

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
01	JSON	<pre> mappingTopic: /plant1/+/+ mappingTopicSample: /plant1/line1/device1_measure1_Type check: Create non existing device sub: 1.[* _TOPIC_LEVEL_[0]&" "& _TOPIC_LEVEL_[1]&" "&\$\$substringBefore(_TOPIC_LEVEL_[2],"_" -> _IDENTITY_.externalId] 2.[\$substringAfter(_TOPIC_LEVEL_[2],"_" -> type] 3.[\$now() -> time] 4.[value -> measure1_Type.V.value] </pre>	M	<pre> { "value": 100 } </pre>	<pre> { "measure1_Type": { "V": { "value": 110, "unit": "C" } }, "time": "2022-08-05T00:14:49.389+02:00", "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.
02	JSON	<pre> mappingTopic: devices/+ mappingTopicSample: devices/device_best_01 check: Create non existing device sub: 1.[* _TOPIC_LEVEL_[1] -> _IDENTITY_.externalId] 2.[mea.values[0].value -> c8y_ProcessLoadMeasurement.L.value] 3.[\$map(\$map(mea.values[0].timestamp, \$number), function(\$v, \$i, \$a) { \$fromMillis(\$v) }) -> time] check expand to array for this substitution </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 }] }] } </pre>	<pre> { "c8y_ProcessLoadMeasurement": { "L": { "value": 110, "unit": "%" } }, "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_ProcessLoadMeasurement" } </pre>	For the device with external id: device_best_01 multiple measurements should be created. The device is created implicitly.
03	JSON	<pre> mappingTopic: device/express/+ mappingTopicSample: device/express/berlin_01 check: Use external id sub: 1.[* _TOPIC_LEVEL_[2] -> _IDENTITY_.externalId] 2.[customType -> type] 3.[operator&"-"&line -> name] 4.[capacity -> capacity] </pre>	I	<pre> { "line": "Bus-Berlin-Rom", "operator": "EuroBus", "customFragment": { "customFragmentValue": "Express" }, "capacity": 64, "customArray": ["ArrayValue1", "ArrayValue2"], "customType": "type_International" } </pre>	<pre> { "c8y_IsDevice": {}, "com_cumulocity_model_Agent": {}, "name": "Bus Name", "type": "type_bus", "capacity": 100, "time": "2022-08-05T00:14:49.389+02:00", } </pre>	Create device with: 1.External id: berlin_01 2.Name: EuroBus-Bus-Berlin-Rom 3.Type: type_International
04	JSON	<pre> mappingTopic: event/+ mappingTopicSample: event/berlin_01 check: Use external id sub: 1.[* _TOPIC_LEVEL_[1] -> _IDENTITY_.externalId] 2.[txt->text] 3.[msg_type -> type] 4.[\$now() -> time] </pre>	E	<pre> { "msg_type": "c8y_BusStopEvent", "txt": "Bus stopped at petrol station today!", "td": "2022-09-08T16:21:53.389+02:00", "ts": "1665473038000" } </pre>	<pre> { "text": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_GeneralBusEvent" } </pre>	Event for existing device should be created mention [\$fromMillis(\$number(device Timestamp))>time]
05	JSON	<pre> mappingTopic: measurement/+/gazoline mappingTopicSample: measurement/berlin_01/gazoline check: Use external id sub: 1.[* _TOPIC_LEVEL_[1] -> _IDENTITY_.externalId] 2.[fuel -> c8y_FuelMeasurement.F.value] 3.[\$now() -> time] </pre>	M	<pre> { "fuel": 65, "ts": "2022-08-05T00:14:49.389+02:00", "mea": "c8y_FuelMeasurement" } </pre>	<pre> { "c8y_FuelMeasurement": { "L": { "value": 110, "unit": "L" } }, "time": "2022-10-18T00:14:49.389+02:00", "type": "c8y_FuelMeasurement" } </pre>	Add c8y_FuelMeasurement to bus.

