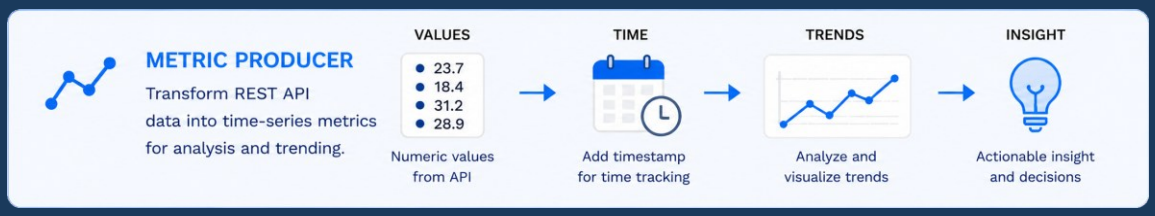
CEEVIEW

REST Monitor Learning Series

Tutorial 2

Creating a Time-Series Metric Producer

Generating Native Ceeview Metrics



Version 2.19 · DataIntegration Package

This tutorial is Tutorial 2 in the REST Monitor Learning Series and assumes Tutorial 1 – Creating a Multi-Asset Health Producer has been completed.

In this tutorial, you'll extend the same Nordic Retail Example REST Monitor by adding a Metric Producer that creates native Ceeview time-series Metrics from the same REST API response.

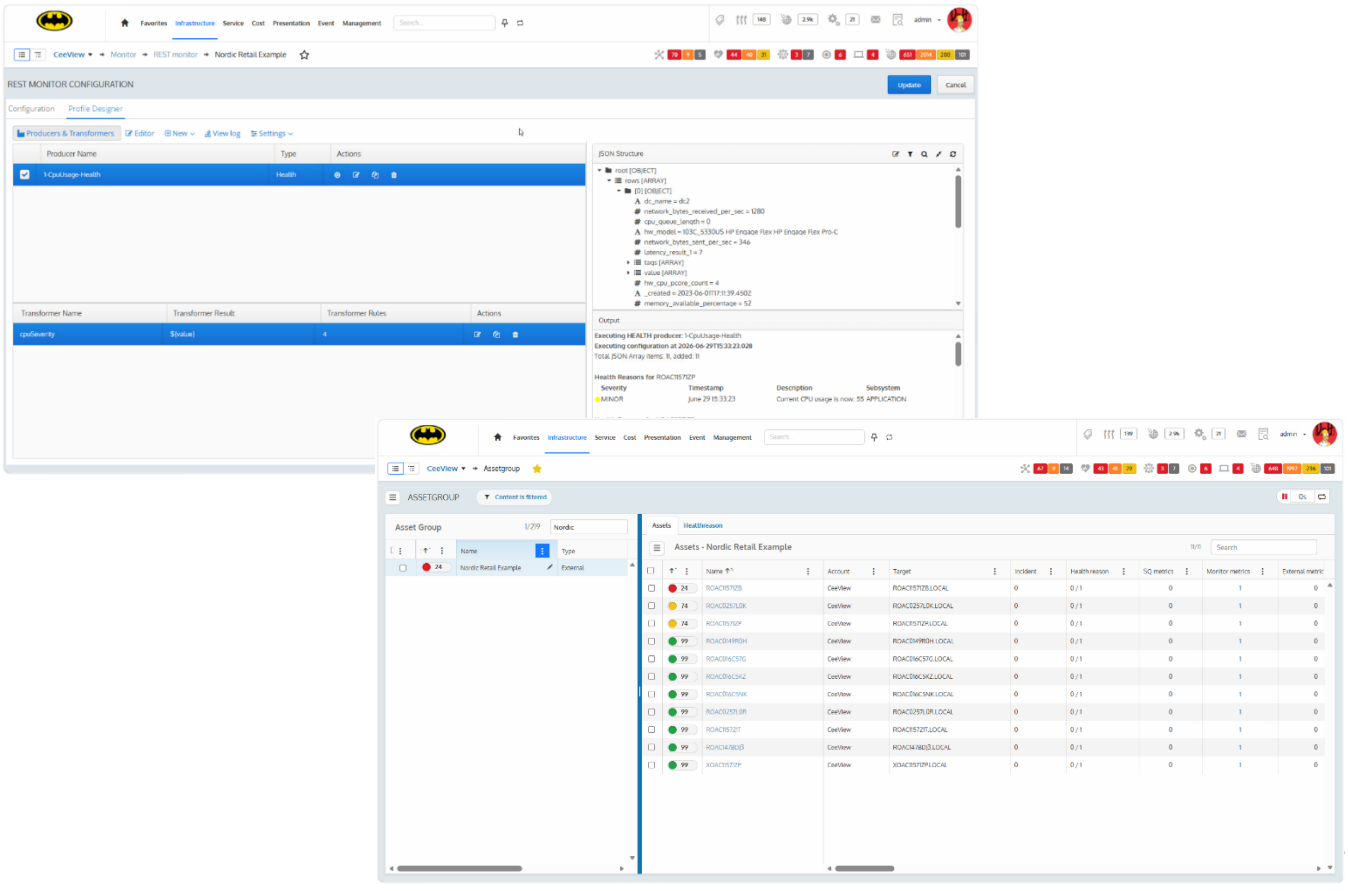
1. Starting Point from Tutorial 1

Before beginning this tutorial, your environment should match the completed configuration from **Tutorial 1 – Creating a Multi-Asset Health Producer**.

You should already have:

✓ REST Monitor: **Nordic Retail Example**
✓ Health Producer: **1-CpuUsage-Health**
✓ Transformer: **cpuSeverity**
✓ Editor function: **stripWorkgroup()**
✓ Asset Group: **Nordic Retail Example** with eleven Health Assets

The first screenshot confirms the REST Monitor configuration from Tutorial 1. The second confirms that the Nordic Retail Example Asset Group and Health Assets have been created.

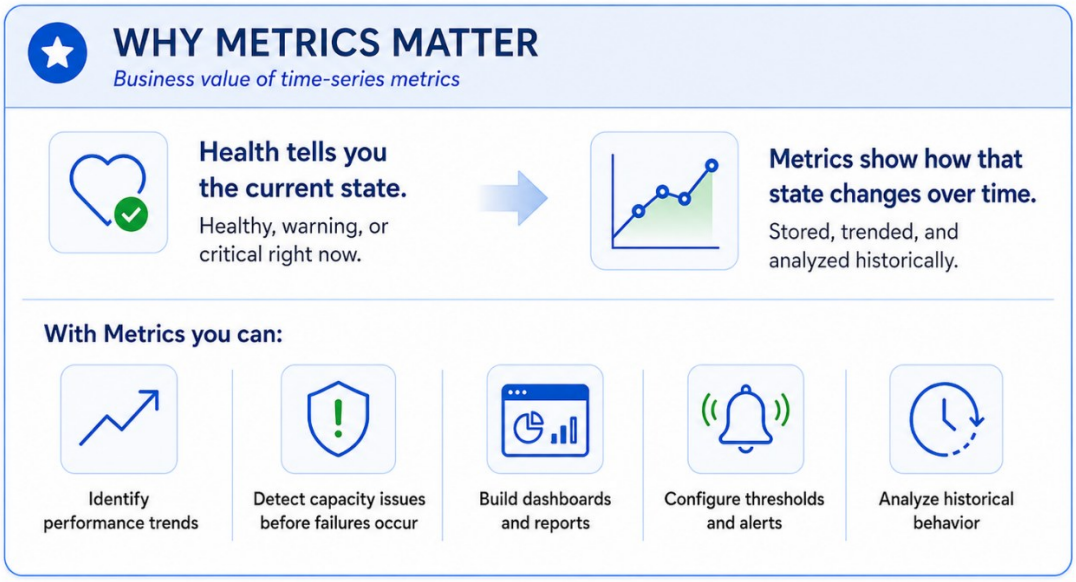


What You'll Add: This tutorial adds a Metric Producer to the existing Nordic Retail Example REST Monitor. When complete, the Health Producer will continue creating Health Assets, while the new Metric Producer records time-series Memory Usage Metrics for those same Assets.

2. Understanding Time-Series Metrics

The Health Producer shows the current state of each device. The Metric Producer records numeric API values over time, making those values available for trending, dashboards, reports, thresholds, and alerts.

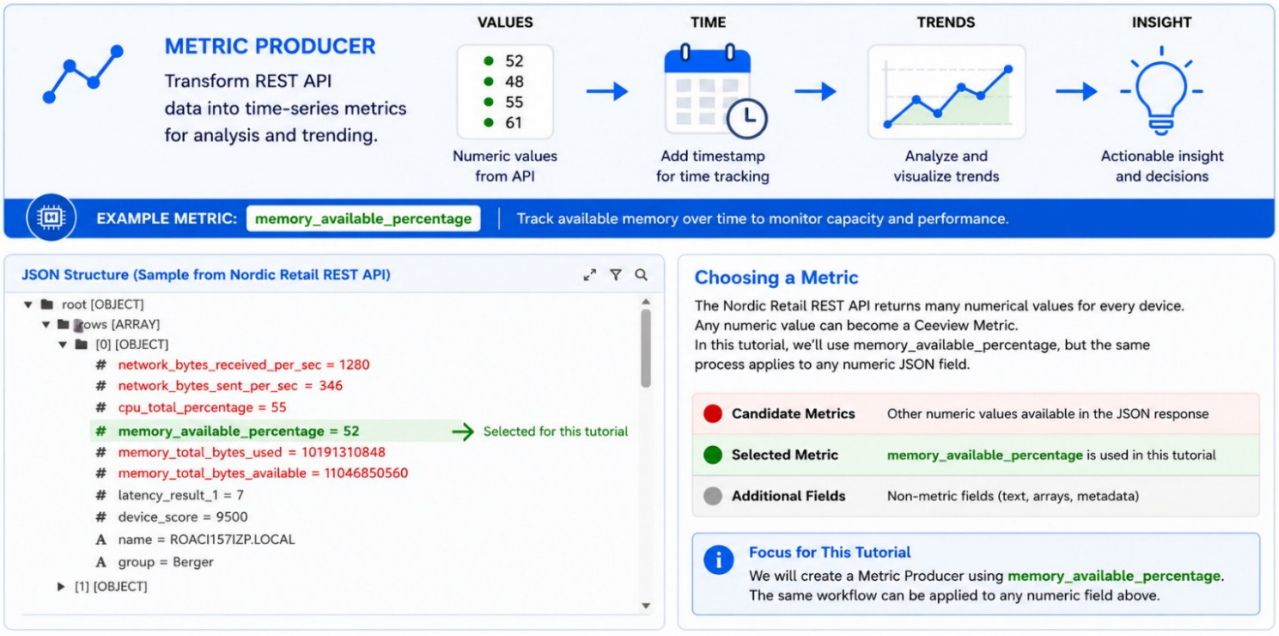
A single REST API often exposes many numerical values — from CPU, memory, and latency to usage, cost, sales, and other operational measurements. This tutorial uses one example metric, but the same configuration approach applies to virtually any numeric field returned by a REST API.



