Qualcomm Car-to-Cloud Platform

IoT Wrapper Usage Document

Version No.3.1

	Prepared By / Last Updated By	Reviewed By	Approved By
Name	Jean Johnson		
Role	Developer		
Signature			
Date	July 07, 2021		

Table of Contents

1 Project Overview

We have developed an interface adapter for IoT Gateway thing management and for Device Communication to IoT Gateway, so that applications, can consume it for Device management specific operations and for communication to IoT Gateway.

1.1 Interface Projects

IoT Gateway Thing Management

1. Create Thing: Create a new Device will all required resources
2. Delete Thing: Delete a Device and all associated resources

Device Connection to IoT Gateway

1. Connect to IoT Gateway
2. Publish to topic
3. Subscribe to topic
4. Disconnect from IoT Gateway
5. Processor Interface: Process messages received from IoT

1.2 Implementation Projects

AWS Implementation is provided for the above said Interfaces to carry out various operations in IoT Core as well to connect and communicate with AWS IoT Core using the MQTT Protocols.

AWS IoT Core Thing Management

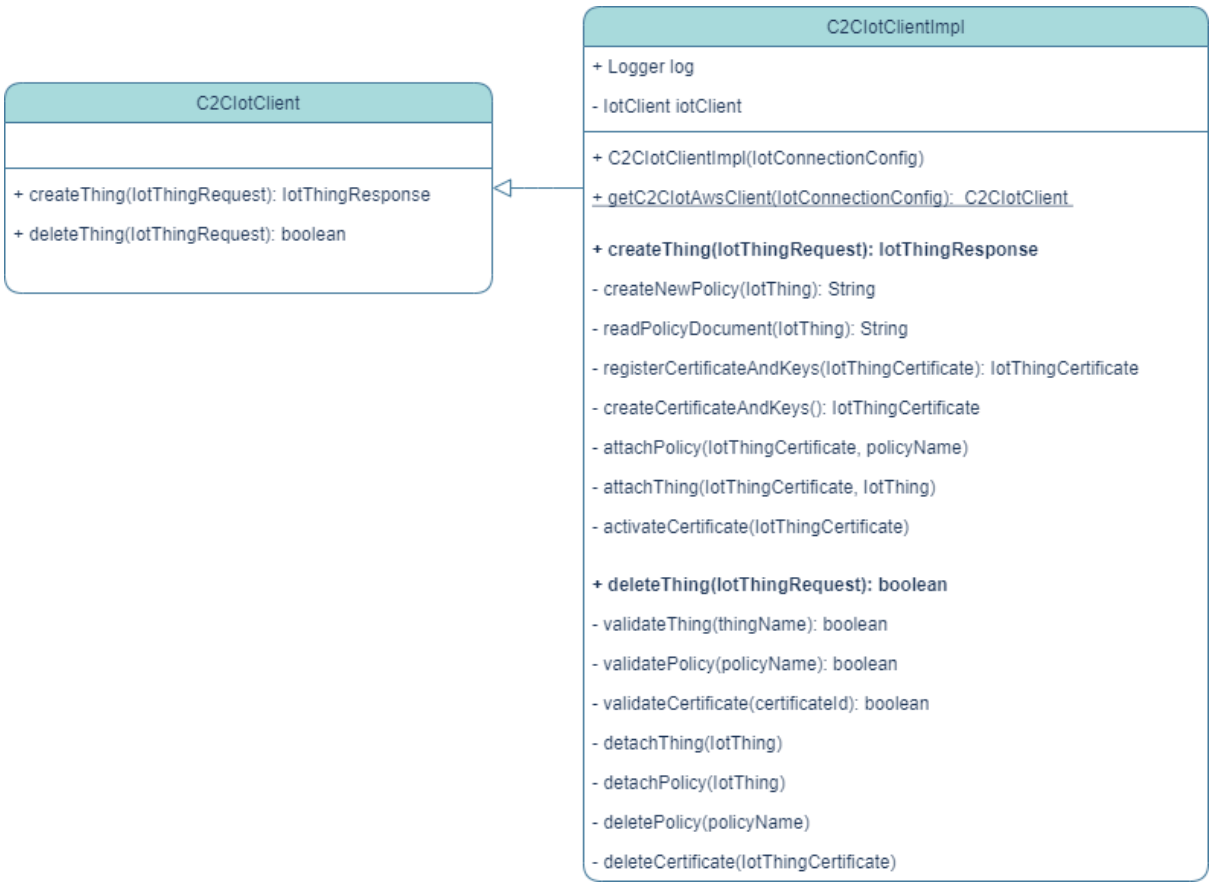
1. Create Thing
    a. Create AWS Thing with name *{system_id}*
    b. Create Policy from template
        i. **Create Policy for Production**
            1. Client Id: {*system-id*}
            2. Subscribe Topic Format: */device/{system-id}/in*
            3. Publish Topic Format: */device/{system-id}/out*
        ii. **Create Policy for Non-Production**
            1. Client Id: {*system-id*}*
            2. Subscribe Topic Format: */device/{system-id}/**
            3. Publish Topic Format: */device/{system-id}/**
    c. Register Certificate
    d. Attach Policy to Certificate
    e. Attach Thing to Certificate
    f. Activate Certificate
2. Delete Thing
    a. Detach Thing from Certificate
    b. Detach Policy from Certificate
    c. Delete Thing
    d. Delete Policy
    e. Delete Certificate