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
06	JSON	mappingTopic: multiarray/devices mappingTopicSample: multiarray/devices check: Use external id sub: 1.[* device -> _IDENTITY_.externalId] , 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"
07	JSON	mappingTopic: arrayType/devices mappingTopicSample: arrayType/devices check: Create non existing device check: Use external id sub: 1. [\$substringBefore(\$[0].devicePath,"_AL") -> _IDENTITY_.externalId] 2. [\${}.values[0].value -> c8y_TemperatureMeasurement.T.value] , choose option "Expand Array" 3. [\$map(\$map(\$[0].values[0].timestamp, \$number), function(\$v) { \$fromMillis(\$v)}) -> time] , choose option "Expand Array"	M	[{ "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 }] }]	{ "c8y_TemperatureMeasurement": { "T": { "value": 110, "unit": "C" } }, "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_TemperatureMeasurement" }	Create one device implicitly with the name: device_c8y_Serial_c2818e07- 4c09-42f0-ba24- ddb712573ab5 and for this device create two measurements of type: "c8y_TemperatureMeasuremen t"
08	JSON	mappingTopic: eventObject/+ mappingTopicSample: eventObject/berlin_01 check: Use external id sub: 1. [_TOPIC_LEVEL_[1] -> _IDENTITY_.externalId] 2. [txt -> text] 3. [msg_type -> type] 4. [\$now() -> time] 5. [model -> customProperties] choose Repair Strategy: REMOVE_IF_MISSING_OR_NULL	E	{ "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" } }	{ "text": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_TestEvent", "customProperties": "dummy" }	Create event for device. If the source payload contains the fragment model it is mapped to customProperties. If it does not contain th e fragment the customProperties is removed from the target payload.

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
09	JSON	mappingTopic: measurementObject/+gazoline mappingTopicSample: measurementObject/berlin_01/gazoline check: Use external id sub: 1.[[* _TOPIC_LEVEL_[1] -> _IDENTITY_.externalId] 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_OR_NULL	M	{ "fuel": 65, "oil": 4.5, "ts": "2022-08-05T00:14:49.389+02:00", "mea": "c8y_FuelMeasurement" }	{ "c8y_FuelMeasurement": { "Tank": { "value": 110, "unit": "l" } }, "c8y_OilMeasurement": "undefined", "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_FuelMeasurement" }	This mapping makes use of the option "REMOVE_IF_MISSING_OR_NULL". The incoming payload can contain either properties: "fuel", "oil" or both. Depending on this the relevant fragments in the Cumulocity measurement are created.
10	GENERIC_BINARY	mappingTopic: binary/+ mappingTopicSample: binary/berlin_01 check: Use external id sub: 1.[[* _TOPIC_LEVEL_[1] -> _IDENTITY_.externalId] 2.[[\$number(\$substring(message,0,6)) -> value]	E	Hex Code: 0x41b1 No leading 0x, only the plain payload as hexadecimal numbers	{ "text": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_TestEvent", "value": 99 }	Snoop recorded message
11	JSON	mappingTopic: operation/+ mappingTopicSample: operation/berlin_01 check: Use external id sub: 1.[[* _TOPIC_LEVEL_[1] -> deviceId] 2.[[\$join([text,"_", \$now()]) -> description]	O	{ "text": "Special operation restart" }	{ "description": "New camera operation!", "type": "maintenance_operation" }	Create operation "maintenance_operation" for device with externalId berlin_01
12	GENERIC_BINARY	mappingTopic: binaryEvent/+ mappingTopicSample: binaryEvent/berlin_01 check: Use external id sub: 1. ["Temp: "&\$number(\$substring(message,0,4))&" C" -> text] 2. [* _TOPIC_LEVEL_[1] -> _IDENTITY_.externalId] 3. [\$now() -> time]	E	Hex Code: 0x5a75 No leading 0x, only the plain payload as hexadecimal numbers	{ "text": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_TestEvent" }	Send c8y_TestEvent to device with externalId berlin_01
13	JSON	mappingTopic: device/update/+ mappingTopicSample: device/update/berlin_01 check: Use external id sub: 1.[[* _TOPIC_LEVEL_[2] -> _IDENTITY_.externalId] 2.[[customType->type]	I	{ "customType": "type_Overnight" }	{ "type": "type_any" }	Update type of existing device.