Choosing Your First Metric

In this tutorial, we'll use **memory_available_percentage** because it is easy to recognize in the JSON structure and produces a simple percentage-based Metric.

The following illustration shows the Metric Producer workflow used throughout this tutorial.



3. Extending the REST Monitor with Metrics

Now that the Tutorial 1 configuration is in place and the tutorial metric has been selected, this section extends the Nordic Retail Example REST Monitor by adding a Metric Producer. The Metric Producer will create one native Ceeview Memory Usage Metric for each Health Asset generated in Tutorial 1.

1 Create a Metric Producer

In the Profile Designer select New → Producer → Metric.

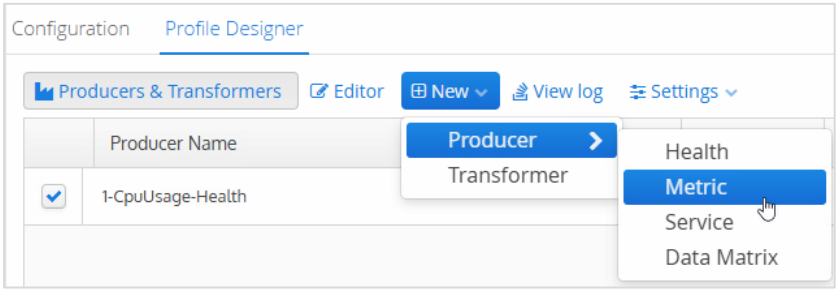


Figure 1 · Selecting New → Producer → Metric from the menu.

General Properties

Complete the General Properties tab as shown below. These settings define how the Metric Producer creates native Ceeview Metrics from the REST API response and associates each Metric with the corresponding Health Asset created in Tutorial 1.

