How to Guide

Flatten JSON (DP)

Version: R-4.5

Contents

1. Component Overview

BDB Pipeline provides the Flatten JSON component for flatting a nested JSON as well as filtering a data from a nested JSON. It can filter and remove the unnecessary dumping of data on a Kafka topic.

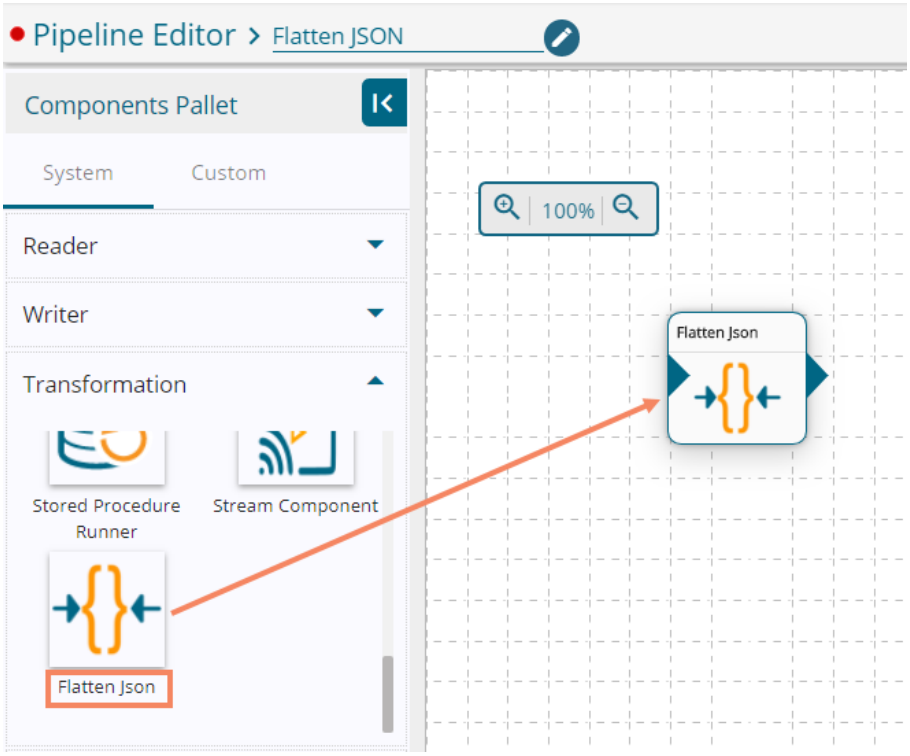
It can run in the following scenarios:

1. Real-Time
2. Batch Process

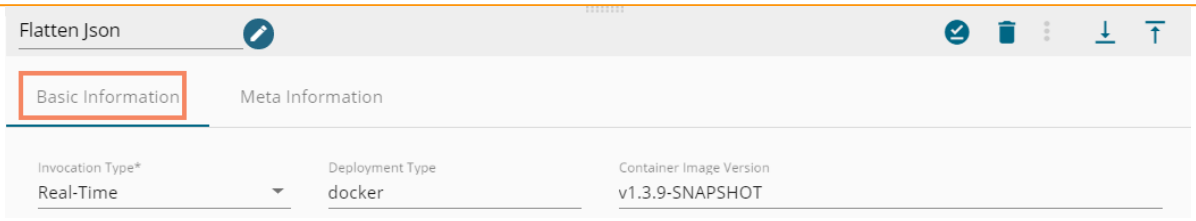
2. Step by Step Process to Use Flatten JSON

The steps to configure the Flatten JSON component is described in this section.

i) Navigate to the Pipeline Editor.
ii) Drag the Flatten Json from the Transformation section provided under the Components Pallet and drop to the Pipeline Workspace.



iii) Click the dragged Flatten JSON component to get the Configuration fields:
iv) The Basic Information tab opens by default.
   a. Select the invocation type (Real-Time/Batch)
   b. Deployment Type: It comes preselected based on the component.
   c. Container Image Version: It comes preselected based on the component.



v) Click on the Meta Information tab to open the configuration fields for the selected columns.

