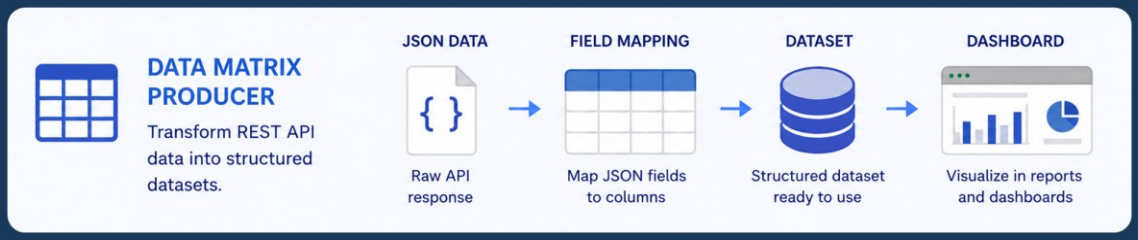
CEEVIEW

REST Monitor Learning Series

Tutorial 4

Creating a Structured Data Matrix Producer

Transforming REST API Responses into Reusable Operational Datasets



Version 2.19 · DataIntegration Package

This tutorial is Tutorial 4 in the REST Monitor Learning Series and assumes the Health, Metric, and Service Producer tutorial have been completed.

In this tutorial, you'll extend the Nordic Retail REST Monitor by adding a Data Matrix Producer that transforms REST API responses into reusable datasets for dashboards, reporting, and analysis.

1. Starting Point from Tutorials 1–3

Before beginning this tutorial, your REST Monitor should match the completed configuration from the previous 3 tutorials:

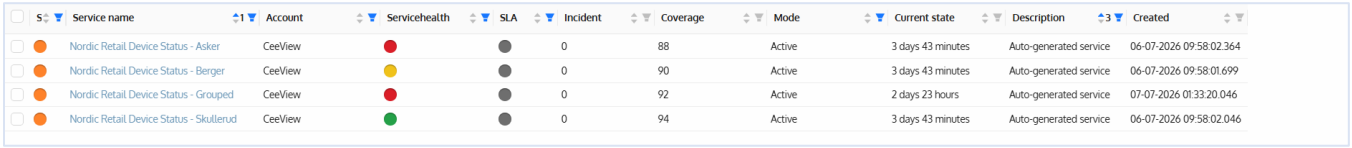
Tutorial 1 — Creating a Multi-Asset Health Producer
Tutorial 2 — Creating a Time-Series Metric Producer
Tutorial 3 — Creating Operational Service Models with Service Producers

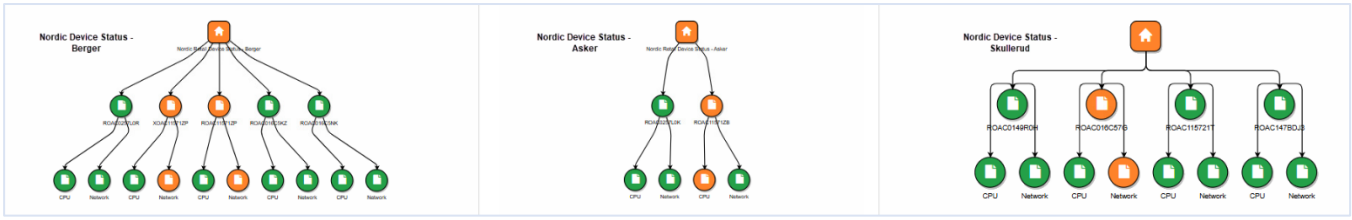
---

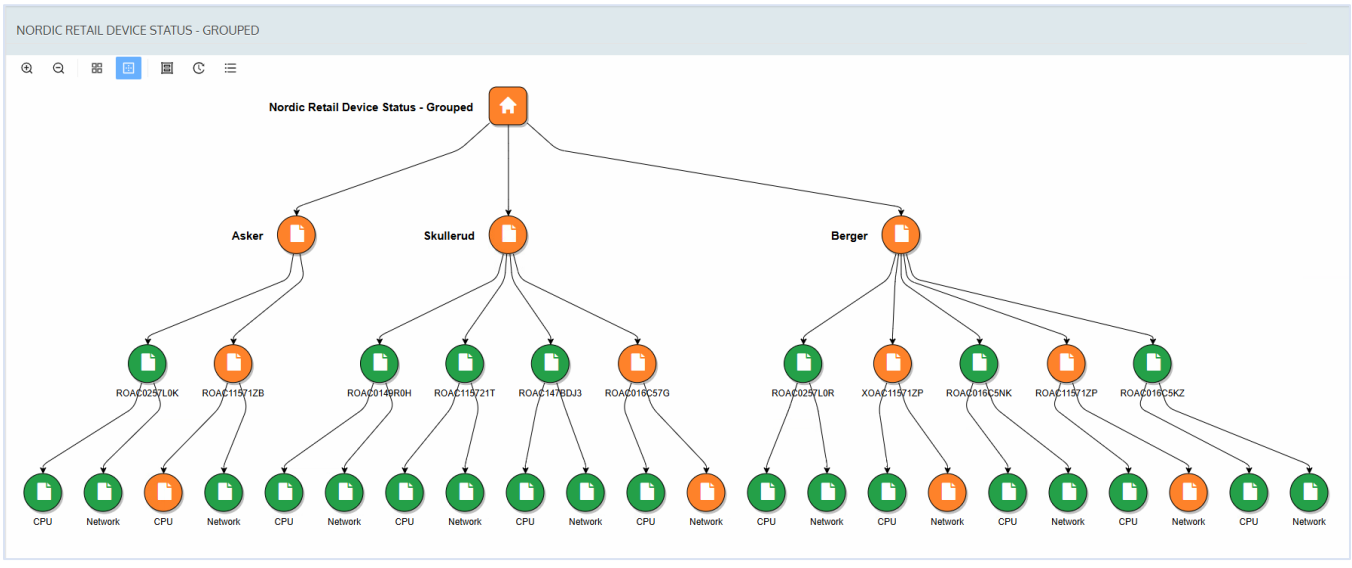
You should already have:

