Appendix

Device Protocol for MQTT

The MQTT version that DECADA Supports is:

- MQTT v3.1.1 on port 11883 if Agencies is using secret-based one-way authentication
- MQTT v3.1.1 over SSL/TLS port 18883 if Agencies is using certificate based two-way authentication

DECADA offers a series of APIs that devices can call via the publishing and subscribing of formatted MQTT topics, akin to that of a typical HTTPS RESTful call. The only difference here is that the request/response mechanism of a HTTPS call is implemented using publish/subscribe instead.

The following table summarizes the available APIs:

API Name	Req Topic	Reply Topic
Upstream		
Device Registration	/sys/{productKey}/{deviceKey}/thing/device/register	/sys/{productKey}/{deviceKey}/thing/device/register_reply
Add Topological Relationships of Sub-devices	/sys/{productKey}/{deviceKey}/thing/topo/add	/sys/{productKey}/{deviceKey}/thing/topo/add_reply
Delete Topological Relationships of Sub-devices	/sys/{productKey}/{deviceKey}/thing/topo/delete	/sys/{productKey}/{deviceKey}/thing/topo/delete_reply
Get Topological Relationships of Sub-devices	/sys/{productKey}/{deviceKey}/thing/topo/get	/sys/{productKey}/{deviceKey}/thing/topo/get_reply
Connect Sub-devices to EnOS Cloud	/ext/session/{productKey}/{deviceKey}/combine/login	/ext/session/{productKey}/{deviceKey}/combine/login_reply
Disconnect Sub-devices from EnOS Cloud	/ext/session/{productKey}/{deviceKey}/combine/logout	/ext/session/{productKey}/{deviceKey}/combine/logout_reply
Query Tags	/sys/{productKey}/{deviceKey}/thing/tag/query	/sys/{productKey}/{deviceKey}/thing/tag/query_reply
Report Tags	/sys/{productKey}/{deviceKey}/thing/tag/update	/sys/{productKey}/{deviceKey}/thing/tag/update_reply
Delete Tags	/sys/{productKey}/{deviceName}/thing/tag/delete	/sys/{productKey}/{deviceName}/thing/tag/delete_reply
Get Attributes	/sys/{productKey}/{deviceKey}/thing/attribute/query	/sys/{productKey}/{deviceKey}/thing/attribute/query_reply
Delete Attributes	/sys/{productKey}/{deviceKey}/thing/attribute/delete	/sys/{productKey}/{deviceKey}/thing/attribute/delete_reply
Report Attributes	/sys/{productKey}/{deviceKey}/thing/attribute/update	/sys/{productKey}/{deviceKey}/thing/attribute/update_reply
Report Device Measure Points	/sys/{productKey}/{deviceKey}/thing/measurepoint/post	/sys/{productKey}/{deviceKey}/thing/measurepoint/post_reply
Report Device Events (Non-Passthrough)	/sys/{productKey}/{deviceKey}/thing/event/{tsl.event.identifier}/post	/sys/{productKey}/{deviceKey}/thing/event/{tsl.event.identifier}/post_reply
Report Device Events (Passthrough)	/sys/{productKey}/{deviceKey}/thing/model/up_raw	/sys/{productKey}/{deviceKey}/thing/model/up_raw_reply
Downstream		
Set Device Measuring Points	/sys/{productKey}/{deviceKey}/thing/service/measurepoint/set	/sys/{productKey}/{deviceKey}/thing/service/measurepoint/set_reply
Get Device Measuring Points	/sys/{productKey}/{deviceKey}/thing/service/measurepoint/get	/sys/{productKey}/{deviceKey}/thing/service/measurepoint/get_reply

Invoke Device Services (Non-Passthrough)	/sys/{productKey}/{deviceKey}/thing/service/ {tsl.service.identifier}	/sys/{productKey}/{deviceKey}/thing/service/ {tsl.service.identifier}_reply
Invoke Device Services (Passthrough)	/sys/{productKey}/{deviceKey}/thing/model /down_raw	/sys/{productKey}/{deviceKey}/thing/model /down_raw_reply
Disable Sub-devices	/ext/session/{productKey}/{deviceKey} /combine/disable	/ext/session/{productKey}/{deviceKey} /combine/disable_reply
Enable Sub-devices	/ext/session/{productKey}/{deviceKey} /combine/enable	/ext/session/{productKey}/{deviceKey} /combine/enable_reply
Delete Sub-devices	/ext/session/{productKey}/{deviceKey} /combine/delete	/ext/session/{productKey}/{deviceKey} /combine/delete_reply

Device Protocol for CoAP

The following section describes the device protocol for CoAP connection to DECADA Cloud.