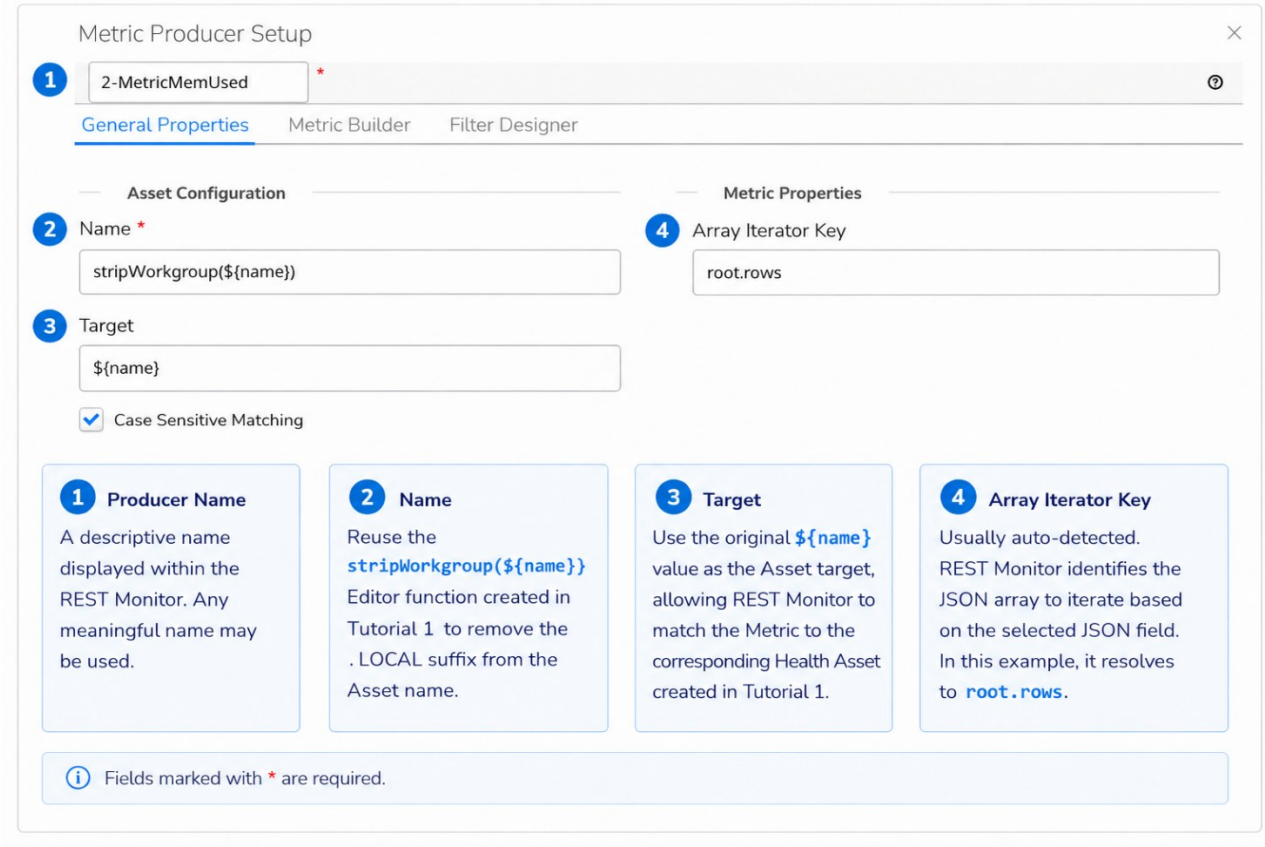


Figure 2 · Metric Producer General Properties configuration. These settings create one native Ceeview Metric for each device returned by the Nordic Retail REST API.

Use the following values when configuring the General Properties tab. The table below summarizes each setting and the value used throughout this tutorial.

#	Configuration	Tutorial Value	Purpose
1	Producer Name	2-MetricMemUsed	Display name within REST Monitor.
2	Name	stripWorkgroup(${name})	Reuse the Editor function from Tutorial 1 to remove the .LOCAL suffix
3	Target	${name}	Associates each Metric with its Health Asset.
4	Array Iterator Key	root.rows	Creates one Metric per JSON device.

Next, configure the Metric Builder to define the Metric information generated for each Asset.

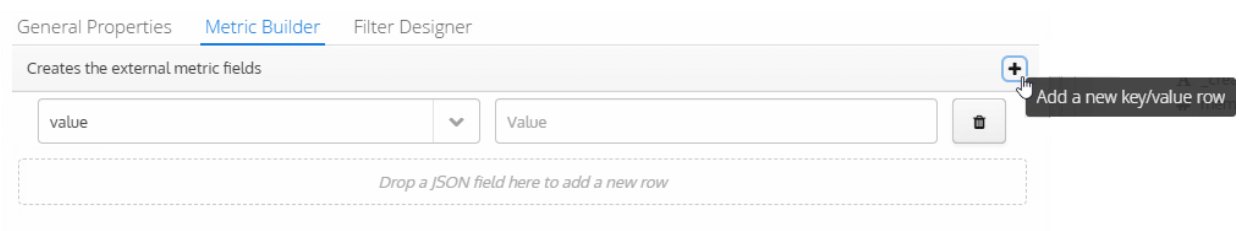
2 Configure the Metric Builder

The **Metric Builder** defines what information each Metric stores. Unlike the Health Producer, the Metric Builder starts by defining a Metric category before mapping the JSON value.

Complete the following steps to configure the Metric Builder.

Step 1 — Configure the Metric Category

1. Click the Metric Builder tab, then click the + icon to add the first row.



2. In the new row, change the left field from **value** to **category**.
3. In the right field, begin typing **mem**, then select **1.3.1: System/Memory/Memory Usage**.

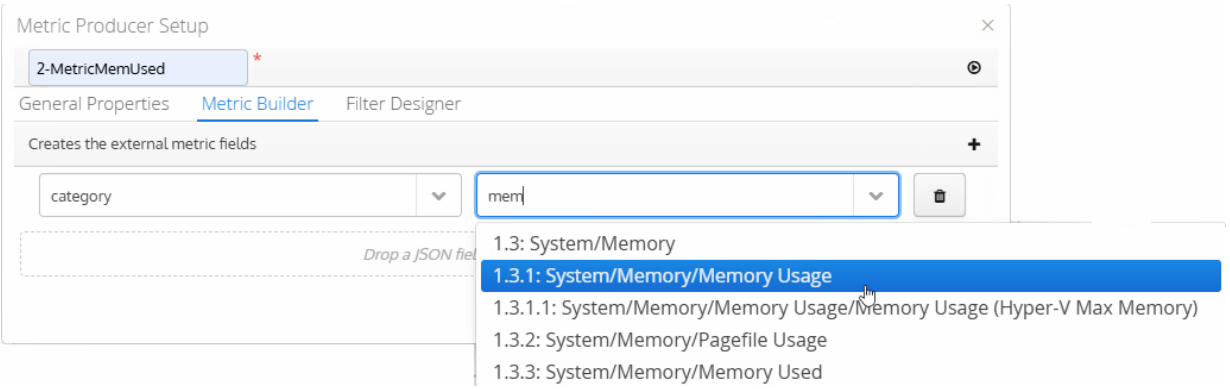


Figure 3 · The first Metric Builder row configured as the System/Memory/Memory Usage category.

The Metric Category determines where the generated Metric will appear within Ceeview.

Step 2 — Create the Metric Value

With the Metric Category defined, the next step is to create the Metric value itself. For this row, do not click **+**. Instead, drag the JSON field into the dashed Metric Builder drop zone.

1. Drag the **memory_available_percentage** field from the JSON Structure into the Metric Builder drop zone beneath the category row.

REST Monitor automatically creates a second row with:

- **Type:** value
- **Value:** ${memory_available_percentage}

No additional configuration is required for this row.

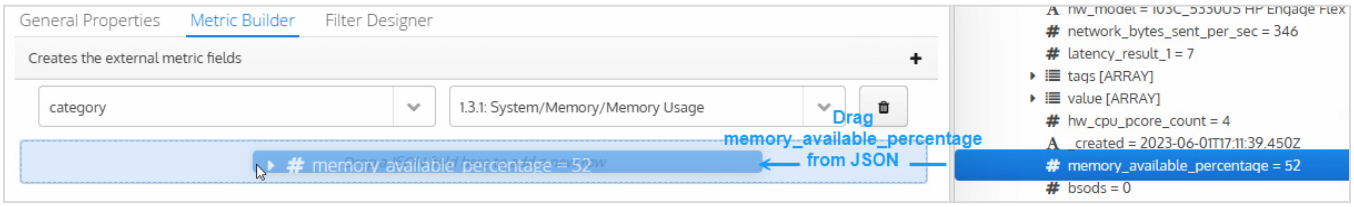


*Figure 4 · Dragging **memory_available_percentage** into the Metric Builder drop zone automatically creates the Metric Value row.*

The completed Metric Builder should now appear as shown below.

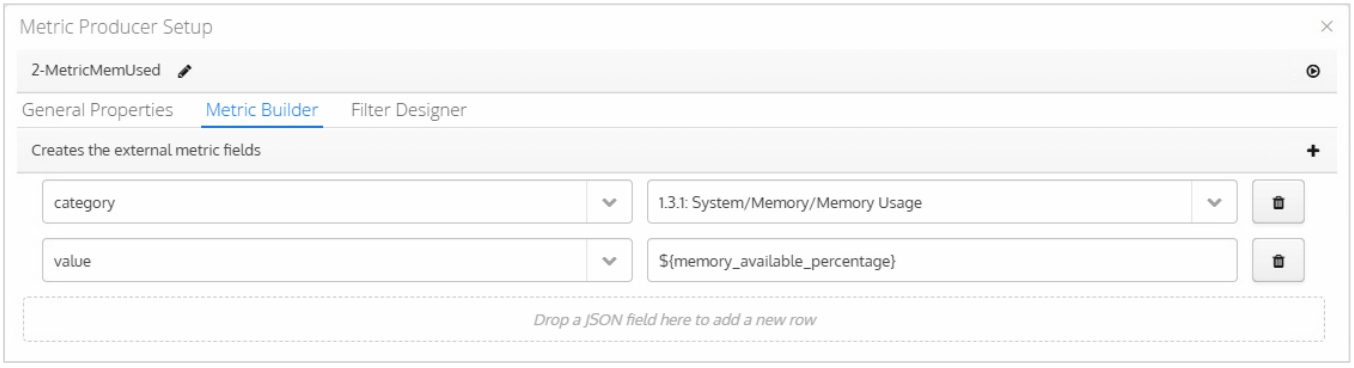


Figure 5 · Completed Metric Builder configuration showing both the Metric Category and Metric Value.

3 Execute and Activate the Metric Producer

Click **Run** in the upper-right corner of the Metric Producer window to execute the configuration using the current JSON response. This confirms that the Metric Producer can generate one Metric for each device returned by the REST API.

Review the Output panel to confirm that Metrics are generated before saving the Producer.

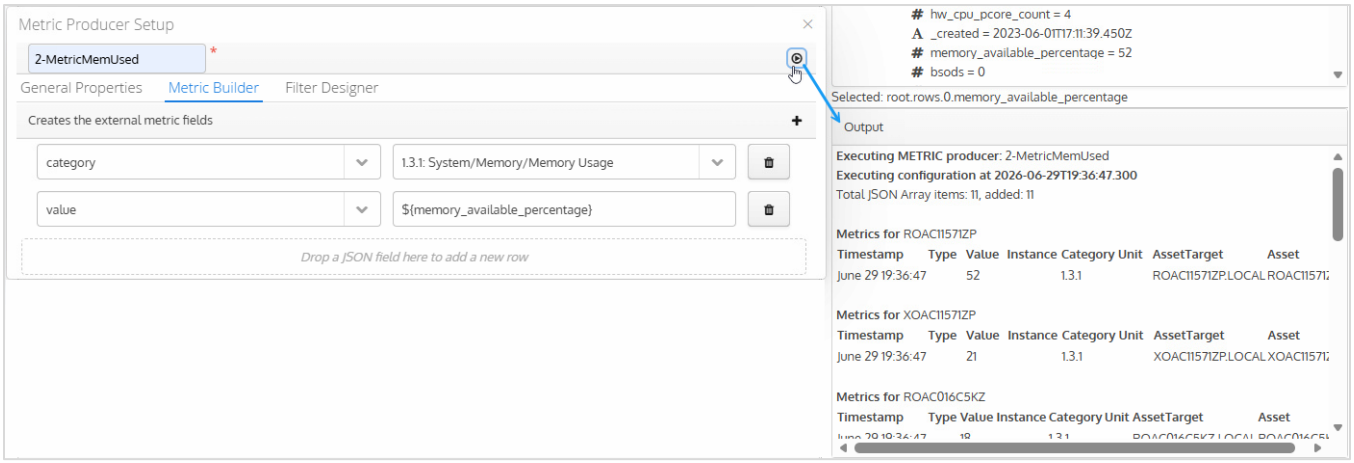


Figure 6 · Executing the Metric Producer generates one Metric for each device returned by the Nordic Retail Example REST API.

Note: After verifying the Output panel, close the Metric Producer window using the **X** in the upper-right corner. REST Monitor prompts you to save the Metric Producer before returning to the Profile Designer.

Update the REST Monitor

Click **Update** to save the REST Monitor configuration. The Metric Producer is now part of the Nordic Retail Example REST Monitor and will execute each time the monitor runs.

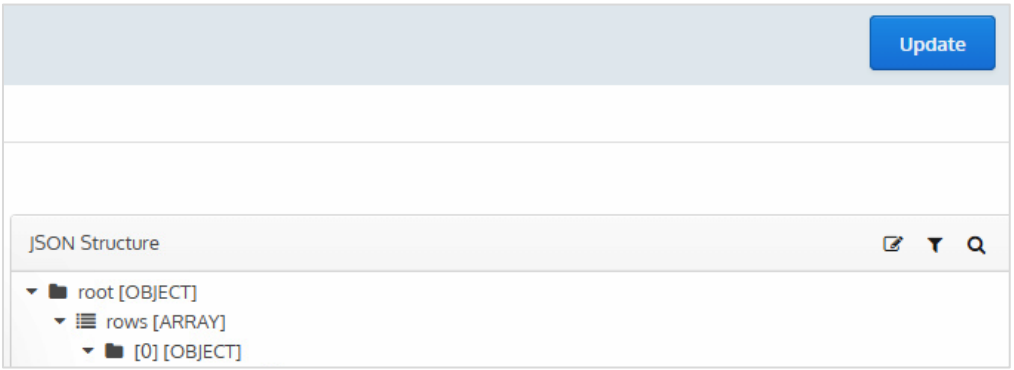
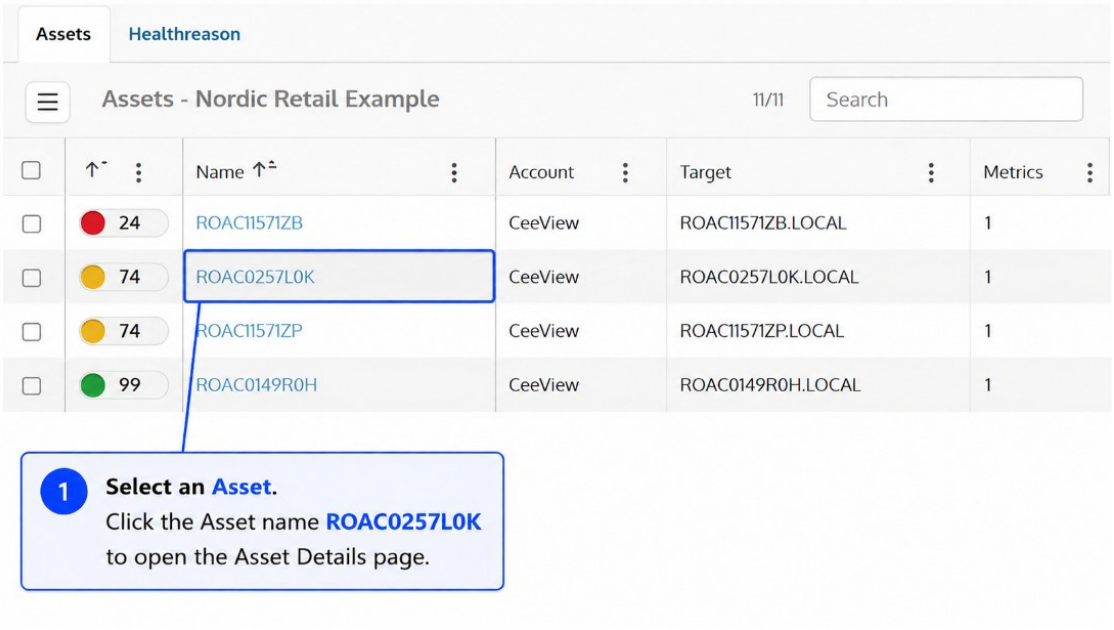


Figure 7 · Click Update to save the REST Monitor configuration and activate the new Metric Producer.

