

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
1	JSON	<pre> st: /plant1/# tt: /plant1/+/+ tts: /plant1/line1/device1_measure1_Type sub: 1.[* _TOPIC_LEVEL_[0]&"_&_TOPIC_LEVEL_[1]&"_&\$substringBefore(_ TOPIC_LEVEL_[2],"_") -> source.id] 2.[\$substringAfter(_TOPIC_LEVEL_[2],"_") -> type] 3.[\$now() -> time] 4.[value -> measure1_Type.V.value] </pre>	M	<pre> { "mea": [{ "tid": "uuid_01", "psid": "Crest", "devicePath": "path01_80_X03_VVB001StatusB_Crest", "values": [{ "value": 4.6, "timestamp": 1648562285347 }] }, { "tid": "uuid_02", "psid": "Crest", "devicePath": "path01_80_X03_VVB001StatusB_Crest", "values": [{ "value": 5.6, "timestamp": 1648563285347 }] }], "_TOPIC_LEVEL_": { "devices", "device_best_01" } } </pre>	<pre> { "measure1_Type": { "v": { "value": 110, "unit": "C" } }, "time": "2022-08-05T00:14:49.389+02:00", "source": { "id": "909090" }, "type": "c8y_TemperatureMeasurement" } </pre>	For the device with external id: plant1_line1_device1 a measurement c8y_TemperatureMeasurement should be created. The device is created implicitly.
2	JSON	<pre> st: devices/# tt: devices/+ tts: devices/device_best_01 sub: 1.[* _TOPIC_LEVEL_[1] -> source.id] 2.[mea.values[0].value -> c8y_ProcessLoadMeasurement.L.value] 3.[\$map(\$map(mea.values[0].timestamp, \$number), function(\$v, \$i, \$a) { \$fromMillis(\$v) }) -> time] </pre>	M	<pre> { "mea": [{ "tid": "uuid_01", "psid": "Crest", "devicePath": "path01_80_X03_VVB001StatusB_Crest", "values": [{ "value": 4.6, "timestamp": 1648562285347 }] }, { "tid": "uuid_02", "psid": "Crest", "devicePath": "path01_80_X03_VVB001StatusB_Crest", "values": [{ "value": 5.6, "timestamp": 1648563285347 }] }], "_TOPIC_LEVEL_": { "devices", "device_best_01" } } </pre>	<pre> { "c8y_ProcessLoadMeasurement": { "L": { "value": 110, "unit": "%" } }, "time": "2022-08-05T00:14:49.389+02:00", "source": { "id": "909090" }, "type": "c8y_ProcessLoadMeasurement" } </pre>	For the device with external id: device_best_01 multiple measurements should be created. The device is created implicitly.

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
3	JSON	st: device/# tt: device/express/+, tts: device/express/berlin_01 sub: 1.[* _TOPIC_LEVEL_[2]->DEVICE_IDENT_] 2.[customType->type] 3.[operator&"-&line->name] 4.[capacity->capacity]	I	{ "line": "Bus-Berlin-Rom", "operator": "EuroBus", "customFragment": { "customFragmentValue": "Express" }, "capacity": 64, "customArray": { "ArrayValue1", "ArrayValue2" }, "customType": "type_International" }	{ "c8y_IsDevice": {}, "name": "Bus Name", "type": "type_bus", "capacity": 100, "_DEVICE_IDENT_": "909090" } 	Create device with: 1.external id: berlin_01 2.name: EuroBus-Bus-Berlin-Rom 3.type: type_International
4	JSON	st: event/# tt: event/+ tts: event/berlin_01 sub: 1.[* _TOPIC_LEVEL_[1] -> source.id] 2.[txt->text] 3.[msg_type->type] 4.[\$now()->time]	E	{ "msg_type": "c8y_BusStopEvent", "txt": "Bus stopped at petrol station today!", "td": "2022-09-08T16:21:53.389+02:00", "ts": "1665473038000" } 	{ "source": { "id": "909090" }, "text": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_GeneralBusEvent" } 	Event for existing device should be created mention [\$fromMillis(\$number(deviceTimestamp))>time]
5	JSON	st: measurement/# tt: measurement/+gazoline tts: measurement/berlin_01/gazoline sub: 1.[* _TOPIC_LEVEL_[1] -> source.id] 2.[fuel->c8y_FuelMeasurement.F.value] 3.[\$now()->time]	M	{ "fuel": 65, "ts": "2022-08-05T00:14:49.389+02:00", "mea": "c8y_FuelMeasurement" } 	{ "c8y_FuelMeasurement": { "L": { "value": 110, "unit": "l" }, }, "time": "2022-10-18T00:14:49.389+02:00", "source": { "id": "909090" }, "type": "c8y_FuelMeasurement" } 	Add c8y_FuelMeasurement to bus.
