Currently the rrps.dt.sapmasterdata integrator expects fields for both the **Engine** and **the Aggregate (Powerpack)** from a single request to

[sap/opu/odata/sap/Y\_DIGITAL\_TWIN\_SRV/EQUI01Set?$filter=SernrAgg%20eq%20'{asset\_id}'&$format=json](https://sap-fes-p.org-intra.net/sap/opu/odata/sap/Y_DIGITAL_TWIN_SRV/EQUI01Set?$filter=SernrAgg%20eq%20%27%7basset_id%7d%27&$format=json)

with the serial number of the aggregate as the value of asset\_id.

The flow needs to be changed to be compatible with the updated SAP APIs.

The integrator configuration needs to have two endpoint parameters:

* hierarchy\_endpoint: this points to the endpoint of the Equipment Hierarchy API
* master\_endpoint: this points to the endpoint of the Equipment Master API

The new flow to populate a SapMasterDataSet event would then become as follows:

* Call the hierarchy endpoint with the following query string:

$filter=Sernr%20eq%20'<serial number of the aggregate>'&?format=json .

For example, $filter=Sernr%20eq%20'4711-0006'&?format=json.

* From the response to the previous request, find the element in the array that has **Hequi** set to an empty string and save the value of **Equnr** to a temporary variable named **aggregate\_equipment\_number**.
* Find the element in the array that has **Hequi** equal to **aggregate\_equipment\_number** and **Eqart**equal to **ENG**; store the value of **Sernr** to a variable named **engine\_serial**.
* Call the master\_endpoint with the following query string:

 $filter=Sernr%20eq%20'<serial number of the aggregate>'&?format=json . For

example, $filter=Sernr%20eq%20'4711-0006'&?format=json.

* Process the data returned from the response as follows (including as usual checks for empty fields):

event\_data = data['d']['results'][0]

event\_data['Datab'] = math.floor(datetime.datetime.strptime(event\_data['Datab'], '%Y-%m-%d').timestamp()) \* 1000

event\_data['Datbi'] = math.floor(datetime.datetime.strptime(event\_data['Datbi'], '%Y-%m-%d').timestamp()) \* 1000

event\_data['EqunrAgg'] = event\_data['Equnr']

event\_data['MaktxAgg'] = event\_data['Maktx']

event\_data['MatnrAgg'] = event\_data['Matnr']

event\_data['SernrAgg'] = event\_data['Sernr']

for field in ('Equnr', 'Maktx', 'Matnr', 'Sernr'):

    delete event\_data[field]

* Call the master endpoint by passing the **engine\_serial** as the asset id
* Insert the data returned from the response into event\_data as follows:

engine\_data = resp.json()['d']['results'][0]

event\_data['SernrEng'] = engine\_data['Sernr']

event\_data['MatnrEng'] = engine\_data['Matnr']

event\_data['MaktxEng'] = engine\_data['Maktx']

event\_data['EqunrEng'] = engine\_data['Equnr']

* the resulting event\_data can be then returned to the function that builds the event.

To test the implementation, the endpoint in rrps.dt.mockdataapi that returns masterdata should be changed to return different responses depending on the asset id

If the asset id is 4711-0006 it should return the following response:

{

  "d": {

    "results": [

      {

        "Equnr": "000000000200000031",

        "Matnr": "R-REM-0002",

        "Maktx": "Reman engine 0002",

        "Sernr": "4711-0006",

        "Kunde": "",

        "Name1": "",

        "Datab": "2015-02-10",

        "Datbi": "2015-12-31",

        "Yybau": "SYS",

        "Yyerz": "",

        "Vkbur": "0100",

        "Eqart": "ENG"

      }

    ]

  }

}

If the asset id is 150001, the response should be as follows:

{

  "d": {

    "results": [

      {

        "Equnr": "000000000200000465",

        "Matnr": "SYSTEM",

        "Maktx": "SYSTEM",

        "Sernr": "150001",

        "Kunde": "",

        "Name1": "",

        "Datab": "2015-02-10",

        "Datbi": "9999-12-31",

        "Yybau": "SYS",

        "Yyerz": "",

        "Vkbur": "0100",

        "Eqart": "SYS"

      }

    ]

  }

}

An additional endpoint should mock the response of the Equipment Hierarchy API and return the sample json from it from the documentation if the asset id is 150001.