Device MQTT Connection to AWS IoT Core

1. Create MQTT Client
    a. Using Credential
    b. Using Certificate
2. Connect to IoT Core
3. Publish
    a. Publish as a Device (D2C Flow) to topic *device/{system-id}/out*
    b. Publish to Device (C2D Flow) to topic *device/{system-id}/in*
4. Subscribe to Topic *device/{system-id}/in*
5. Forward message to the Processor
6. Disconnect from IoT Core

2 Class Diagram

2.1 Aws IoT Core Thing Management



2.2 Device MQTT Connection to Aws IoT Core

3 Interface Method Details

Method Name	Purpose	Input	Output
C2CIot	Interface for IoT Gateway thing management		
createThing(C2CIotThingRequest)	Method to create a new Device and all associated resources	C2CIotThingRequest containing: 1. SystemId 2. isProd *(optional)* 3. Certificate *(optional)*	IotThingResponse
deleteThing(C2CIotThingRequest)	Method to delete a Device and all associated resources	C2CIotThingRequest containing: 1. System Id 2. Certificate	IotDeleteResponse

Method Name	Purpose	Input
C2CIotMqttTopic	Interface to perform subscribe and publish activities using an MQTT Client	
connect ()	Establish connection to the IoT Gateway	
disconnect ()	End connection with the IoT Gateway	
subscribe (Mqttconfig, C2CIotMqttProcessor)	Subscribe to a topic	MqttConfig, C2CIotMqttProcessor
publishC2D (MqttConfig, CommunicationCoreMessage)	Publish to a Device	MqttConfig, CommunicationCoreMessage
publichD2C (MqttConfig, CommunicationCoreMessage)	Publish as a Device (Simulator Use)	MqttConfig, CommunicationCoreMessage

Method Name	Purpose	Input
C2CIotTopicProcessor	Interface to process messages received on Subscription to IoT Gateway	
onMessage (CommunicationCoreMessage)	Method to process data received from IoT Gateway	CommunicationCoreMessage

4 Usage Details

4.1 Dependencies

IOT MQTT Interface

```
<dependency>
      <groupId>com.c2c.iot.mqtt</groupId>
      <artifactId>c2c_base_iot_mqtt_intf</artifactId>
      <version>0.0.3-SNAPSHOT</version>
</dependency>
```

IOT MQTT AWS implementation

```
<dependency>
	<groupId>com.c2c.iot.mqtt</groupId>
	<artifactId>c2c_base_iot_mqtt_aws_impl</artifactId>
	<version>0.0.3-SNAPSHOT</version>
</dependency>
```

IOT Interface

```
<dependency>
	<groupId>com.c2c.iot</groupId>
	<artifactId>c2c_base_iot_intf</artifactId>
	<version>0.0.4-SNAPSHOT</version>
</dependency>
```

IOT AWS implementation

```
<dependency>
	<groupId>com.c2c.iot</groupId>
	<artifactId>c2c_base_iot_aws_impl</artifactId>
	<version>0.0.4-SNAPSHOT</version>
</dependency>
```

4.2 Aws IoT Core Thing Management Sample Code

4.2.1 Create Thing Using Non-Production Policy Template

```
// CREATE CONNECTION CONFIG
IotConnectionConfig connConfig = new IotConnectionConfig();
connConfig.setRegion("<region>");

// CREATE REQUEST
IotThing iotThing = new IotThing();
iotThing.setSystemId("<system-id>");
IotThingRequest thingRequest = new IotThingRequest(iotThing);
thingRequest.setProd(false);

// CREATE CLIENT
C2CIotClient c2cIotClient = C2CIotClientImpl.getC2CIotAwsClient(connConfig);

// CALL createThing() METHOD
IotThingResponse iotThingResponse = c2cIotClient.createThing(thingRequest);
```

4.2.2 Create Thing Using Production Policy Template

```
// CREATE CONNECTION CONFIG
IotConnectionConfig connConfig = new IotConnectionConfig();
connConfig.setRegion("<region>");

// CREATE REQUEST
IotThing iotThing = new IotThing();
iotThing.setSystemId("<system-id>");
IotThingRequest thingRequest = new IotThingRequest(iotThing);
thingRequest.setProd(true);

// CREATE CLIENT
C2CIotClient c2cIotClient = C2CIotClientImpl.getC2CIotAwsClient(connConfig);

// CALL createThing() METHOD
IotThingResponse iotThingResponse = c2cIotClient.createThing(thingRequest);
```

