지합니다.
Parameters	None
Return	None
Example	<pre> MQTTSDK.AloohMessage_Disconnect(); </pre>

4. APIs Work Flow

- Publish

1. *AloohCloud_Login(Email, Passwd) => LoginKey* 획득
2. *AloohCloud_GetDeviceList(LoginKey) => DeviceInfo* 획득(AUID, TopicId)
3. *AloohCloud_GetKeyInfos(AUID) => KeyInfo* 획득(AccessKey)
4. *AloohCloud_GetSensorList(TopicId, AccessKey) => 센서명 리스트*
5. *AloohMessage_SetPublishData(센서명, 데이터, 시간(UTC))*
6. *AloohMessage_Publish(TopicID, AccessKey)*

- Subscribe

1. *AloohCloud_Login(Email, Passwd) => LoginKey* 획득
2. *AloohCloud_GetDeviceList(LoginKey) => DeviceInfo* 획득(AUID, TopicId)
3. *AloohCloud_GetKeyInfos(AUID) => KeyInfo* 획득(AccessKey)
4. *AloohMessage_Subscribe(TopicID, AccessKey)*