Connecting to DECADA DTLS CoAP Server

DECADA's DTLS CoAP server is hosted on GCC (For exact IP address, please contact GovTech's Technical Team) on port 5684.

Devices are expected to use a Pre-shared Key (PSK) to establish a secured DTLS session with the server. DECADA CoAP server currently supports the following PSK standards:

- TLS-PSK-WITH-AES-128-CCM-8
- TLS-PSK-WITH-AES-128-CBC-SHA256

During the handshaking process, devices are expected to provide both identity and key in the following format:

- identity: `\${ProductKey},\${DeviceKey},\${SecureMode},\${Lifetime}`

  key: `SHA-256(\${DeviceSecret})`

The following APIs are available upon successful establishment of the DTLS session:

API Name	CoAP Method	Request	Response
Send Device Data	POST	/sys/${ProductKey}/${DeviceKey} /thing/model/up_raw	/sys/${productKey}/${deviceKey}/thing /model/up_raw_reply
Retrieval of Device Data	GET	/topic/sys/${ProductKey} /${DeviceKey}/thing/model/down_raw	-

RESTful APIs via HTTPS

API Name	HTTP Method	Path	Summary
Product Service			
createProduct	POST	/connectService/products	Create a product.
getProduct	GET	/connectService/products/ {productKey}	Query product information.
listProducts	GET	/connectService/products	Query product list.
listProductTags	GET	/connectService/product/{productKey} /tags	List product tags.
updateProduct	PUT	/connectService/products/ {productKey}	Update product basic information.
updateProductTags	PUT	/connectService/product/{productKey} /tags	Update product tags.
deleteProductTags	DELETE	/connectService/product/{productKey} /tags	Delete product tags.
deleteProduct	DELETE	/connectService/products/ {productKey}	Delete a product
Device Service			
listDevices	GET	/connectService/devices	List all devices under the specified organization and product by pages.

registerDevices	POST	/connectService/products/ {productKey}/devices	Register multiple devices.
checkDevices	GET	/connectService/products/ {productKey}/devices/check	Check whether the device key is available. Classify and return device keys by Available, Existing, and Invalid groups.
getDeviceByDeviceKey	GET	/connectService/products/ {productKey}/devices/{deviceKey}	Get device information by product key and device key.
updateDeviceByDeviceKey	PUT	/connectService/products/ {productKey}/devices/{deviceKey}	Update device information by product key and device key.
deleteDeviceByDeviceKey	DELETE	/connectService/products/ {productKey}/devices/{deviceKey}	Delete a device by product key and device key.
enableDeviceByDeviceKey	PUT	/connectService/products/ {productKey}/devices/{deviceKey} /enable	Enable a device by product key and device key.
disableDeviceByDeviceKey	PUT	/connectService/products/ {productKey}/devices/{deviceKey} /disable	Disable a device by product key and device key.
listTopologyByDeviceKey	GET	/connectService/products/ {productKey}/devices/{deviceKey} /topology/subDecices	List device topology (sub-device list) by product key and device key.
listDeviceTagsByDeviceKey	GET	/connectService/products/ {productKey}/devices/{deviceKey} /tags	List device tags by product key and device key.
getMeasurepointsDataByDeviceKey	GET	/connectService/products/ {productKey}/devices/{deviceKey} /measurepoints/latestData	Get the latest data of all measurepoints of a device by product key and device key.
invokeServiceByDeviceKey	PUT	/connectService/products/ {productKey}/devices/{deviceKey} /services/{serviceId}/invoke	Call a service by product key, device key, and service ID.
setMeasurepointByDeviceKey	PUT	/connectService/products/ {productKey}/devices/{deviceKey} /measurepoints/{measurepointId}/set	Set a measurepoint by the measurepoint ID, the product Key, and the device key of device
getDeviceStatusByDeviceKeys	GET	/connectService/products/ {productKey}/devices/status	Get status of devices by product key and device keys.
createTopologyByDeviceKey	POST	/connectService/products/ {productKey}/devices/{deviceKey} /topology/subDecices	Create topology (sub-devices) for the specified device by its product key and device key.
updateDeviceTagsByDeviceKey	PUT	/connectService/products/ {productKey}/devices/{deviceKey} /tags	Update or overwrite device tags by product key and device key.
deleteDeviceTagsByDeviceKey	DELETE	/connectService/products/ {productKey}/devices/{deviceKey} /tags	Delete device tags by product key and device key.
deleteTopologyByDeviceKey	DELETE	/connectService/products/ {productKey}/devices/{deviceKey} /topology/subDecices	Delete topology (sub-devices) from the specified device by its product key and device key.
getDeviceByAssetId	GET	/connectService/devices/{assetId}	Get device information by its asset ID.
updateDeviceByAssetId	PUT	/connectService/devices/{assetId}	Update device information by its asset ID.
deleteDeviceByAssetId	DELETE	/connectService/devices/{assetId}	Delete a device by its asset ID.
enableDeviceByAssetId	PUT	/connectService/devices/{assetId} /enable	Enable a device by its asset ID.
disableDeviceByAssetId	PUT	/connectService/devices/{assetId} /disable	Disable a device by its asset ID.
listTopologyByAssetId	GET	/connectService/devices/{assetId} /topology/subDecices	List device topology (sub-device list) by its asset ID.
listDeviceTagsByAssetId	GET	/connectService/devices/{assetId} /tags	List device tags by its asset ID.