14	PROTOBUF_STATIC	mappingTopic: protobuf/measurement mappingTopicSample: protobuf/measurement check: Use external id sub: Defined in cumulocity-dynamic-mapper/dynamic-mapping-service/src/main/java/dynamic/mapping/processor/processor/Fixed/StaticProtobufProcessor.java	M	Send message in protobuf format: <pre>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; }</pre> Use test client: cumulocity-dynamic-mapper/dynamic-mapping-service/src/test/java/dynamic/mapping/ProtobufMqttClient.java	{ "c8y_GenericMeasurement": { "Module": { "value": 110, "unit": "l" } }, "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_GenericMeasurement_type" }	Use test client: cumulocity-dynamic-mapper/dynamic-mapping-service/src/test/java/dynamic/mapping/ProtobufMqttClient.java to create a new measurement for bus "berlin_01"

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
15	EXTENSION_SOURCE	mappingTopic: protobuf/event mappingTopicSample: protobuf/event check: Use external id sub: Defined in cumulocity-dynamic-mapper/dynamic-mapping-extension/src/main/java/dynamic/mapping/processor/extension/external/ProcessorExtensionCustomEvent.java In selection: Extensions for PROCESSOR_EXTENSION choose: dynamic-mapping-extension In selection: Events for dynamic-mapping-extension choose: CustomEvent	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-mapper/dynamic-mapping-extension/src/test/java/dynamic/mapping/ProtobufMqttClient.java	{ "text": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "type": "c8y_TestEvent" }	Use test client: cumulocity-dynamic-mapper/dynamic-mapping-extension/src/test/java/dynamic/mapping/ProtobufMqttClient.java to create a new event for bus "berlin_01"
16	JSON	mappingTopic: panel mappingTopicSample: panel check: Create non existing device check: Use external id sub: 1.[* deviceId -> _IDENTITY_.externalId] 2.[\$fromMillis(\$number(deviceTimestamp)) -> time] 3.[temperature -> c8y_TemperatureMeasurement.T.value]	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", "type": "c8y_TemperatureMeasurement" }	Devices with external id: 863859042393327 does not exist and is implicitly created. For this device an new measurement is created.
17	JSON	mappingTopic: panel mappingTopicSample: panel check: Use external id sub: 1.[deviceId -> _IDENTITY_.externalId] 2.[[\$now() -> time]] 3.['New device status: ' & deviceStatus & '!' -> text]	E	{ "deviceId": "863859042393327", "version": "1", "deviceType": "20", "deviceTimestamp": "1665473038000", "deviceStatus": "BTR", "temperature": 90 }	{ "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.
18	JSON	mappingTopic: flexM/+gazoline mappingTopicSample: flexM/berlin_01/gazoline check: Use external id sub: 1. [* _TOPIC_LEVEL_[1] > _IDENTITY_.externalId] 2. [Measurementname & "type" -> type] 3. [Measurementname = "Airsensor" ? (Seriesname:{"value": value, "unit": unit}) : null -> Airsensor]] select: Repair Strategy: REMOVE_IF_NULL_OR_MISSING 4. [Measurementname = "Liquidsensor" ? (Seriesname:{"value": value, "unit": unit}) : null -> Liquidsensor] select: Repair Strategy: REMOVE_IF_NULL_OR_MISSING 5. [\$now() -> time]	M	{ "Measurementname": "Airsensor", "Seriesname": "Humidity", "value": 10, "unit": "%" }	{ "Airsensor": "dummy", "Liquidsensor": "dummy", "time": "2022-08-05T00:14:49.389+02:00", "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 created.
19	JSON	mappingTopic: alarm/tires mappingTopicSample: alarm/tires check: Use external id sub: 1.[bus_id -> _IDENTITY_.externalId] 2.[msg_type -> type] 3.[tx -> text]	A	{ "msg_type": "c8y_FlatTireAlarm", "tx": "Left rear tire loses air!", "bus_id": "berlin_01" }	{ "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.

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
20	EXTENSION_SOURCE	mappingTopic: measurementExt mappingTopicSample: measurementExt mappingType: Processor Extension Source check: Use external id sub: Extension for PROCESSOR_EXTENSION_SOURCE: dynamic-mapping-extension Events for dynamic-mapping-extension: CustomMeasurement Defined in cumulocity-dynamic-mapper/dynamic-mapping-extension/src/main/java/dynamic/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 }	{ "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 reapiar 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 reapiar strategy.
21	JSON	mappingTopic: v2/things/+ mappingTopicSample: v2/things/berlin_01 check: Use external id 1. [* _TOPIC_LEVEL [2] -> _IDENTITY_.externalId] 2. [\$now() -> time] 3. [values(key: {'Measurement':{'value':value, 'key':'U'}}) -> \$]	M	{ "values":[{ "key":"velocidad_cabezal", "value":136.34 }, { "key":"temperature", "value":25 }] }	{ "time":"2022-08-05T00:14:49.389+02:00", "type":"c8y_FlexibleMeasurement" }	A measurement with two fragments: 1. velocidad_cabezal 2. temperature is created. It demonstrates the use of a substitution using "\$" as a target. This results in merging the extracted content with the predefined target template.