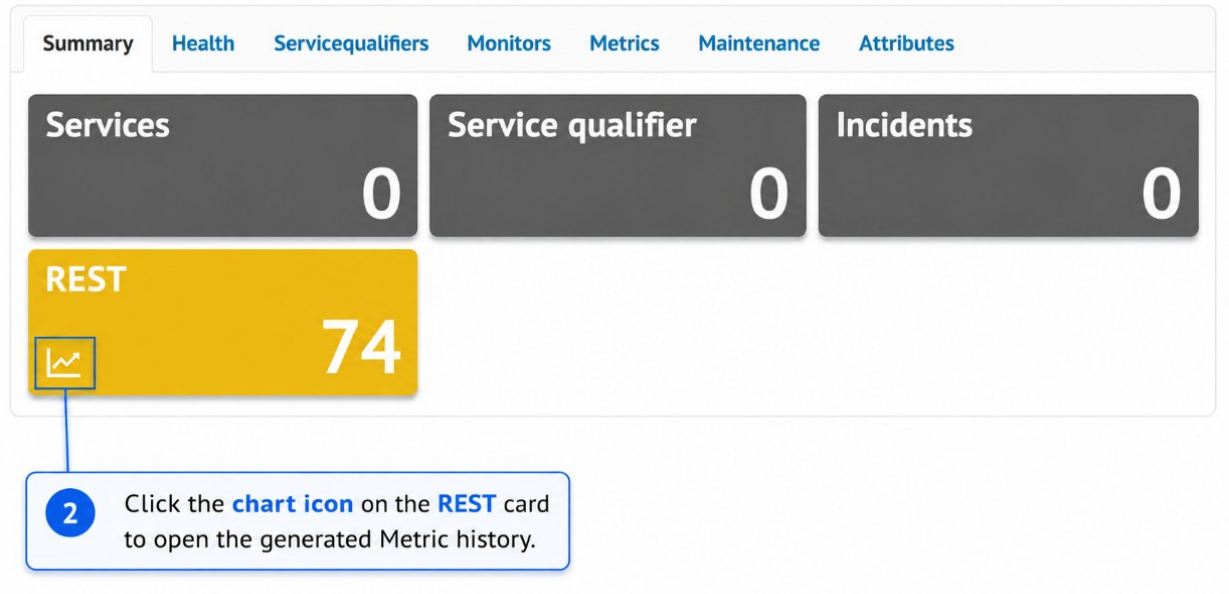
Verify the Generated Metrics

The Metric Producer has now created one Metric for each Health Asset in the Nordic Retail Example. The following steps demonstrate how to locate a generated Metric and view its historical values within Ceeview.

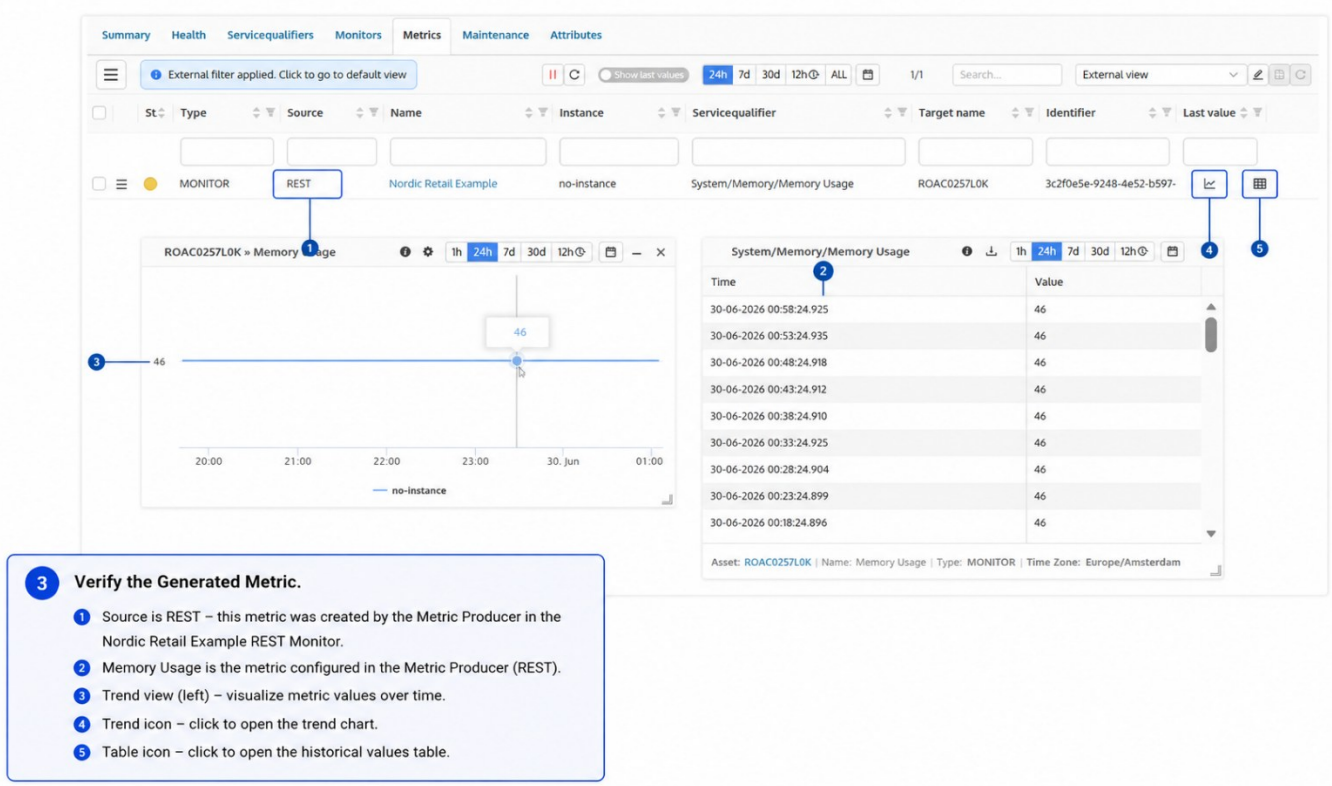
Step 1: Select an Asset.