- ✓ REST Monitor: **Nordic Retail Example**
- ✓ Health Producer: **1-CpuUsage-Health**
- ✓ Metric Producer: **2-MetricMemUsed**
- ✓ Transformer: **cpuSeverity**
- ✓ Editor function: **stripWorkgroup()**
- ✓ Asset Group: **Nordic Retail Example** with eleven Assets
- ✓ Memory Usage Metrics associated with the same Health Assets
- ✓ Service Producers: **3-Svc-Group-CPU, 4-Svc-Group-Network**
- ✓ Service Producers: **5-Service-CPU, 6-Service-Network**
- ✓ Service Models: **Asker**, **Berger**, **Skullerud**, & **Nordic Retail Grouped**

The screenshots below confirm the completed environment from Tutorials 1–3. Health Assets and Metrics continue to be generated, while the Service Producers have created four operational Service Models—three location models and one unified Nordic Retail business service.







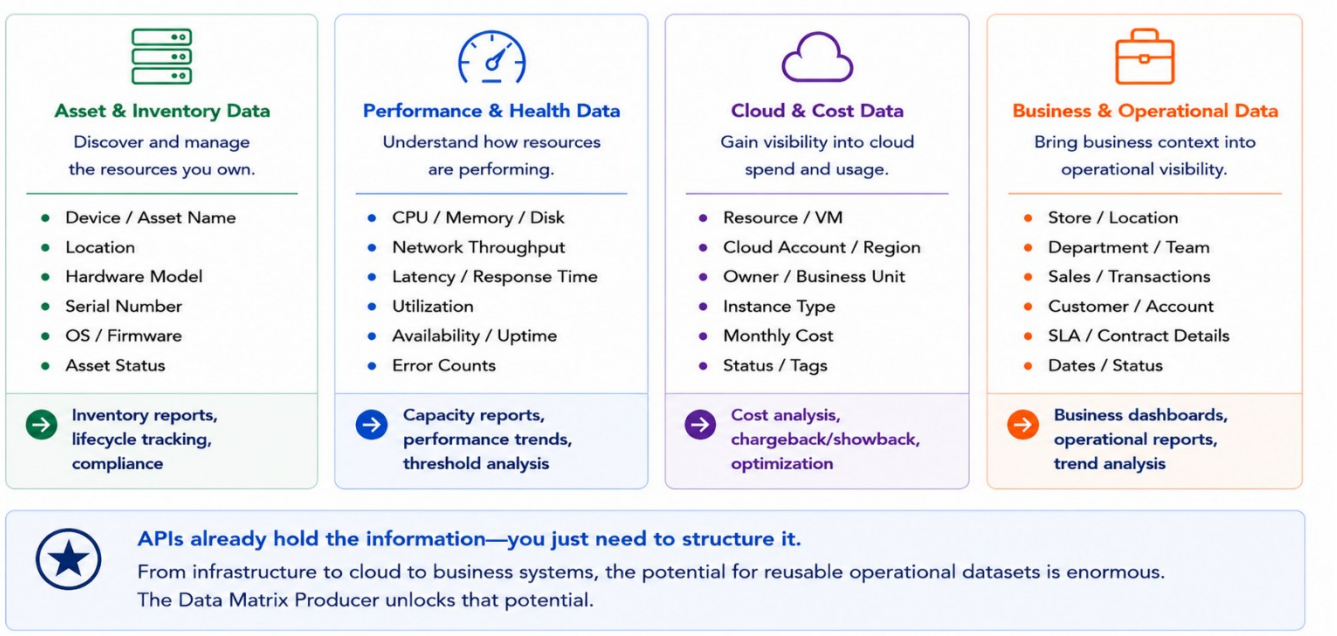
What You'll Add: This tutorial adds a **Data Matrix Producer** to the existing Nordic Retail Example REST Monitor. When complete, the Health Producer will continue creating Health Assets, the Metric Producer will continue recording time-series Metrics, the Service Producers will continue maintaining Service Models, and the new Data Matrix Producer will transform the same REST API response into reusable operational datasets for dashboards, reports, and analysis.

2. Making Hidden API Data Usable

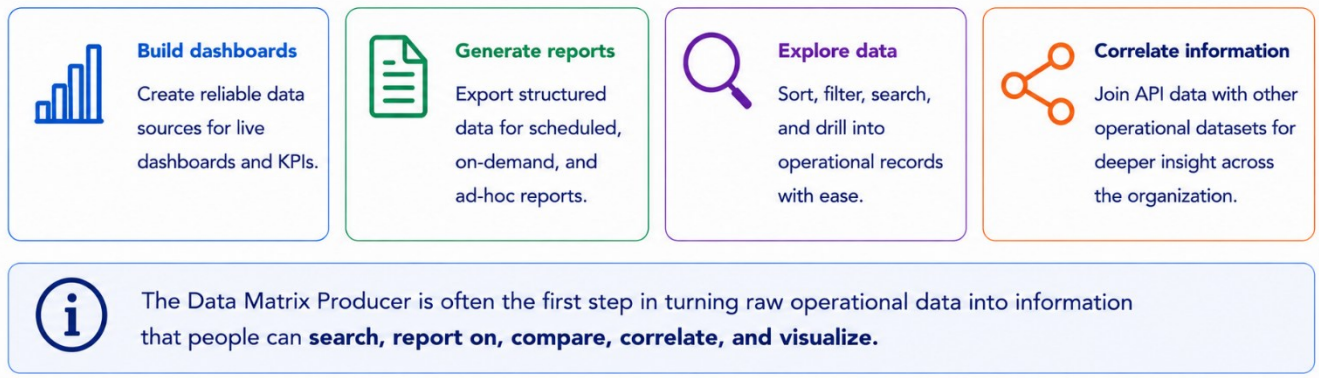
Every REST API contains far more operational information than can be visualized directly. Those APIs return data in JSON format—ideal for applications, but difficult to search, report on, or reuse.

The Data Matrix Producer reveals this hidden value by organizing JSON response intro structure datasets—turning raw API data into information that can be analyzed, compared and shared throughout Ceeview.

What Kinds of API Data Become Valuable When Structured?
API across technology and business systems expose many types of information. When structured into tables, this data becomes clear, searchable, and operationally useful.



With Data Matrices you can:



In the next section, you'll configure a Data Matrix Producer that maps JSON fields into columns, creates a reusable operational dataset, and stores it in Ceeview for dashboards, reports, search, and further analysis.

3. Extending the REST Monitor with Data Matrices

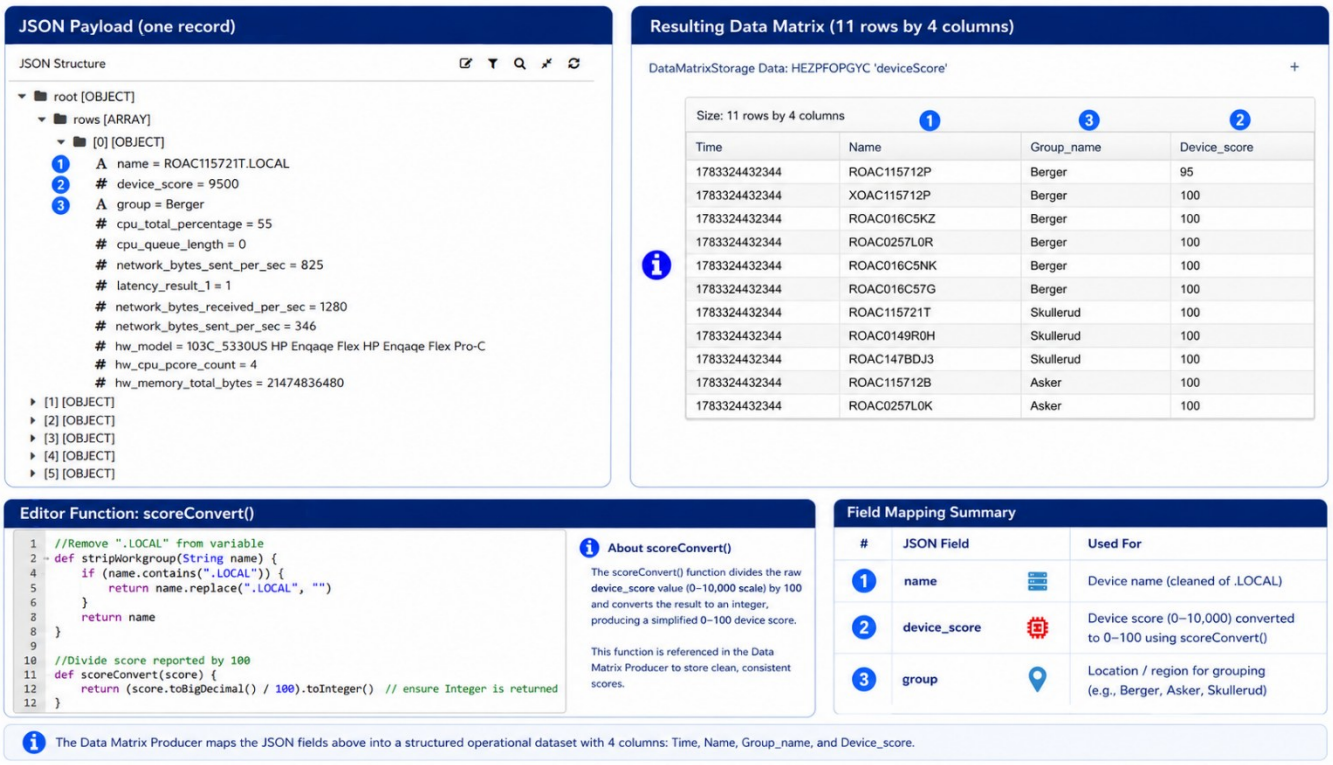
Now that the REST Monitor contains the Health, Metric, and Service Producers from the previous tutorials, this section adds a Data Matrix Producer that transforms operational JSON data into a reusable Ceeview dataset.

Rather than creating Assets, Metrics, or Service Models, the Data Matrix Producer maps JSON fields, applies transformation logic, and organizes the results into a structured table that can be searched, reported on, and reused throughout Ceeview.

The blueprint below illustrates how operational JSON data is transformed into a reusable Ceeview dataset.

Throughout this tutorial, the Data Matrix Producer:

- maps JSON fields into reusable columns
- applies editor functions where needed
- creates a structured Data Matrix
- stores the dataset in Ceeview Data Matrix Storage



The remainder of this chapter walks through the complete transformation—from individual JSON fields to a reusable operational dataset stored in Ceeview.

Step 1: Add the scoreConvert() Function

The blueprint on the previous page introduced the **scoreConvert()** function used by the Data Matrix Producer. In this step, you'll add the reusable function to the REST Monitor Editor so it's available when configuring the Data Matrix Builder.

Create the Function

Open the **Editor** from the Profile Designer toolbar and add the following function beneath the existing stripWorkgroup() function:

```
//Divide score reported by 100
def scoreConvert(score) {
    return (score.toBigDecimal() / 100).toInteger()  // ensure Integer is returned
}
```

After adding the function, the completed Editor should appear as shown below.

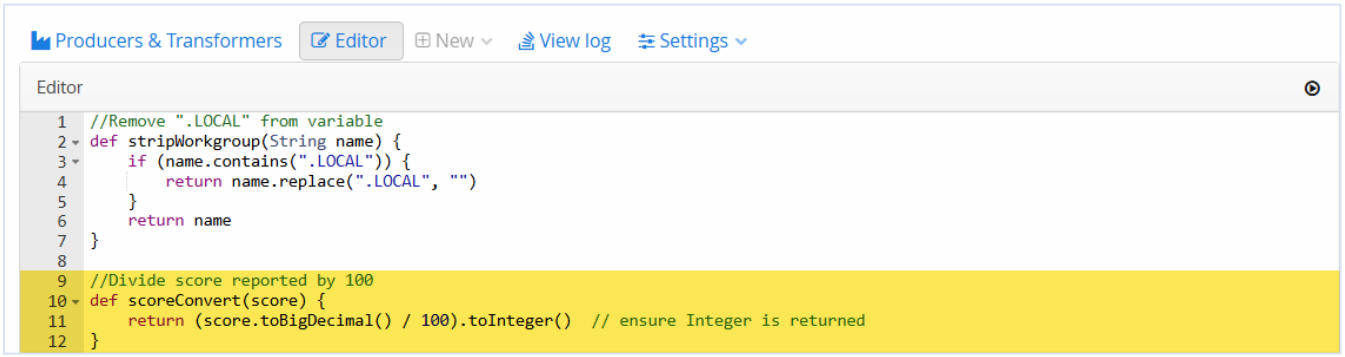


*Figure 1 · Editor containing the reusable **stripWorkgroup()** and **scoreConvert()** functions used throughout this REST Monitor configuration.*

Click Update to save the new Editor function before continuing.

With the **scoreConvert()** function in place, you're ready to configure the Data Matrix Builder and map the JSON fields introduced in the blueprint.

Step 2: Create Data Matrix Producer

In the Profile Designer select New → Producer → Data Matrix.

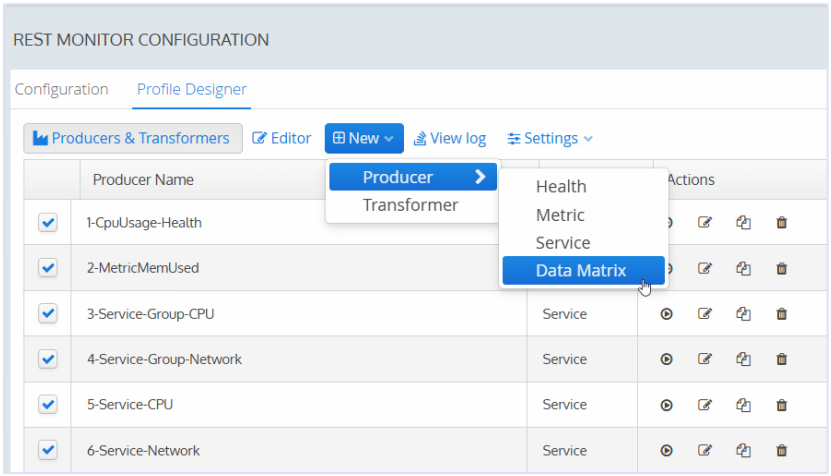


Figure 2 · Selecting New → Producer → Data Matrix from the menu

Step 3: Configure General Properties

Complete the **General Properties** tab as shown below. These settings define where the Data Matrix will be stored and how the producer iterates through the REST API response to create a reusable operational dataset.

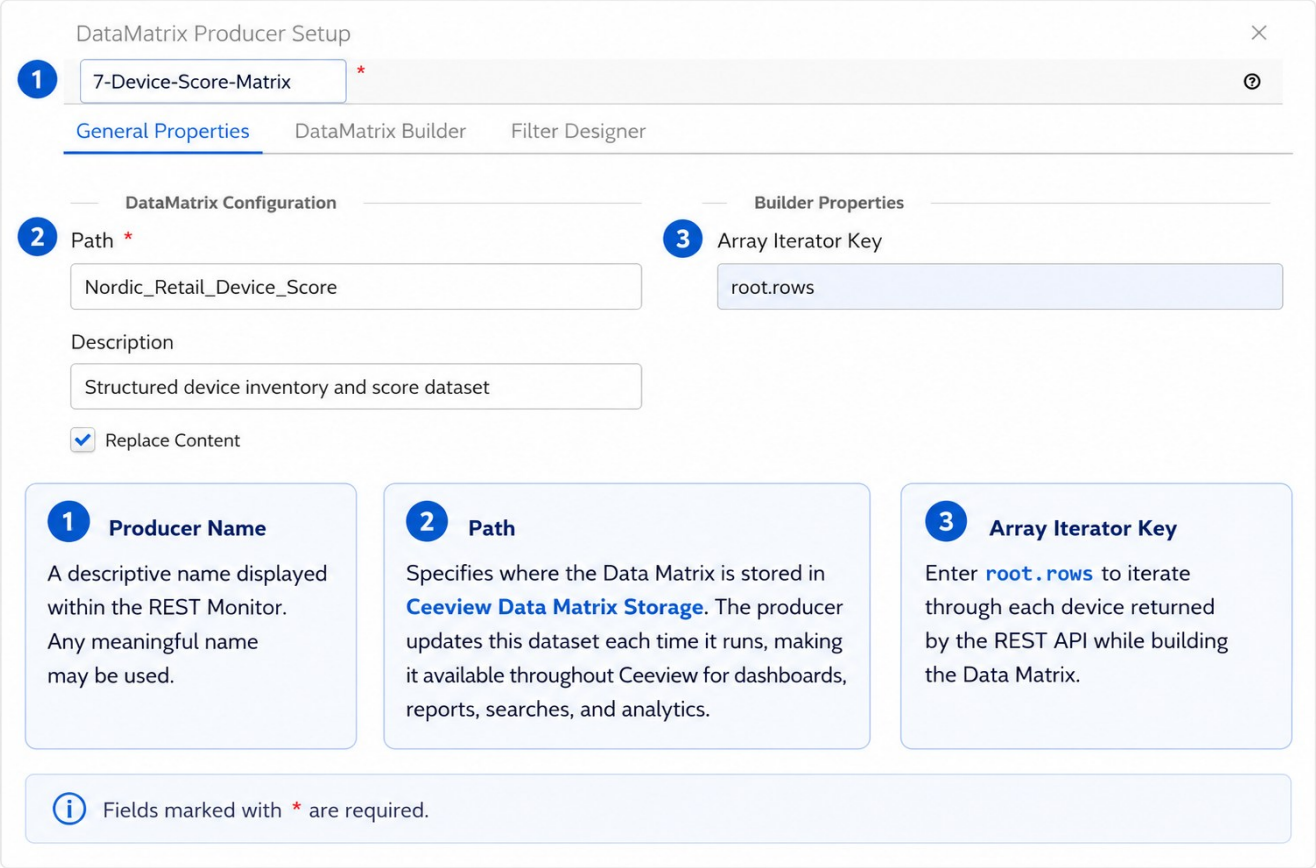


Figure 3 · Data Matrix Producer General Properties configuration. These settings define the Data Matrix location and how the producer processes each device returned by the Nordic Retail REST API.

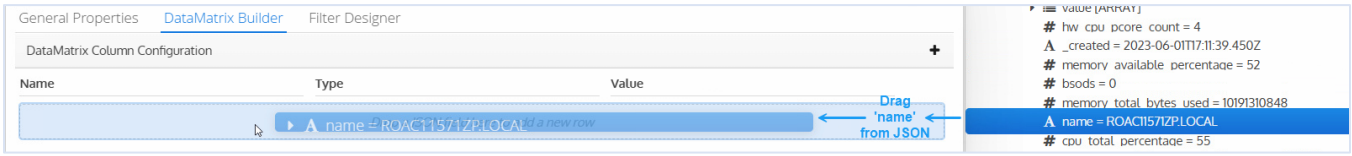
Next, configure the **Data Matrix Builder** to define the columns and transformation logic used to build the Data Matrix.

Step 4: Build the Data Matrix

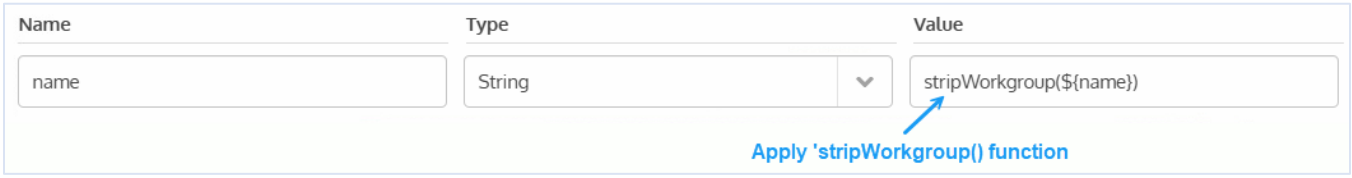
The Data Matrix Builder defines the columns that make up the operational dataset. Each row maps a JSON field into a Data Matrix column, with optional editor functions applied where data transformation is required.

Complete the following steps to build the Data Matrix columns.

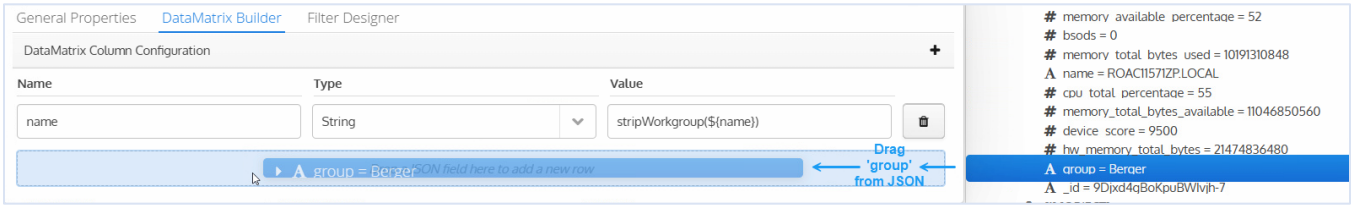
1. Drag **name** from the JSON Structure into the first Data Matrix row.



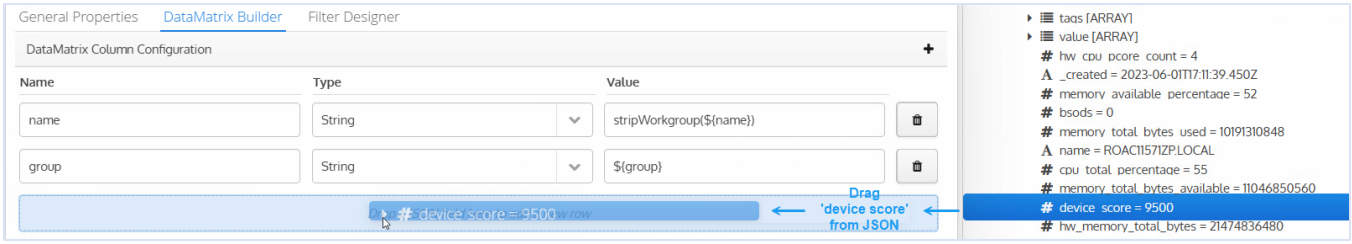
2. Apply **stripWorkgroup(${name})** to remove the *.LOCAL* suffix from the device name.



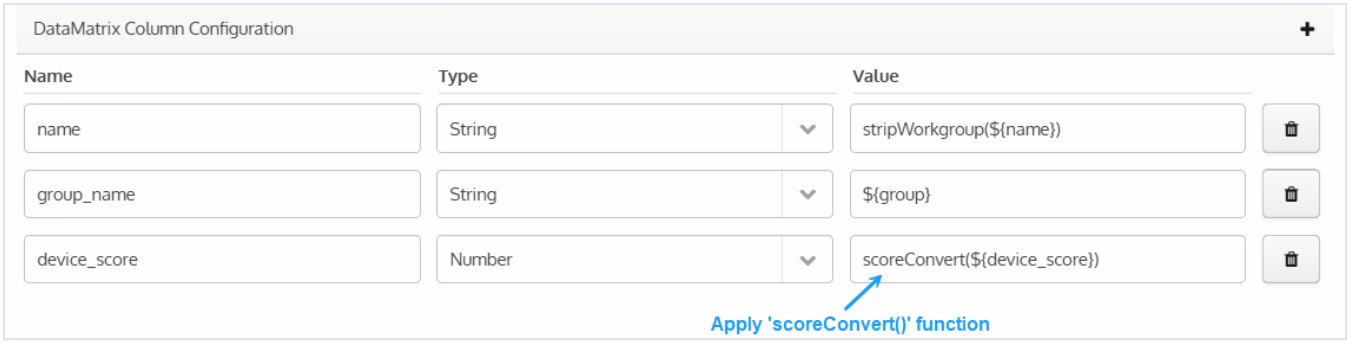
3. Drag **group** into the next Data Matrix row.