getMeasurepointsDataByAssetId	GET	/connectService/devices/{assetId} /measurepoints/latestData	Get the latest data of all measurepoints of a device by its asset ID.
invokeServiceByAssetId	PUT	/connectService/devices/{assetId} /services/{serviceId}/invoke	Call a service by the asset ID of a device and service ID.
setMeasurepointByAssetId	PUT	/connectService/devices/{assetId} /measurepoints/{measurepointId}/set	Set a measurepoint by the measurepoint ID and the asset ID of device.
getDeviceStatusByAssetIds	GET	/connectService/devices/status	Get status of devices by device asset IDs.
createTopologyByAssetId	POST	/connectService/devices/{assetId} /topology/subDecices	Create topology (sub-devices) for the specified device by its asset ID.
updateDeviceTagsByAssetId	PUT	/connectService/devices/{assetId} /tags	Update or overwrite device tags by device asset ID.
deleteDeviceTagsByAssetId	DELETE	/connectService/devices/{assetId} /tags	Delete device tags by the device asset ID.
deleteTopologyByAssetId	DELETE	/connectService/devices/{assetId} /topology/subDecices	Delete topology (sub-devices) from the specified device by its asset ID.
uploadDeviceMeasurepoints	POST	/connectService/products/ {productKey}/devices/measurepoints	Upload device measurepoints data.
Certificate Service			
applyCertificateByAssetId	POST	/connectService/devices/{assetId} /certificates/apply	Apply for a certificate and bind it with the specified device.
applyCertificateByDeviceKey	POST	/connectService/products/ {productKey}/devices/{deviceKey} /certificates/apply	Apply for a certificate and bind it with the specified device.
listCertificatesByAssetId	GET	/connectService/devices/{assetId} /certificates/list	List the certificates bound with the specified device
listCertificatesByDeviceKey	GET	/connectService/products/ {productKey}/devices/{deviceKey} /certificates/list	List the certificates bound with the specified device
revokeCertificateByAssetId	DELETE	/connectService/devices/{assetId} /certificates/revoke	Revoke certificates bound with the specified device.
revokeCertificateByDeviceKey	DELETE	/connectService/products/ {productKey}/devices/{deviceKey} /certificates/revoke	Revoke certificates bound with the specified device.
Statistics Service			
getDeviceStatistics	GET	/connectService/statistics/devices	Get device statistics of the specified organization.
Model Service			
getThingModel	GET	/modelService/thingModels/ {thingModelId}	Get thing model detailed information by thing model ID.
listThingModels	GET	/modelService/thingModels	Get the list of thing models.
Asset Service			
createAsset	POST	/assetService/assets	Create asset.
getAsset	GET	/assetService/assets/{assetId}	Get detailed asset information by its asset ID.
listAssets	GET	/assetService/assets	List assets in the specified organization.
updateAsset	PUT	/assetService/assets/{assetId}	Update asset information
deleteAsset	DELETE	/assetService/assets/{assetId}	Delete asset.
createTree	POST	/assetService/assetTrees	Create asset tree.
getTree	GET	/assetService/assetTrees/ {assetTreeId}	Get asset tree information.
listTrees	GET	/assetService/assetTrees	Get the list of asset trees.