22	JSON	mappingTopic: v3/things/+ mappingTopicSample: v3/things/berlin_01 check: Use external id 1. [* _TOPIC_LEVEL [2] -> _IDENTITY_.externalId] 2. [\$now() -> time] 3. [\$map(values, function (\$v) { {\$v.key: {'Measurement':{'value':\$v.value, 'unit':'U'}}}}) -> \$ }] select: Expand as array	M	{ "values":[{ "key":"velocidad_cabezal", "value":136.34 }, { "key":"temperature", "value":25 }] }	{ "time":"2022-08-05T00:14:49.389+02:00", "type":"c8y_FlexibleMeasurement" }	Two measurements with different fragments: 1. velocidad_cabezal 2. temperature are created. It demonstrates the use of a substitution using "\$" as a target. This results in merging the extracted content with the predefined target template in combination with the attribute "expand2Array". See as well mapping 21.
23	JSON	mappingTopic: datalogger/0018 mappingTopicSample: datalogger/0018 check: Use external id 1. [* ID -> _IDENTITY_.externalId] 2. [\$replace(ts, ' ','T') -> time] 3. [\$map(\$spread(meas), function(\$v, \$k) { { \$keys(\$v): { "value": \$lookup(\$v,\$keys(\$v))[0], "unit": "l/h" } } }) -> \$merge() -> onguardMeasurement]	M	{ "ID": "0018", "meas": { "Product1_Flow": [14.93], "Water_Flow": [18.54], "Product2_Flow": [272.9],], "ts": "2024-06-18 13:20:45.000Z", "timestampUTC": "2024-06-18T13:20:45.000Z" }	{ "onguardMeasurement": null, "time": "2022-08-05T00:14:49.389+02:00", "type": "onguardMeasurement" }	Map the structure under meas as fragments in the measurements. The number of keys in meas can vary and must therefore be generated dynamically.
24	EXTENSION_SOURCE_TARGET	mappingTopic: extension/source_target mappingTopicSample: extension/source_target check: Use external id Extensions for Processor Extension Source Target: select Events for: select Defined in cumulocity-dynamic-mapper/dynamic-mapping-extension/src/main/java/dynamic/mapping/processor/extension/external/ProcessorExtensionCustomAlarm.java	A	{ "alarmType": "MAJOR", "message": "This is an alamr for the extension!", "type": "c8y_ExtensionAlarm", "externalId": "berlin_01", "time": "2024-06-18T13:20:45.000Z" }	{ "onguardMeasurement": null, "time": "2022-08-05T00:14:49.389+02:00", "type": "onguardMeasurement" }	the extraction and the substitution in the targetPayload are implemented in java, see. ProcessorExtensionCustomAlarm.java. This is useful if the processing of the source payload can't be achieved in JSONata and the building of the targetP payload (Cumulocity) cant be achieved by standard substitutions.