6	JSON	st: multiarray/devices tt: multiarray/devices tts: multiarray/devices sub: 1.[* device->_DEVICE_IDENT_] , choose option "Expand Array" 2.[types.type_A->type] 3.[\$map(used_name, function(\$v, \$i, \$a) { \$contains(\$v,'dl') ? \$join(['Special_i0', \$string(\$i))] : \$join([\$string(\$v), \$string(\$i)]) })->name] , choose option "Expand Array"	I	{ "device": ["d1_id", "d2_id"], "types": { "type_A": "type_A", "type_B": "type_B" }, "used_name": ["Pressure_d1", "Pressure_d2"] } 	{ "c8y_IsDevice": {}, "name": "Vibration Sensor", "type": "maker_Vibration_Sensor" } 	New Devices: 1.Pressure_d21 2.Special_i00 should be created. All device have the same type "type_A"

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
7	JSON	st: arrayType/devices tt: arrayType/devices tts: arrayType/devices sub: 1. [\$substringBefore(\${0}.devicePath,"_AL") -> source.id] 2. [\${}.values[0].value -> c8y_TemperatureMeasurement.T.value] , choose option "Expand Array" 3. [\$map(\$map(\${}.values[0].timestamp, \$number), function(\$v) { \$fromMillis(\$v)}) -> time] , choose option "Expand Array"	M	<pre>{ { "tid": "5e4bac9f-b47a-499e-8601-68fc16a9847c", "psid": "Crest", "devicePath": "c2818e07-4c09-42f0-ba24-ddb712573ab5_AL1352_192168221_80_X03_VVB001StatusB_Crest", "processDataUnit": "20", "values": [{ "value": 4.6, "timestamp": 1648562285347 }] }, { "tid": "5e4bac9f-b47a-499e-8601-68fc16a9847c", "psid": "Crest", "devicePath": "c2818e07-4c09-42f0-ba24-ddb712573ab5_AL1352_192168221_80_X03_VVB001StatusB_Crest", "processDataUnit": "20", "values": [{ "value": 5.6, "timestamp": 1648562285347 }] } }</pre>	<pre>{ "c8y_TemperatureMeasurement": { "T": { "value": 110, "unit": "C" } }, "time": "2022-08-05T00:14:49.389+02:00", "source": { "id": "909090" }, "type": "c8y_TemperatureMeasurement" }</pre>	Create one device with the name: device_c8y_Serial_c2818e07-4c09-42f0-ba24-ddb712573ab5 and for this device create two measurements of type: "c8y_TemperatureMeasurement"
8	JSON	st: eventObject/# tt: eventObject/+ tts: eventObject/berlin_01 sub: 1. [_TOPIC_LEVEL_[1] -> source.id] 2. [txt -> text] 3. [msg_type -> type] 4. [\$now() -> time] 5. [model -> customProperties] , Choose Repair Strategy: REMOVE_IF_MISSING	E	<pre>{ "msg_type": "c8y_BusStopEvent", "txt": "Bus stopped at petrol station today!", "td": "2022-09-08T16:21:53.389+02:00", "model": { "name": "MAN e-Bus" }, "_TOPIC_LEVEL_": ["eventObject", "berlin_01"] }</pre>	<pre>{ "source": { "id": "909090" }, "text": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_TestEvent", "customProperties": "dummy" }</pre>	Create event for device
9	JSON	st: measurementObject/# tt: measurementObject/+ /gasoline tts: measurementObject/berlin_01/gasoline sub: 1.[* _TOPIC_LEVEL_[1] -> source.id] 2.[mea -> type] 3.[\$now() -> time] 4.[fuel*3.78541 -> c8y_FuelMeasurement.Tank.value] 5.[oil?({"Motor\": {"value\":"oil", \"unit\":"l\"}}):null] -> c8y_OilMeasurement] Choose Repair Strategy: REMOVE_IF_MISSING	M	<pre>{ "fuel": 65, "oil": 4.5, "ts": "2022-08-05T00:14:49.389+02:00", "mea": "c8y_FuelMeasurement" }</pre>	<pre>{ "c8y_FuelMeasurement": { "Tank": { "value": 110, "unit": "l" } }, "c8y_OilMeasurement": "undefined", "time": "2022-08-05T00:14:49.389+02:00", "source": { "id": "909090" }, "type": "c8y_FuelMeasurement" }</pre>	This mapping makes use of the option "REMOVE_IF_MISSING". The incoming payload can contain either properties: "fuel", "oil" or both. Depending on this the relevant fragments in the Cumulocity measurement are created.