bindAssetToNode	PUT	/assetService/assetTrees/ {assetTreeId}/nodes/{nodeId} /bindAsset	Bind asset to node.
insertNode	POST	/assetService/assetTrees/ {assetTreeId}/nodes	Insert a node in asset tree.
getNode	GET	/assetService/assetTrees/ {assetTreeId}/nodes/{nodeId}	Get node information.
listChildNodes	GET	/assetService/assetTrees/ {assetTreeId}/nodes/{nodeId} /childNodes	Get the child nodes of a node.
searchNodes	GET	/assetService/assetTrees/ {assetTreeId}/nodes	Search node information by asset ID and model ID.
searchNodePaths	GET	/assetService/assetTrees/ {assetTreeId}/nodePaths	Search for node paths in an asset tree by the model IDs of a pair of nodes (as the starting node and ending node).
updateNodeName	PUT	/assetService/assetTrees/ {assetTreeId}/nodes/{nodeId}	Update node name.
deleteAllNodesAndAsset	DELETE	/assetService/assetTrees/ {assetTreeId}	Delete related nodes by asset ID.
deleteNode	DELETE	/assetService/assetTrees/ {assetTreeId}/nodes/{nodeId}	Delete a node by the specified node ID.
Data Service			
getAssetsRawDataByTimeRange	GET	/dataService/assets/tsdb/rawData /timeRange	Get the raw data (AI, DI, and generic data) of certain measure points of specified assets in a time range.
getAssetsAIRawData	GET	/dataService/assets/tsdb/ai/rawData /measurepoints	Get the AI raw data of certain measure points of specified assets in a time range.
getAssetsAINormalizedData	GET	/dataService/assets/tsdb/ai /normalizedData/measurepoints	Get the minute-level normalized AI data of certain measure points of specified assets in a time range.
getAssetsStatusData	GET	/dataService/assets/tsdb/di /measurepoints	Get the status data (DI) of specified assets in a time range.
getAssetsGenericData	GET	/dataService/assets/tsdb/generic /measurepoints	Get the generic data of certain measure points of specified assets in a time range.
getAssetsLatestData	GET	/dataService/assets/latest /measurepoints	Get the latest data of all measure points of the specified assets.
getPointsTSDBMetaData	GET	/dataService/metaData/storage/tsdb	Get the TSDB storage metadata of the specified model
Alert Service			
listActiveAlerts	GET	/eventService/alerts/activeAlerts/list	List active alerts by organization ID and other filtering conditions (applicable for querying less than 10,000 records).
listHistoryAlerts	GET	/eventService/alerts/historyAlerts/list	List history alerts by organization ID and other filtering conditions (applicable for querying less than 10,000 records).
scrollActiveAlerts	GET	/eventService/alerts/activeAlerts/scroll	List active alerts by organization ID and other filtering conditions (applicable for querying more than 10,000 records).
scrollHistoryAlerts	GET	/eventService/alerts/historyAlerts /scroll	List history alerts by organization ID and other filtering conditions (applicable for querying more than 10,000 records).
insertActiveAlerts	POST	/eventService/alerts/activeAlerts	Insert active alerts.
insertHistoryAlerts	POST	/eventService/alerts/historyAlerts	Insert history alerts.

updateActiveAlertTags	PUT	/eventService/alerts/activeAlerts/tags	Update tags of active alerts (applicable for inserted alerts only).
updateHistoryAlertTags	PUT	/eventService/alerts/historyAlerts/tags	Update tags of history alerts (applicable for inserted alerts only).
closeActiveAlerts	PUT	/eventService/alerts/activeAlerts	Close active alerts. Closed alerts will be saved as history alerts data.
deleteActiveAlerts	DELETE	/eventService/alerts/activeAlerts	Delete active alerts.
deleteHistoryAlerts	DELETE	/eventService/alerts/historyAlerts	Delete history alerts (applicable to inserted alerts only).
Alert Rule Service			
getAlertSeverity	GET	/eventService/rules/alertSeverities	Get alert severity by organization ID and severity ID.
getAlertType	GET	/eventService/rules/alertTypes	Get alert type by organization ID and type ID.
getAlertContent	GET	/eventService/rules/alertContents	Get alert content by organization ID and content ID.
getAlertRule	GET	/eventService/rules/alertRules	Get alert rule by organization ID and rule ID.
listAlertSeverities	GET	/eventService/rules/alertSeverities	List alert severities by organization ID.
listAlertTypes	GET	/eventService/rules/alertTypes	List alert types by organization ID.
listAlertContents	GET	/eventService/rules/alertContents	List alert contents by organization ID and model ID.
listAlertRules	GET	/eventService/rules/alertRules	List alert rules by organization ID and other filtering conditions.
File Management Service			
createFiles	POST	/commonFileService/files	Upload multiple files to a single directory.
getFile	GET	/commonFileService/files	Download a file.
deleteFile	DELETE	/commonFileService/files	Delete a file.
ifFileExist	GET	/commonFileService/files/ifExist	Check if a file already exists.
getFileInfo	GET	/commonFileService/files/info	Get file information
Multipart File Management Servicce			
initiateMultipartUpload	POST	/commonFileService/file/parts/init	Initiate uploading a file by multiple parts.
uploadPart	POST	/commonFileService/file/parts	Upload a file part.
completeMultipartUpload	POST	/commonFileService/file/parts/complete	Complete uploading a file by multiple parts.
cancelMultipartUpload	POST	/commonFileService/file/parts/cancel	Cancel uploading a file by multiple parts.
listParts	GET	/commonFileService/file/parts/list	Get information about all file parts.
getPart	GET	/commonFileService/file/parts	Download a file part