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
25	JSON	mappingTopic: alarm/tires_c8ySourceId mappingTopicSample: alarm/tires_c8ySourceId check: Use external id sub: 1.[bus_c8ySourceId -> _IDENTITY_.c8ySourceId] 2.[msg_type -> type] 3.[tx -> text]	A	{ "msg_type": "c8y_FlatTireAlarm", "tx": "Left rear tire loses air!", "bus_c8ySourceId": "10203040" }	{ "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 identified by a Cumulocity id. This is in contrast to the mapping 19, which uses the external id: "berlin_01"

Sample Mapping #	Mapping Type	Topics/Substitutions	API	Template-Source	Target-Payload	Expected Result
51	JSON OUTBOUND	<p>publishTopic: evt/outbound/# mappingTopicSample: evt/outbound/berlin_01 filterMapping: bus_event NOTE: for outbound mappings no tt (template topic) is defined. check: Use external id</p> <p>sub: 1.[[_IDENTITY_.c8ySourceId -> _TOPIC_LEVEL_[2]] 2.[[type -> eventType] 3.[[\$now() -> time] 4.[[bus_event -> bus_event] 5.[[_IDENTITY_.externalId -> deviceId]</p>	E	<pre>{ "type": "c8y_BusEvent", "text": "Bus was stopped", "time": "2022-08-05T00:14:49.389+02:00", "bus_event": "stop_event" }</pre>	<pre>{ "deviceId": "909090", "description": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "eventType": "TestEvent", "bus_event": "stop_event", "_TOPIC_LEVEL_": ["evt", "outbound", "berlin_01"] }</pre>	<p>Publish mqtt msg. with event on topic 'evt/outbound/berlin_01' Use following command to create sample event: c8y events create --device "YOUR_DEVICE_ID" --data 'bus_event="stop_event", text="Bus was stopped today!", type="c8y_BusEvent"'</p>
52	JSON OUTBOUND	<p>publishTopic: opp/outbound/# mappingTopicSample: opp/outbound/berlin_01 filterMapping: bus_opp check: Use external id</p> <p>NOTE: for outbound mappings no tt (template topic) is defined.</p> <p>sub: 1. [_IDENTITY_.externalId -> _TOPIC_LEVEL_[2]] choose: Resolve to externalId 2. [bus_opp -> decription] 3. [* _IDENTITY_.c8ySourceId -> c8y_Id]</p>	O	<pre>{ "bus_opp": "New engine restart operation!" }</pre>	<pre>{ "c8y_Id": "909090", "decription": "dummy operation", "type": "bus_operation", "_TOPIC_LEVEL_": ["opp", "outbound", "berlin_01"] }</pre>	<p>Publish mqtt msg. with operation on topic 'opp/outbound/berlin_01' Use following command to create sample event: c8y operations create --device "YOUR_DEVICE_ID" --data 'bus_opp="New engine restart operation!"' Note: The option "resolve2ExternalId = false" esures that the C8Y internal device id is used in substitution 3.</p>
53	JSON OUTBOUND	<p>publishTopic: evt/outDeviceId/# mappingTopicSample: evt/outDeviceId/berlin_01 filterMapping: bus_event NOTE: for outbound mappings no tt (template topic) is defined. check: Use external id</p> <p>sub: 1.[[_IDENTITY_.externalId -> _TOPIC_LEVEL_[2]] 2.[[type -> eventType] 3.[[\$now() -> time] 4.[[bus_event -> bus_event] 5.[[_IDENTITY_.c8ySourceId -> c8y_Id] 6. [text -> description]</p>	E	<pre>{ "type": "c8y_BusEvent", "text": "Bus was stopped", "time": "2022-08-05T00:14:49.389+02:00", "bus_event": "stop_event" }</pre>	<pre>{ "c8y_Id": "909090", "description": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "eventType": "TestEvent", "bus_event": "stop_event", "_TOPIC_LEVEL_": ["evt", "outbound", "berlin_01"] }</pre>	<p>Publish mqtt msg. with event on topic 'evt/outbound/berlin_01' Use following command to create sample event: c8y events create --device "YOUR_DEVICE_ID" --data 'bus_event="stop_event", text="Bus was stopped today!", type="c8y_BusEvent"'</p>
54	JSON OUTBOUND	<p>publishTopic: ednvcfnr-event mappingTopicSample: ednvcfnr-event filterMapping: bus_event NOTE: for outbound mappings no tt (template topic) is defined. check: Use external id check: Use message context</p> <p>sub: 1.[[_IDENTITY_.externalId -> _CONTEXT_DATA_.key] 2.[[type -> eventType] 3.[[\$now() -> time] 4.[[bus_event -> bus_event] 5.[[_IDENTITY_.externalId -> deviceId]</p>	E	<pre>{ "type": "c8y_BusEvent", "text": "Bus was stopped", "time": "2022-08-05T00:14:49.389+02:00", "bus_event": "stop_event" }</pre>	<pre>{ "deviceId": "909090", "description": "This is a new test event.", "time": "2022-08-05T00:14:49.389+02:00", "eventType": "TestEvent", "bus_event": "stop_event", "_TOPIC_LEVEL_": ["evt", "outbound", "berlin_01"] }</pre>	<p>Publish msg. to kafka with event on topic 'ednvcfnr-event' Use following command to create sample event: c8y events create --device "YOUR_DEVICE_ID" --data 'bus_event="stop_event", text="Bus was stopped today!", type="c8y_BusEvent"'. The Cumulocity device id is added as key (message context).</p>