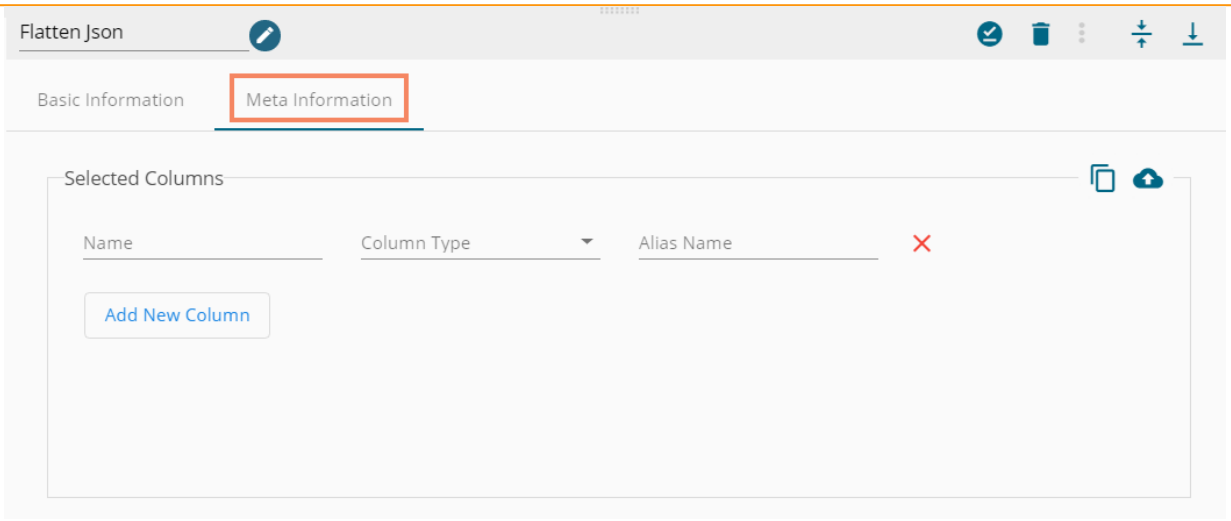
a. Name: The user must provide the full path of data by using dot(.) which he wants to retrieve from a nested json.
For instance - We have a nested JSON of the following format, and we want to retrieve device type of the Device in dashboard_1 we must give the name as **dashboard_1.Device.devicetype.**


```
"dashboard_1": {
    "Device": [
      {
        "devicetype": "Motor-N1",
        "Vibration": 53,
        "Temperature": 23
      },
      {
        "devicetype": "Motor-N2",
        "Vibration": 48,
        "Temperature": 40
      },
      {
        "devicetype": "Motor-S1",
        "Vibration": 98,
        "Temperature": 33
      },
      {
        "devicetype": "Motor-S2",
        "Vibration": 77,
        "Temperature": 37
      },
      {
        "devicetype": "Gearbox-LS",
        "Vibration": 56,
        "Temperature": 43
      },
      {
        "devicetype": "Gearbox-SS",
        "Vibration": 97,
        "Temperature": 50
      }
    ],
    "Brake": [
      {
        "devicetype": "Brake-N",
        "Thickness": 19,
        "Temperature": 414
      },
      {
```

```
            "devicetype": "Brake-S",
            "Thickness": 11,
            "Temperature": 397
          }
        ],
        "RPM": 675,
        "LOAD(T)": 40,
        "MTBF": 562,
        "MTTR": 62,
        "RUN": 3609,
        "RTBM": 317
      },
      "dashboard_2": {
        "Cast_Spd_St1": 0.013,
        "Cast_Spd_St2": -0.01,
        "Ladle_Gate_Hyd_Press": -181.277,
        "Mold_CW_Back_St1": 4365,
        "Mold_CW_Back_St2": 4338,
        "Mold_CW_East_St1": 459,
        "Mold_CW_East_St2": 469,
        "Mold_CW_Front_St1": 4348,
        "Mold_CW_Front_St2": 4443,
        "Mold_CW_West_St1": 461,
        "Mold_CW_West_St2": 479,
        "Mold_Eff_Back_St1": -1308,
        "Mold_Eff_Back_St2": -1018,
        "Mold_Eff_East_St1": -194,
        "Mold_Eff_East_St2": -177,
        "Mold_Eff_Front_St1": -1137,
        "Mold_Eff_Front_St2": -1375,
        "Mold_Eff_West_St1": -200,
        "Mold_Eff_West_St2": -193,
        "Mold_Lvl_St1": 3,
        "Mold_Lvl_St2": 1,
        "Tundish_Temp_5": 2886,
        "DO_High_Cast_Spd_St1": 0,
        "DO_High_Cast_Spd_St2": 1,
        "Full_Time_Local": 1559641339126
      }
    }
```


b. Column Type: Select a column type using the drop-down.
   (The supported column types are String, Long, Integer, Float, Double, Date, Date Time)
c. Alias Name: The user defined name for the selected column.
d. Click the '**Add New Column**' option to add more columns.

vi) Click the '**Save**' icon to save the configuration.



Note:

e. The Flatten JSON component cannot generate data in rows.

f. The Flatten JSON component must have an in-event and out-event in the workflow, as displayed below: