MQTT router for third party device adapter

implementing class: `com.arrow.selene.device.mqttrouter.MqttRouterModule;`

OS dependency: None

Requirements: None

Name	Data type	Unit of measurement	Valid values	Default value	Required parameters	Description
telemetryTopics	string	N/A	N/A	N/D	Yes	Defines comma-separate list of MQTT topics that device should subscribe telemetry topics.
deviceUidToken	integer	N/A	0-2147483647 (0x7fffffff)	1	Yes	Defines position of device UID inside MQTT topic name. Used to extract device UIDs from topic name formated as follows <token>/<token>/.../<toke
mqttUrl	string	N/A	*<URL>*	tcp://localhost:1883	Yes	Defines address of MQT server with port
mqttUserName	string	N/A	N/A	*<empty>*	No	Defines MQTT user nam
mqttPassword	string	N/A	N/A	*<empty>*	No	Defines MQTT user password
clientId	string	N/A	N/A	<mqttRouter class name>	No	Defines MQTT clientId.
mqttBrokerCertified	boolean	N/A	N/A	false	No	Defines MQTT broker certified broker.
caCertPath	string	N/A	N/A	<empty>	No(Required if mqttBrokerCertified is true)	Defines path of caCert fil that is required for mqtt certified broker connectic
clientCertPath	string	N/A	N/A	<empry>	No(Required if mqttBrokerCertified is true)	Defines path of Cert file t is required for mqtt certifi broker connection.
privateKeyPath	string	N/A	N/A	<empty>	No(Required if mqttBrokerCertified is true)	Defines path of private K file that is required for mc certified broker connectic
deviceRegistrationOverMqtt	boolean	N/A	N/A	false	No	Defines new device registration over mqtt activate or activate. DeviceRegistrationTopic properties required if it is true.
deviceRegistrationTopic	string	N/A	N/A	selene/mqtt/device/register	No(Required if deviceRegistrationOverMqtt is true)	Defines MQTT topic that need subscribe for device registration over mqtt.
deviceNameTag	string	N/A	N/A	<deviceUidTag>	No	Defines device name key that will available in regis payload It is optional, it take deviceUidTag as default.
deviceUidTag	string	N/A	N/A	<empty>	Yes(Required if deviceRegistrationOverMqtt is true and registrationTransposerScript is empty)	Defines device uid key th will available in register payload
devices	string	N/A	N/A	<empty>	No	Defines default devices
registrationTransposerScript	string	N/A	N/A	<empty>	No	Defines java transposer script path for registratior payload

telemetryTransposerScript	string	N/A	N/A	<empty>	No	Defines java transposer script path for telemetry payload
stateTransposerScript	string	N/A	N/A	<empty>	No	Defines java transposer script path for state paylc

Module Detection over MQTT:

Selene is a standalone software stack that needs to work hand-in-hand with a Third Party Application. From the architecture point of view, both these applications can be considered as two separate modules. In order to properly exchange the information, these modules needs to first detect the presence of each other.

MQTT is asynchronous in terms of connectivity. Modules do simply connect with each other instead they utilize broker for connectivity. Modules publish and subscribe to/from Broker. Thus to introduce detection capabilities in module a procedure called presence detection is provided in selene

A below specified topic and payload has been pre-configured in selene on which if a message is received, then Selene will assume the third party application is present. Whenever a device command is received by Selene, it will verify the presence of third party application. If application is not present, it will give a negative acknowledgement to Arrow Portal indicating the message cannot be forwarded to actual device.

Topic : selene/mqtt/status

Payload : {"status": "active"}

Response Topic : selene/mqtt/status/response

Response payload : {"status": "active"}

The third party application can utilize a polling mechanism wherein it will send a small json payload on Module Detection topic at regular interval. If the selene responds back on Module Detection Response topic, then application can assume that Selene is present. If the response is not received in 10 seconds, application can assume that Selene is not present in the network. Further, the third party application can utilize MQTT’s Last Will and Testament feature with the help of which when the application crashes in some worst case scenario, the MQTT broker on behalf of application will send a negative payload on Module Detection topic indicating the Selene that third party application is lost from the network.