**Passing Parameters from ADF to Databricks using dbutils**

**Objective**

* Capture those parameters in the notebook using dbutils.widgets.
* Use parameters to make notebook logic dynamic (e.g., different dataset paths, filters, dates).

**Step 1 – Prepare a Notebook in Databricks**

Create a new notebook: **OrdersProcessingNotebook**

# Define widgets to capture parameters from ADF

dbutils.widgets.text("input\_path", "/mnt/raw/orders.csv", "Input Path")

dbutils.widgets.text("output\_path", "/mnt/curated/sales", "Output Path")

dbutils.widgets.text("run\_date", "2025-09-12", "Run Date")

# Get widget values

input\_path = dbutils.widgets.get("input\_path")

output\_path = dbutils.widgets.get("output\_path")

run\_date = dbutils.widgets.get("run\_date")

print(f"Notebook Parameters:\n Input: {input\_path}\n Output: {output\_path}\n Run Date: {run\_date}")

# Example: Read and filter orders

orders = spark.read.option("header", True).csv(input\_path)

filtered = orders.filter(orders.OrderDate == run\_date)

# Write output

filtered.write.mode("overwrite").parquet(output\_path)

dbutils.notebook.exit(f"Processed {filtered.count()} records for {run\_date}")

**Step 2 – Setup Linked Service in ADF**

1. In **ADF Studio → Manage → Linked Services**:
   * Add **Azure Databricks** linked service.
   * Provide Workspace URL, Authentication (MSI, PAT, or Key Vault), Cluster.

**Step 3 – Build ADF Pipeline**

1. In ADF, create a new **Pipeline** → drag **Databricks Notebook activity**.
2. Configure:
   * Linked service = your Databricks workspace.
   * Notebook path = /Shared/OrdersProcessingNotebook.
3. In **Base Parameters**, define values to pass:

| **Name** | **Value** |
| --- | --- |
| input\_path | abfss://raw@<storageaccount>.dfs.core.windows.net/orders.csv |
| output\_path | abfss://curated@<storageaccount>.dfs.core.windows.net/sales |
| run\_date | @pipeline().parameters.RunDate |

**Step 4 – Add Pipeline Parameters in ADF**

1. Go to Pipeline → Parameters.
   * Add RunDate (string).
2. Default value → 2025-09-12 (or use dynamic system variable).

Example dynamic value:

@formatDateTime(pipeline().TriggerTime, 'yyyy-MM-dd')

**Step 5 – Trigger the Pipeline**

1. Debug or Publish → Trigger now.
2. Check **Monitor** tab for pipeline execution.
3. Notebook output in Databricks should show:

Notebook Parameters:

Input: abfss://raw@<storageaccount>.dfs.core.windows.net/orders.csv

Output: abfss://curated@<storageaccount>.dfs.core.windows.net/sales

Run Date: 2025-09-12

**Step 6 – Validate Output**

* Go to curated/sales folder in ADLS → verify parquet file.
* The file should only contain orders where OrderDate = RunDate.