The sample JSON for the Equipment Hierarchy API would be as follows:

{

  "d": {

    "results": [

      {

        "Equnr": "000000000200000465",

        "Sernr": "1000015",

        "Eqart": "SYS",

        "Hequi": ""

      },

      {

        "Equnr": "000000000200000031",

        "Sernr": "4711-0006",

        "Eqart": "ENG",

        "Hequi": "000000000200000465"

      }

    ]

  }

}

The masterdata endpoint should return the following JSON when the asset id is 1000015:

{

  "d" : {

    "results" : [

      {

        "\_\_metadata" : {

          "id" : "https://sap-fes-i.org-intra.net/sap/opu/odata/sap/Y\_DIGITAL\_TWIN\_SRV/EQUI01Set('000000000200000082')",

          "uri" : "https://sap-fes-i.org-intra.net/sap/opu/odata/sap/Y\_DIGITAL\_TWIN\_SRV/EQUI01Set('000000000200000082')",

          "type" : "Y\_DIGITAL\_TWIN\_SRV.EQUI01"

        },

        "Equnr": "000000000200000081",

        "Sernr": "1000015",  
        "Maktx": "AUFNAHMEKOERPER",

        "Matnr": "12V2000C10",

        "Kunde" : "0022111236",

        "Name1" : "Land Badenwürtenberg",

        "Datab" : "2017-09-08",

        "Datbi" : "9999-12-31",

        "Yybau" : "030",

        "Yyerz" : "10V1600A50",

        "Vkbur" : "0101",

        "Eqart" : "SYS",

        "EquiPartSet" : {

          "\_\_deferred" : {

            "uri" : "https://sap-fes-i.org-intra.net/sap/opu/odata/sap/Y\_DIGITAL\_TWIN\_SRV/EQUI01Set('000000000200000082')/EquiPartSet"

          }

        }

      }

    ]

  }

}

And the following when the asset id is 4711-001 :

{

  "d" : {

    "results" : [

      {

        "\_\_metadata" : {

          "id" : "https://sap-fes-i.org-intra.net/sap/opu/odata/sap/Y\_DIGITAL\_TWIN\_SRV/EQUI01Set('000000000200000082')",

          "uri" : "https://sap-fes-i.org-intra.net/sap/opu/odata/sap/Y\_DIGITAL\_TWIN\_SRV/EQUI01Set('000000000200000082')",

          "type" : "Y\_DIGITAL\_TWIN\_SRV.EQUI01"

        },

        "Equnr" : "000000000200000082",

        "Matnr" : "R16V4000G83",

        "Maktx" : "R16V4000G83 REMAN ENGINE",

        "Sernr" : "4711-001",

        "Kunde" : "0022111236",

        "Name1" : "Land Badenwürtenberg",

        "Datab" : "2017-09-08",

        "Datbi" : "9999-12-31",

        "Yybau" : "030",

        "Yyerz" : "10V1600A50",

        "Vkbur" : "0101",

        "Eqart" : "ENG",

        "EquiPartSet" : {

          "\_\_deferred" : {

            "uri" : "https://sap-fes-i.org-intra.net/sap/opu/odata/sap/Y\_DIGITAL\_TWIN\_SRV/EQUI01Set('000000000200000082')/EquiPartSet"

          }

        }

      }

    ]

  }

}

if Datbi is 0000-00-00 it should be set to None

in the event

if Datab is 0000-00-00 an error should be logged, and the event should not be processed

as Datab should always be set, Datbi might be not

if Datbi (Agg data/1000015) is 0000-00-00 it should be set to None 🡺 able to publish event

if Datab (Eng data/411-001) is 0000-00-00 an error should be logged, and the event should not be processed 🡺 error logged but event processed and give error while publishing

if Datbi (Eng data/411-001) is 0000-00-00 it should be set to None 🡺 able to publish event

(ValueError: None (type <class 'NoneType'>) do not match [{'logicalType': 'date', 'type': 'long'}])