**Order Processing Error Dashboard in Grafana**

**1. Objective**

The purpose of this dashboard is to monitor **order lifecycle events** (created, completed, failed) and drill down into **pre-order and post-order processing errors** with detailed reasons.

This simulates an observability setup where business transactions (orders) can be tracked, and failures can be analyzed with context.

**2. Dataset**

We used a **custom dataset** representing order events with fields such as financialAcc, statusCode, orderId, activityId, and reasonDescription.

**Dataset format: JSON**

**Columns:**

* time – Event timestamp
* financialAcc – Financial account identifier
* statusCode – Error classification (PreOrderProcessingError, PostOrderProcessingError)
* orderId – Unique order identifier
* activityId – Activity identifier for sub-process
* subStatusCode – Technical error type (e.g., CreditCardFailed, LicenseUploadFailed)
* status – Final status (failed, created, completed)
* errorPhase – Phase where error occurred (pre, post)
* count – Event count (used for aggregations)
* reasonDescription – Human-readable explanation of the error

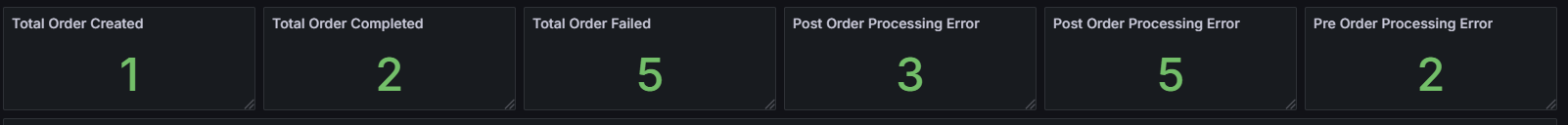
👉 This dataset was uploaded into Grafana Cloud via the **JSON data source plugin**.

**3. Dashboard Overview**

The dashboard is named **Order Processing Error Dashboard** and consists of the following panels:

**🔹 KPI Stat Panels**

1. **Total Order Created**
   * Query filters rows where status = created
   * Aggregation: count
2. **Total Order Completed**
   * Query filters rows where status = completed
   * Aggregation: count
3. **Total Order Failed**
   * Query filters rows where status = failed
   * Aggregation: count
4. **Post Order Processing Error (Count)**
   * Filters statusCode = PostOrderProcessingError
   * Aggregation: count
5. **Pre Order Processing Error (Count)**
   * Filters statusCode = PreOrderProcessingError
   * Aggregation: count



**Finance Number Panel (Table)**

* Visualization: **Table**
* Shows all orders with columns:  
  financialAcc, statusCode, orderId, activityId, subStatusCode
* Transformation: **Organize fields** to control display order
* **Data Link**: Clicking an orderId sends variables (orderId, activityId) to the next panel ( /d/${\_\_dashboard.uid}/${\_\_dashboard.name}?var-orderId=${\_\_data.fields.orderId}&var-activityId=${\_\_data.fields.activityId} )

A screenshot of a computer

AI-generated content may be incorrect.

**Pre and Post Processing Error Panel (Table)**

* Visualization: **Table**
* Filters dynamically based on selected orderId and activityId (from Finance Number Panel)
* Transformation: **Filter by values** →
  + orderId = $orderId
  + activityId = $activityId
* Displays columns:  
  orderId, count, financialAcc, status, statusCode, subStatusCode, reasonDescription
* Example output when clicking on ORD-10003:

A screenshot of a computer

AI-generated content may be incorrect.

**4. Drilldown Interaction**

* **Step 1:** User clicks a row in **Finance Number Panel**.
* **Step 2:** Grafana passes selected orderId and activityId as variables.
* **Step 3:** **Pre and Post Processing Error Panel** filters its query and displays only relevant errors.
* **Result:** A connected workflow for investigating order-specific failures.

A screenshot of a computer

AI-generated content may be incorrect.

**5. summary**

1. **Transformations in Grafana**
   * Used **Filter by values** and **Organize fields** extensively to shape tabular data.
2. **Variables & Drilldowns**
   * Created **dashboard variables** (orderId, activityId) passed via **Data Links** between panels.
3. **Table vs Stat Panels**
   * Stat panels are used for KPIs (single numeric values).
   * Table panels are better for showing contextual details (like error reasons).
4. **Reason Descriptions**
   * Adding a reasonDescription column improved **observability** by making technical error codes human-readable.