Step 2: Within the Asset Details page, locate the REST card and click the chart icon.



Step 3: Verify the Generated Metric.



At this point, the REST Monitor is fully operational. Each execution interval retrieves the latest JSON data, updates the existing Health Assets, records new Memory Usage Metrics, and makes the latest values immediately available for dashboards, trending, reporting, and analysis.

Confirm the Updated REST Monitor

After saving the configuration, the Nordic Retail Example REST Monitor now contains two Producers. The Health Producer creates and maintains Health Assets, while the Metric Producer records historical Memory Usage Metrics for the same Assets.

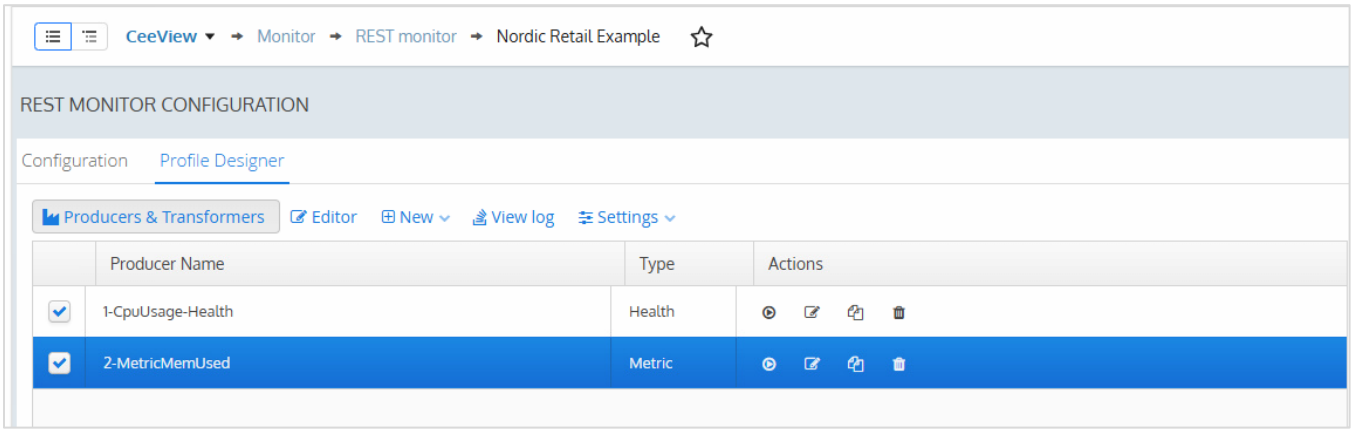
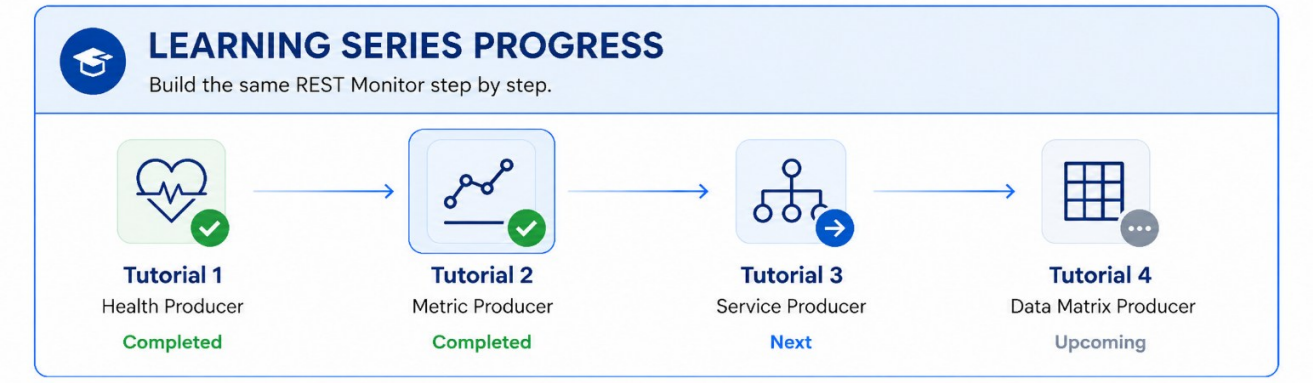


Figure 8 · Nordic Retail Example REST Monitor with both Health and Metric Producers configured.

Continue the REST Monitor Learning Series

Congratulations—you've now extended the Nordic Retail Example REST Monitor with a Metric Producer that creates time-series Metrics for the Health Assets created in Tutorial 1. The concepts introduced here show how REST API values can be stored, trended, and used throughout Ceeview.

The REST Monitor created in Tutorial 1 now contains two active Producers: one that creates Health Assets and one that records Metrics for those same Assets. In the next tutorial, you'll continue extending this same REST Monitor by adding a Service Producer.



Together, these Producer types demonstrate one of REST Monitor's greatest strengths: a single REST API can generate multiple native Ceeview object types from the same JSON source. As the learning series continues, this REST Monitor will evolve into a complete operational model by adding Service Models and Data Matrices.