13	GENERIC_BINARY	st: binary/+ tt: binary/+ tts: binary/berlin_01 sub: 1.[* _TOPIC_LEVEL_[1] -> deviceId] 2.[\$join([text," ", \$now()]) -> description]	O	Hex Code: 5a75207370c3a47420303821	<pre>{ "source": { "id": "909090" }, "text": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_TestEvent" }</pre>	Send c8y_TestEvent to device with externalId berlin_01

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
14	JSON	st: operation/# tt: operation/+ tts: operation/berlin_01 sub: 1.[* _TOPIC_LEVEL_[1] -> deviceId] 2.[\$join([text,"_",\$now())) -> description]	O	{ "text": "Special operation restart" } 	{ "deviceId": "909090", "description": "New camera operation!", "type": "maintenance_operation" } 	Create operation "maintenance_operation" for device with externalId berlin_01
17	JSON	st: device/update/+ tt: device/update/+ tts: device/update/berlin_01 sub: 1.[* _TOPIC_LEVEL_[2] -> _DEVICE_IDENT_] 2.[customType->type]	I	{ "customType": "type_Overnight" } 	{ "type": "type_any" } 	Update type of existing device.
18	PROTOBUF_STATIC	st: protobuf/measurement tt: protobuf/measurement tts: protobuf/measurement sub: Defined in cumulocity-dynamic-mqtt-mapper/mqtt-mapping-service/src/main/java/mqtt/mapping/processor/processor/fixed/StaticProtobufProcessor.java	M	Send message in protobuf format: option java_package = "mqtt.mapping.processor.protobuf"; option java_outer_classname = "MeasurementProto"; message CustomMeasurement { int64 timestamp = 1; float value = 2; string unit = 3; string externalIdType = 4; string externalId = 5; string measurementType = 6; } Use test client: cumulocity-dynamic-mqtt-mapper/mqtt-mapping-service/src/test/java/mqtt/mapping/ProtobufPahoClient.java	{ "c8y_GenericMeasurement": { "Module": { "value": 110, "unit": "1" } }, "time": "2022-08-05T00:14:49.389+02:00", "source": { "id": "909090" }, "type": "c8y_GenericMeasurement_type" } 	Use test client: cumulocity-dynamic-mqtt-mapper/mqtt-mapping-service/src/test/java/mqtt/mapping/ProtobufPahoClient.java to create a new measurement for bus "berlin_01"
19	PROCESSOR_EXTENSION	st: protobuf/event tt: protobuf/event tts: protobuf/event sub: Defined in cumulocity-dynamic-mqtt-mapper/mqtt-mapping-extension/src/main/java/mqtt/mapping/processor/extension/external/ProcessorExtensionCustomEvent.java	E	Send message in protobuf format: syntax = "proto3"; package processor.protobuf; option java_package = "mqtt.mapping.processor.extension.external"; option java_outer_classname = "CustomEventOuter"; message CustomEvent { int64 timestamp = 1; string txt = 2; string unit = 3; string externalIdType = 4; string externalId = 5; string eventType = 6; } Use test client: cumulocity-dynamic-mqtt-mapper/mqtt-mapping-extension/src/test/java/mqtt/mapping/ProtobufPahoClient.java	{ "c8y_GenericMeasurement": { "Module": { "value": 110, "unit": "1" } }, "time": "2022-08-05T00:14:49.389+02:00", "source": { "id": "909090" }, "type": "c8y_GenericMeasurement_type" } 	Use test client: cumulocity-dynamic-mqtt-mapper/mqtt-mapping-extension/src/test/java/mqtt/mapping/ProtobufPahoClient.java to create a new event for bus "berlin_01"

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
20	JSON	st: panel tt: panel tts: panel sub: 1.[* deviceId->source.id] 2.[\$fromMillis(\$number(deviceTimestamp))->time] 3.[temperature->c8y_TemperatureMeasurement.T.value]	I.M	{ "deviceId": "863859042393327", "version": "1", "deviceType": "20", "deviceTimestamp": "1665473038000", "deviceStatus": "BTR", "temperature": 90 }	{ "c8y_TemperatureMeasurement": { "T": { "value": 110, "unit": "C" } }, "time": "2022-08-05T00:14:49.389+02:00", "source": { "id": "909090" }, "type": "c8y_TemperatureMeasurement" }	Devices with external id: 863859042393327 does not exist and is implicitly created. For this device an new measurement is created.