4. Drag **device_score** into the next Data Matrix row.



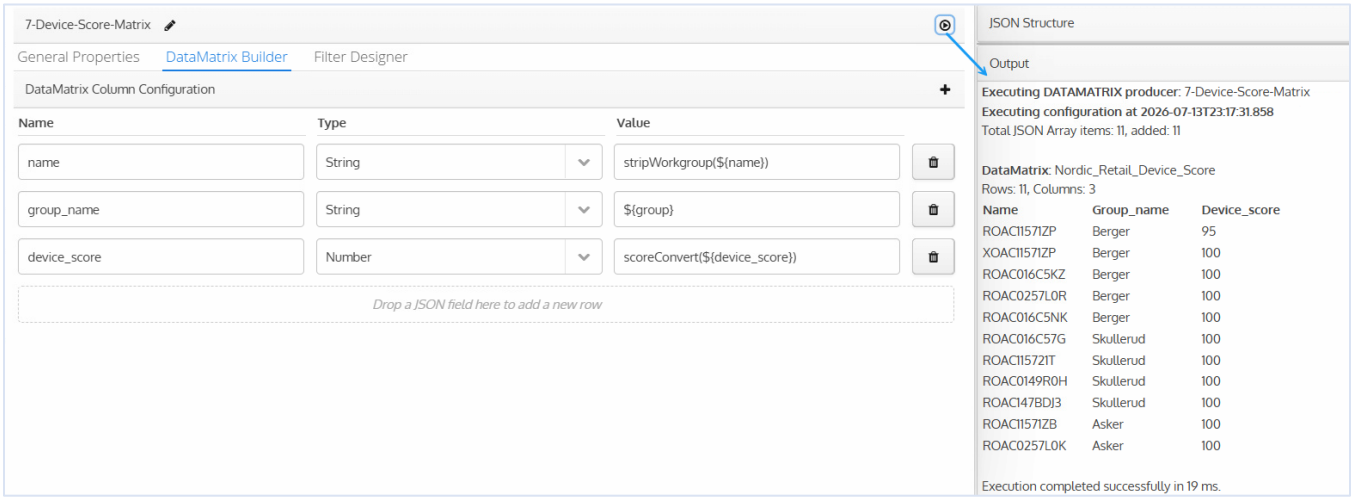
5. Apply **scoreConvert(${device_score})** to normalize the device score before storing it in the Data Matrix.



Step 5: Execute the DataMatrix Producer

Click **Run** in the upper-right corner of the Data Matrix Producer to execute the configuration using the current JSON response.

Review the Output panel to verify that the Data Matrix is created successfully before saving the Producer.



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

Step 6: Update the REST Monitor

Click Update to save the REST Monitor configuration and activate the new Data Matrix Producer. The producer will execute automatically each time the REST Monitor runs.

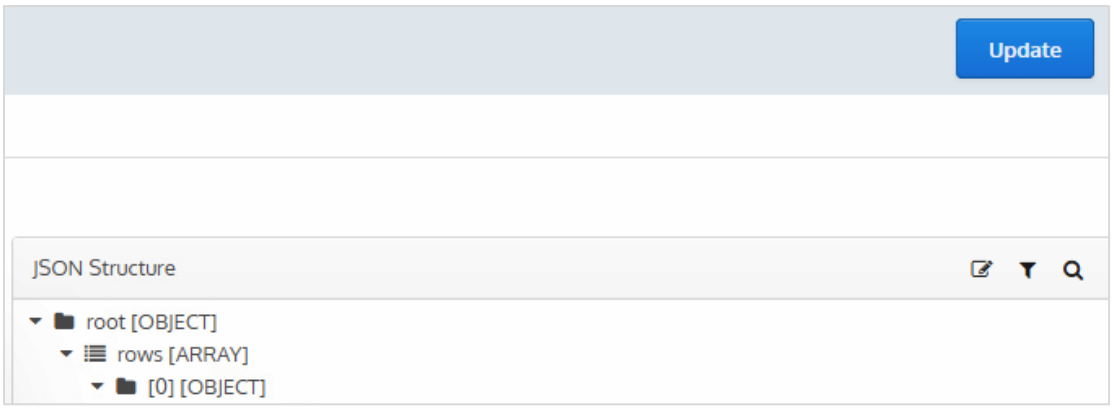


Figure 4 · Saving the REST Monitor activates the new Data Matrix Producer.

Step 7: Verify the Data Matrix

The Data Matrix has now been created and stored in Ceeview. In this step, you'll verify the resulting dataset by viewing it in **Data Matrix Storage**.

1. From the Ceeview main menu, open **Management → Integration → Sidekick Utilities**.
2. Select **Data Sources** from the left navigation.
3. Open the **DataMatrix Storage** tab.
4. Locate the **Nordic_Retail_Device_Score** entry.
5. Select the entry and click **View Table**.

