

# Databricks Initiator for Alchemer Workflow

## Overview

**Databricks** is a unified data and AI platform that helps organizations build, deploy, and manage data engineering and analytics workloads. Organizations use Databricks to query, analyze, and process large-scale data stored in data warehouses and lakehouses.

The Alchemer integration with Databricks supports automated workflow initiation based on table rows. The Alchemer → Databricks Workflow Initiator allows Alchemer to query a Databricks table, extract each row, and trigger a separate workflow execution for every row—fully automating bulk operations, data processing, and downstream integrations without manual intervention.

## Common uses for the Alchemer Databricks integration

- Send one workflow request per Databricks table row into downstream systems
- Automate batch processing of data from Databricks warehouses or lakehouses
- Drive workflow branching or routing using Databricks column values
- Trigger external API calls or business logic for each data row
- Reduce manual processing of large Databricks datasets
- Keep Alchemer workflows synchronized with data changes in Databricks

## What can the Alchemer Databricks integration do?

- [Start workflow from Databricks](#)

## You will need

- Databricks OAuth authentication. [More details in the authentication how-to guide.](#)
- An Alchemer plan that includes integrations with Workflow and the Integration Manager permission enabled.
  - [Contact us](#) if you are unsure if your plan includes integrations.

---

## Setup Alchemer Databricks integration in Workflow

### [Databricks | Start workflow from Databricks](#)

#### You will need:

- Databricks OAuth authentication. [More details in the authentication how-to guide.](#)
- A Databricks SQL warehouse that is running or can be started
- Access to a Databricks catalog, schema, and table with data

### Configure the action

1. Open your workflow in **Workflow Builder**.
2. On the **Select Initiator** pop up select the **Databricks** initiator.
3. Select **Databricks | Start workflow from Databricks**.
4. **Databricks | Authentication**: Select an existing authentication or [create a new authentication](#).
5. **Databricks | Select warehouse**: Select the SQL warehouse from the dropdown. The warehouse must be accessible with your authentication credentials.
6. **Databricks | Select catalog**: Select the catalog from the dropdown that contains your table. Databricks uses a three-level namespace: catalog > schema > table.
7. **Databricks | Select schema**: Select the schema (also called database) from the dropdown within your selected catalog.
8. **Databricks | Select table**: Select the table from the dropdown within your selected schema.
9. **Databricks | Select criteria**: Add filtering criteria to select specific rows. You will select the Databricks column on the left and the value to match on the right. Multiple fields are combined with the AND operator. Optionally set a limit to how many rows will execute. The default is 100.
10. **Databricks | Schedule for runs**: Set the schedule for how frequently you want to query the Databricks table and trigger workflows.
11. Save the action.

### Status codes

- 200: Successfully triggered the workflow for each table row
- 400: The external integration returned an error

---

## Testing and Troubleshooting

### Testing and Validation

#### How to test

- Trigger the workflow and monitor individual runs in the **Monitor** tab.
  - Click on each run to view metadata outputs for each row sent to the workflow.
- Verify that the expected number of workflow runs were created based on your table row count and row limit.

#### How to verify results

- Check the Monitor tab to confirm one workflow run per table row was created.
- Verify that merge codes in your workflow contain the correct values from the Databricks table

columns.

- Set a send email action to yourself with workflow metadata included to validate data accuracy.
- 

## Monitoring Integration Activity

### Where to find logs

- Go to **Monitor**.
- Check the individual workflow runs and steps.

### What logs display

- Inputs received from Databricks table (one set per row)
  - Workflow execution status and output
  - Timestamp of each workflow trigger
- 

## Troubleshooting

### Authentication issues

- Expired or invalid personal access token
- Missing permissions to access the selected warehouse or table
- Warehouse is suspended or unavailable

### Warehouse or table issues

- Warehouse fails to start or connect
- Selected table does not exist or has been deleted
- Insufficient data in the selected table
- Column names contain special characters or are not accessible

### Query or API errors

- Timeout waiting for query results
  - Incorrect catalog, schema, or table selection
  - Row limit is set too high, causing performance issues
- 

## FAQs

### What permissions do I need?

Integration Manager in Alchemer and API access in Databricks with permissions to access the selected warehouse and table.

### When does the integration run?

On the set schedule. Each scheduled run queries the table and triggers a separate workflow execution for each row.

### Can I filter the rows from my Databricks table?

Yes. Use the "Select criteria" step to add filtering conditions. You can select the Databricks column and specify the value to match. Multiple criteria are combined with the AND operator.

### Why isn't my workflow triggering?

Check the Monitor tab for authentication errors, warehouse connection issues, table access

problems, or empty query results.

**What if I need additional functionality?**

Contact Alchemer Support for enhancement requests.

Related Articles

---