



**Critical**  
manufacturing  
an ASM PT company

# Event Ingestion through IoT Data Platform

11.0

March 2026

## DOCUMENT ACCESS

Public

## DISCLAIMER

The contents of this document are under copyright of Critical Manufacturing S.A. it is released on condition that it shall not be copied in whole, in part or otherwise reproduced (whether by photographic, or any other method) and the contents therefore shall not be divulged to any person other than that of the addressee (save to other authorized offices of his organization having need to know such contents, for the purpose for which disclosure is made) without prior written consent of submitting company.

# Event Ingestion through IoT Data Platform

*Estimated time to read: 4 minutes*

Event ingestion is the process of obtaining and importing events for the immediate use or storage in a database. Events can be streamed in real time, where each event item is imported as the source emits it, or ingested in batches.

This document will provide a quick guide for the configuration of a mechanism of event ingestion using the IoT Data Platform capabilities of Critical Manufacturing.

## Overview

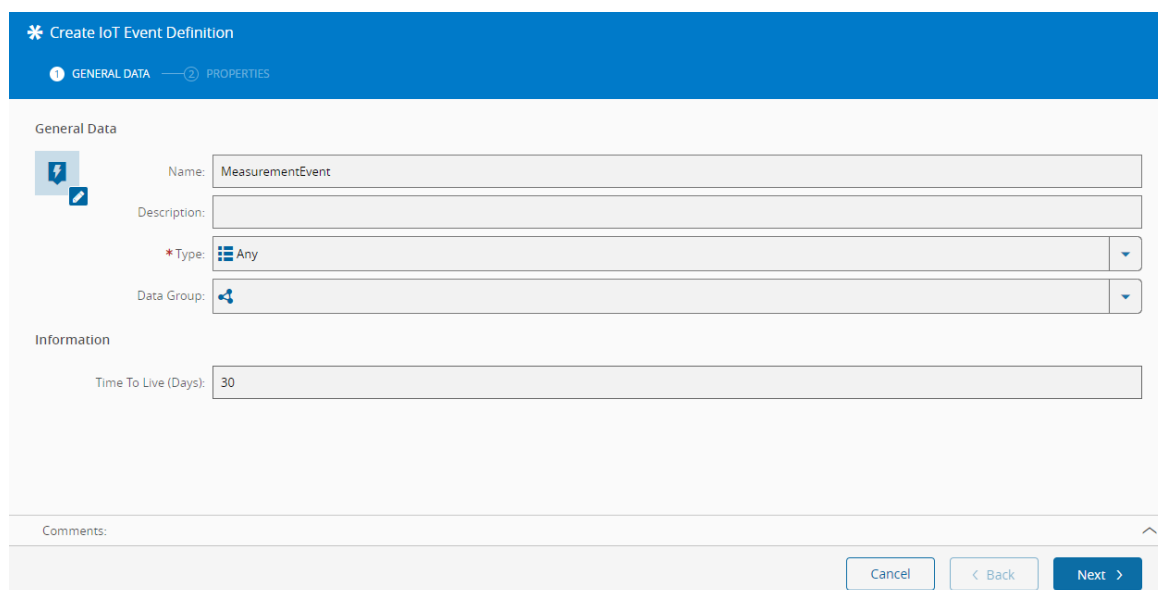
In this example we will define an event that receives values from different sensors on a shop floor. This event will have a specific structure and so we shall define it, make the system aware of it and able to parse the same event when received as an API call to the system. To create the appropriate structure to receive and adequately ingest and handle the events on Critical Manufacturing, a number of steps will have to be followed:

1. Create an **IoT Event Definition** to specify the event structure.
2. Create an **IoT Consumer Definition** to define which type of consumer will receive the event.
3. Ensure that the system ingests and operates on the event through an **IoT Consumer**.

The steps detailed above will create the objects in the system that will handle the events and the payload contained therein. This way, you can start from there:

## Creating an IoT Event Definition

In the Business Data section of the main menu, navigate to the **IoT Event Definition** tile and create a new entry. Provide a name for the **IoT Event Definition**. Bear in mind that the name will identify the Event throughout the Data Platform infrastructure so it should be a meaningful name for later reference. An example of the values can be seen below:



**Create IoT Event Definition**

1 GENERAL DATA — 2 PROPERTIES

**General Data**

Name: MeasurementEvent

Description:

\* Type: Any

Data Group:

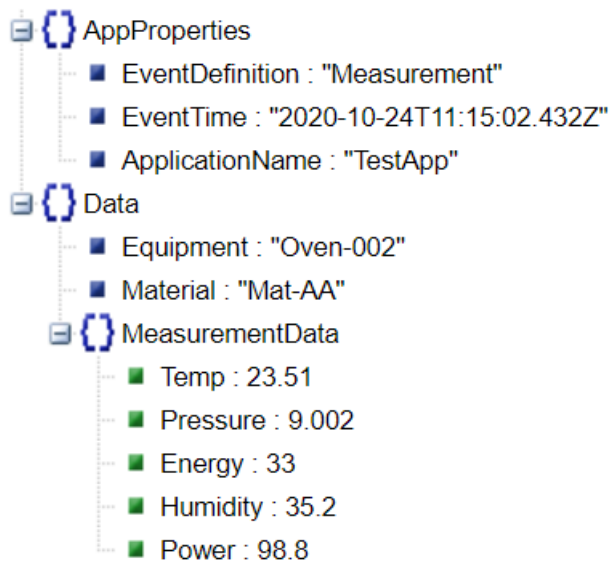
**Information**

Time To Live (Days): 30

Comments:

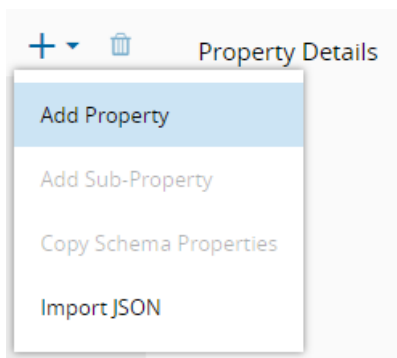
Cancel < Back Next >

The event to be created has the following structure with sample values:



`AppProperties` is a field included by the Kafka host that is running and receives the event, while `Data` is the actual event payload.

Adding properties to the event definition can be done manually or by importing a JSON structure such as the one shown above. The system will infer the data types from the actual values and generate a property for each of the fields.





Several properties are essential for the proper functioning of the event ingestion, so specify for each property:

- **Property name** – name that matches the property in the Event JSON document.
- **Array** – identifies this property as an array of objects.
- **Mandatory** – identifies if the property is mandatory. If true, this information will be used to validate all the arriving events.
- **Indexed** – if this value is defined, the fields will be automatically selected in the creation of the **IoT Consumer**.
- **Data Type** – the data type of the property. Mandatory, an enumeration. Will be mapped to the following JSON types (in parentheses).
- **Default Value** – specifies the default value of the property. If not present in the received event, this value will be set by the event validation process.
- **Shared** – a specific **IoT Schema** can be selected to be matched to a JSON structure and later shared as a separate object in the system.

**Info**

For more information, see [Create IoT Event Definition](#) in the User Guide.

Importing from a sample JSON structure can be simple as well:

PROPERTIES		TYPE	MEASUREMENTDATA (MEASUREMENTDATA)	IOTSCHEMA
Equipment (Equipment)		String	Temp (Temp)	Integer
Material (Material)		String	Pressure (Pressure)	Decimal
MeasurementData (MeasurementData)	 		Energy (Energy)	Integer
			Humidity (Humidity)	Decimal
			Power (Power)	Decimal

After creating the properties manually or importing from a JSON file, the properties will be generated and the **IoT Event Definition** can be created.

Import JSON Schema

Template

Content Type:

\* Filename:  Upload

\* Template: 

```

1
2   "Equipment": "Oven-002",
3   "Material": "Mat-AA",
4   "MeasurementData": {
5     "Temp": 23432,
6     "Pressure": 9.002,
7     "Energy": 33,
8     "Humidity": 35.2,
9     "Power": 98.8
10  }
11
```

Cancel Import

Upon original import, and if the **Shared** option is not defined above, a new **IoT Schema** is also created and associated in a one-to-one relationship with this particular event definition, and it could then be used in other definitions.

MeasurementEvent (Active)

DETAILS

IoT Schema

Name: MeasurementEvent  
 Description:  
 Universal State: Active  
 Shared: X No

PROPERTY

Properties

NAME	FRIENDLY NAME	DESCRIPTION	DATA TYPE	IS MANDATORY	IS ARRAY	INDEXED	DETAILS	
Equipment	Equipment		String	X	X	✓		
Material	Material		String	X	X	✓		
MeasurementData	MeasurementData		IoTSchema	X	X	X		5 >



# Legal Information

## **Disclaimer**

The information contained in this document represents the current view of Critical Manufacturing on the issues discussed as of the date of publication. Because Critical Manufacturing must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Critical Manufacturing, and Critical Manufacturing cannot guarantee the accuracy of any information presented after the date of publication. This document is for informational purposes only.

Critical Manufacturing makes no warranties, express, implied or statutory, as to the information herein contained.

## **Confidentiality Notice**

All materials and information included herein are being provided by Critical Manufacturing to its Customer solely for Customer internal use for its business purposes. Critical Manufacturing retains all rights, titles, interests in and copyrights to the materials and information herein. The materials and information contained herein constitute confidential information of Critical Manufacturing and the Customer must not disclose or transfer by any means any of these materials or information, whether total or partial, to any third party without the prior explicit consent by Critical Manufacturing.

## **Copyright Information**

All title and copyrights in and to the Software (including but not limited to any source code, binaries, designs, specifications, models, documents, layouts, images, photographs, animations, video, audio, music, text incorporated into the Software), the accompanying printed materials, and any copies of the Software, and any trademarks or service marks of Critical Manufacturing are owned by Critical Manufacturing unless explicitly stated otherwise. All title and intellectual property rights in and to the content that may be accessed through use of the Software is the property of the respective content owner and is protected by applicable copyright or other intellectual property laws and treaties.

## **Trademark Information**

Critical Manufacturing is a registered trademark of Critical Manufacturing.

All other trademarks are property of their respective owners.