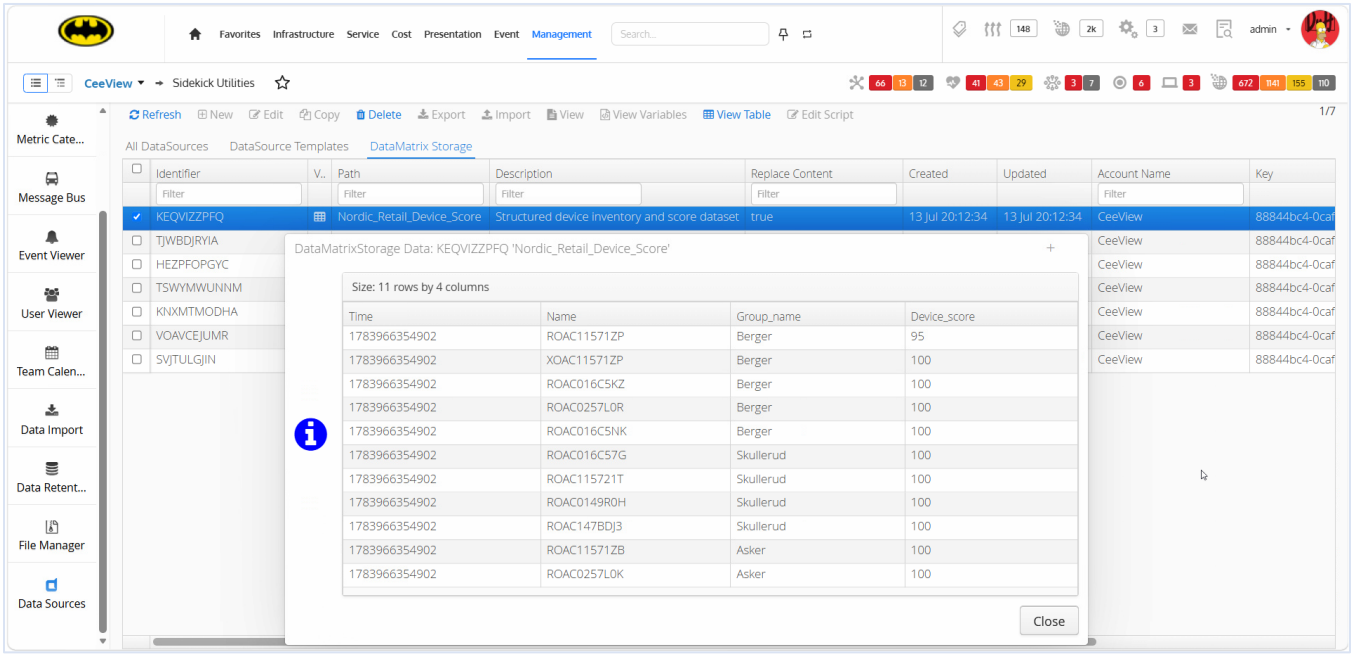


Figure 5 · Viewing the completed Nordic_Retail_Device_Score Data Matrix in Ceeview Data Matrix Storage.

The completed Data Matrix contains one row for each device returned by the REST API. The dataset includes the normalized device name, location, and converted device score that were configured throughout this tutorial.

The Data Matrix is now available within **Data Matrix Storage**. From here, it can be promoted to **All DataSources** where it becomes available for dashboards, reports, searches, and other Ceeview capabilities. That process is covered separately from this tutorial.

Step 8: Verify the REST Monitor Configuration

After updating the configuration, the Nordic Retail Example REST Monitor now includes the new **Data Matrix Producer** alongside the previously created Health, Metric, and Service Producers. Together, these producers transform the same REST API response into multiple native Ceeview objects.