21	JSON	st: panel tt: panel tts: panel sub: 1.[deviceId->source.id] 2.[[\$now()->time]] 3.['New device status: ' & deviceStatus & '!'->text]	E	{ "deviceId": "863859042393327", "version": "1", "deviceType": "20", "deviceTimestamp": "1665473038000", "deviceStatus": "BTR", "temperature": 90 }	{ "source": { "id": "909090" }, "text": "New device status: BTR!", "time": "2022-11-24T00:14:49.389+02:00", "type": "c8y_GeneralPanelEvent" }	For this device an new event is created.
23	JSON	st: flexMeasurement/# tt: flexMeasurement/+gazoline tts: flexMeasurement/berlin_01/gazoline sub: 1.[_TOPIC_LEVEL_[1] -> source.id] 2.[Measurementname & "_type" -> type] 3.[Measurementname = "Airsensor" ? {Seriesname:{ "value": value, "unit": unit}} : null -> Airsensor] / RepairStrategy: REMOVE_IF_NULL 4.[Measurementname = "Liquidsensor" ? {Seriesname:{ "value": value, "unit": unit}} : null -> Liquidsensor] /RepairStrategy: REMOVE_IF_NULL 5. [\$now() -> time]	M	{ "Measurementname": "Airsensor", "Seriesname": "Humidity", "value": 10, "unit": "%" }	{ "Airsensor": "dummy", "Liquidsensor": "dummy", "time": "2022-08-05T00:14:49.389+02:00", "source": { "id": "909090" }, "type": "c8y_measurementtype" }	Depending on the content in the payload: 1. is "Airsensor" present 2. is "Liquidsensor" present either mapping 3. or 4. is evaluated and the relevant fragment in the measurement is crreated.
24	JSON	st: alarm/tires tt: alarm/tires tts: alarm/tires sub: 1.[bus_id->source.id] 2.[msg_type->type] 3.[tx->text]	A	{ "msg_type": "c8y_FlatTireAlarm", "tx": "Left rear tire loses air!", "bus_id": "berlin_01" }	{ "source": { "id": "909090" }, "type": "c8y_FlatTireAlarm", "text": "Left rear tire loses air!", "severity": "MAJOR", "status": "ACTIVE", "time": "2022-03-19T12:03:27.845Z" }	An alarm should be created for the device berlin_01.
25	PROCESSOR_EXTENSION	st: measurementExt tt: measurementExt tts: measurementExt sub: Events for mqtt-mapping-extension: CustomMeasurement Extension for PROCESSOR_EXTENSION: mqtt-mapping-externsion Defined in cumulocity-dynamic-mqtt-mapper/mqtt-mapping-extension/src/main/java/mqtt/mapping/processor/extension/external/ProcessorExtensionCustomMeasurement.java	M	{ "temperature": 120.5, "unit": "Celsius", "time": "2023-07-12T16:21:53.389+02:00", "externalId": "berlin_01", "unexpected": 17.5 }	{ "source": { "id": "909090" }, "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_Temperature", "c8y_Temperature": "dummy", "c8y_Fragment_to_remove": "remove_me" }	A measasement should be created for the device berlin_01. The fragment "c8y_Fragment_to_remove" is not included in the created measurement, as the repair strategy is "REMOVE_IF_NULL". In addition the reapar strategy "CREATE_IF_MISSING" is used. This is required to map the node "unexpected" to the target fragment "c8y_Unexpected". This is created, due to the used reapir strategy.