4.2.3 Delete Thing

```
// CREATE CONNECTION CONFIG
IotConnectionConfig connConfig = new IotConnectionConfig();
connConfig.setRegion("<region>");

// CERTIFICATE
IotThingCertificate certificate = new IotThingCertificate();
certificate.setCertificateId("<certificate-id>");
certificate.setCertificateResourceName("<certificate-arn>");

// CREATE REQUEST
IotThing iotThing = new IotThing();
iotThing.setSystemId("<system-id>");
iotThing.setCertificate(certificate);
IotThingRequest thingRequest = new IotThingRequest(iotThing);

// CREATE CLIENT
C2CIotClient c2cIotClient = C2CIotClientImpl.getC2CIotAwsClient(connConfig);

// CALL deleteThing() METHOD
IotDeleteResponse deleteResponse = c2cIotClient.deleteThing(thingRequest);
```

4.3 Device MQTT Connection To Aws IoT Core Sample Code

4.3.1 Processor to process message once Subscribed

```
public class IoTMessageProcessor implements C2CIotTopicProcessor {

    @Override
    public void onMessage(C2CCommunicationCoreMessage c2cMessage) {

        //PROCESS MESSAGE
    }

}
```

4.3.2 Communication to IoT Core

```
// CREATE CONNECTION CONFIG
IotConnectionConfig connConfig = new IotConnectionConfig();
connConfig.setRegion("<region>");
connConfig.setEndPoint("<end-point>");

// CERTIFICATE
IotThingCertificate certificate = new IotThingCertificate();
certificate.setCertificateId("<certificate-id>");
certificate.setCertificateResourceName("<certificate-arn>");
certificate.setCertificatePem("<certificate-pem>");
certificate.setPrivateKey("<private-key-pem>");

// CREATE REQUEST
IotThing iotThing = new IotThing();
iotThing.setSystemId("<system-id>");
iotThing.setCertificate(certificate);

// MESSAGE
Map<String, Object> propertyBag = new HashMap<>();
propertyBag.put("<property-name>", "<property-value>");
CommunicationCoreMessage c2cMessage = new CommunicationCoreMessage();
c2cMessage.setMessageId("<message-id>");
c2cMessage.setDeviceId("<device-id>");
c2cMessage.setSourceId("source-id>");
c2cMessage.setTargetId("<target-id>");
c2cMessage.setMessageType("<message-type>");
c2cMessage.setTtl(1);
c2cMessage.setSystemId("<system-id>");
c2cMessage.setSubSystemId("<sub-system-id>");
c2cMessage.setCorrelationId("<correlation_id>");
c2cMessage.setVersion("<version>");
c2cMessage.setVin("<vin>");
c2cMessage.setEcuType("<ecu-type>");
c2cMessage.setTime(1625678536L);
c2cMessage.setPropertyBag(propertyBag);
c2cMessage.setBody("<body>");
c2cMessage.setStatus("<status>");
```

```
// CREATE CLIENT WITH CERTIFICATE
C2CIotMqttClient c2cMqttClient = C2CIotMqttClientImpl
      .getC2CIotMqttClientWithCertificate(connConfig, iotThing);

// CREATE CLIENT WITH CREDENTIAL
C2CIotMqttClient c2cMqttClient2 = C2CIotMqttClientImpl
      .getC2CIotMqttClientWithCredential(connConfig);

// CREATE MQTT CONFIGURATION
MqttConfig mqttConfig = new MqttConfig("<system-id>");

// PUBLISH/SUBSCRIBE
c2cMqttClient.connect();
c2cMqttClient.publishC2DMessage(mqttConfig, c2cMessage);
c2cMqttClient.publishD2CMessage(mqttConfig, c2cMessage);
c2cMqttClient.subscribe(mqttConfig, new IoTMessageProcessor());
c2cMqttClient.disconnect();
```

5 Git Repositories

1. IOT MQTT Interface:
   https://github.com/Github-Enterpirse-India/c2c_base_iot_mqtt_intf/tree/develop
2. IOT MQTT AWS implementation:
   https://github.com/Github-Enterpirse-India/c2c_base_iot_mqtt_aws_impl/tree/develop
3. IOT Interface:
   https://github.com/Github-Enterpirse-India/c2c_base_iot_intf/tree/develop
4. IOT AWS implementation:
   https://github.com/Github-Enterpirse-India/c2c_base_iot_aws_impl/tree/develop
5. Base Common Project:
   https://github.com/Github-Enterpirse-India/c2c_base_common/tree/develop
6. Sample Application:
   https://github.com/Github-Enterpirse-India/c2c_base_iot_sample_client/tree/develop