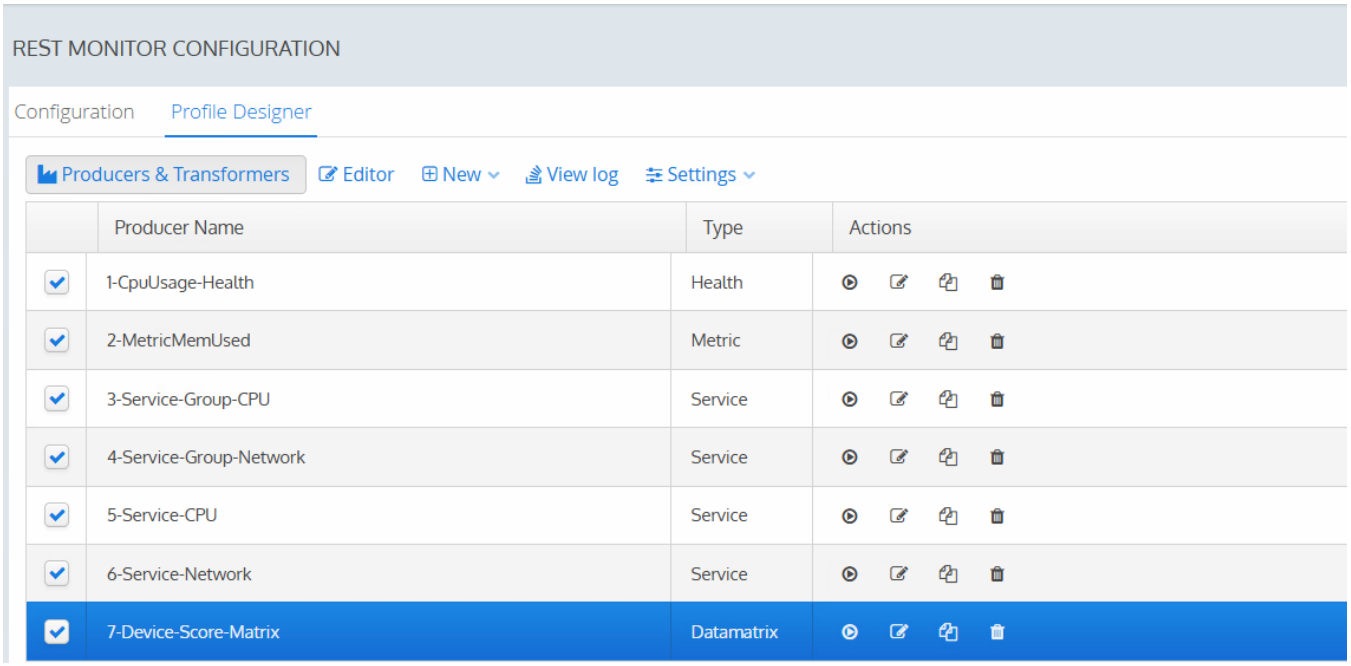
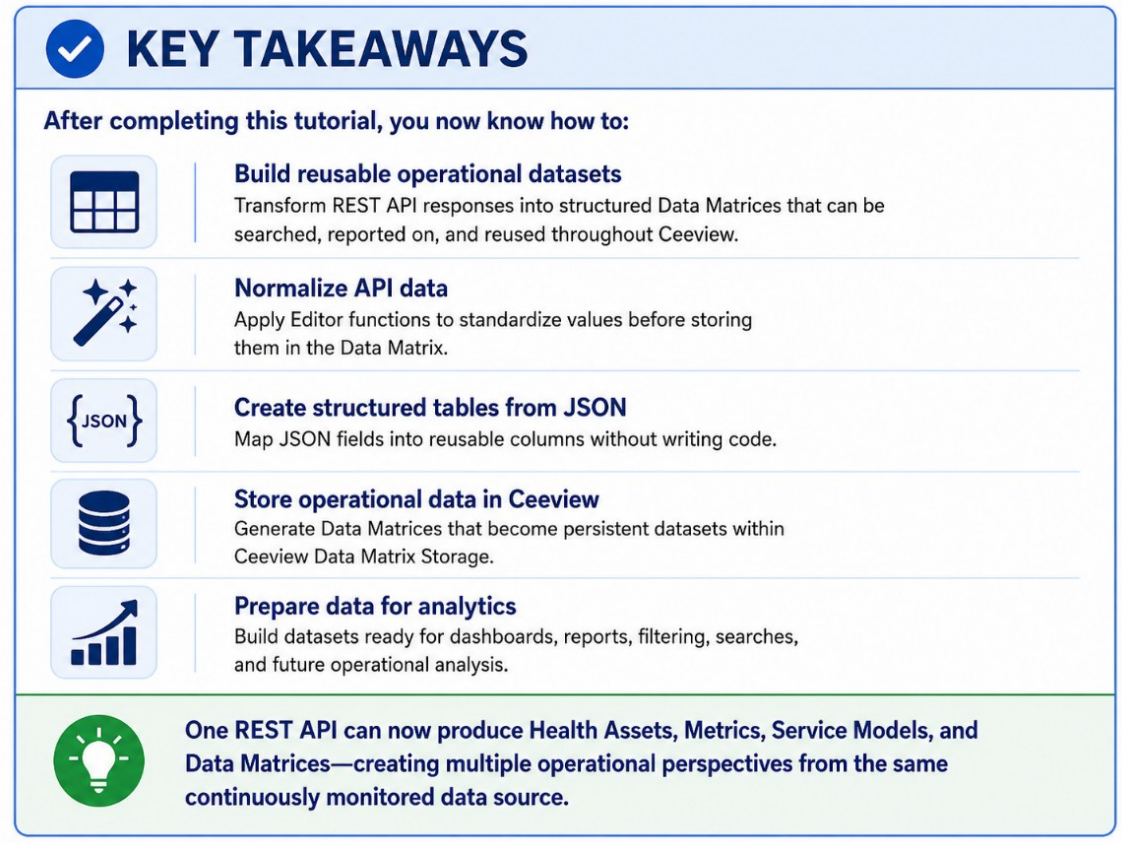
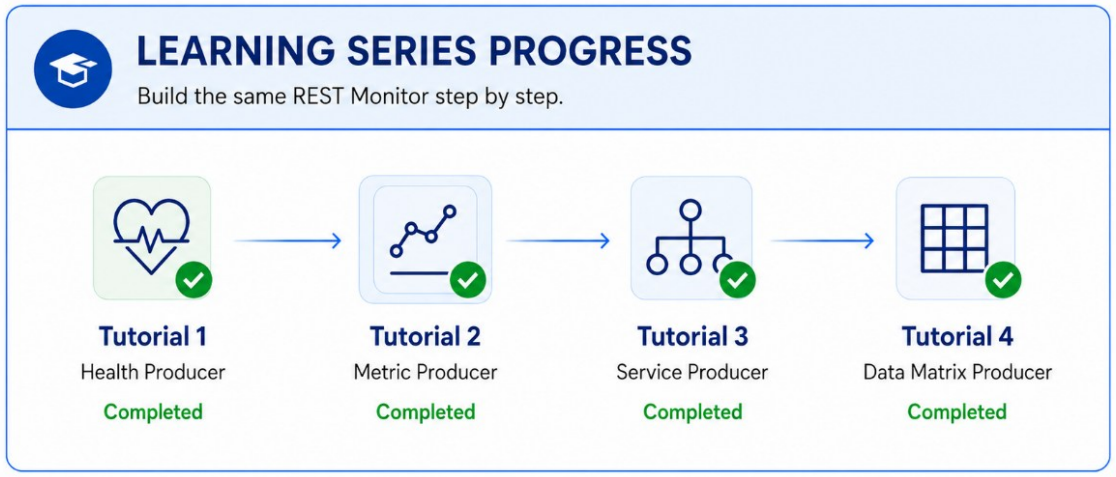


Figure 6 · Completed REST Monitor containing Health, Metric, Service, and Data Matrix Producers.

Complete the REST Monitor Learning Series

Congratulations! You've now completed the REST Monitor Learning Series. Starting from a single REST API, you've progressively built native Ceeview Health Assets, Metrics, Service Models, and reusable Data Matrices that work together to deliver rich operational visibility.





Across these four tutorials, you've progressively extended the same REST Monitor to build multiple operational views from a single continuously monitored REST API—without duplicating data collection or writing code.

A single REST API isn't just a source of data—it becomes a foundation for operational visibility.