Node Guide: Send Table Data

Overview

The **Send Table Data** node is used to **display structured JSON data in a tabular format**. This is helpful for presenting multi-column data sets such as API results, processed records, reports, etc., in a user-friendly and readable layout.

What This Node Does

- Renders a **data table** with customizable columns and rows

- Displays an optional **summary message** above the table

- Both the summary and table content are pulled **from variables** created in the workflow

Configuration Details

1. Summary

- Select a **variable** (from the dropdown) that holds a summary or title text to appear above the table

- This acts as an introductory line or context for the displayed data

2. Data (TableData)

- Select a **JSON-formatted variable** from the dropdown that contains the data structure for the table

- JSON must follow the below schema:

json
CopyEdit

```json
{
  "TableData": {
    "Columns": [
      {
        "name": "columnKey",
        "label": "Column Label",
        "field": "columnKey",
        "align": "left",
        "sortable": true,
        "columnType": "text"
      }
    ],
    "Rows": [
      {
        "columnKey": "Value 1",
        "anotherColumn": "Value 2"
      }
    ]
  }
}
```

Column Fields Explained:

- **name / field**: Key in the data row (must match exactly)

- **label**: Header name shown in the table

- **align**: left or right

- **sortable**: true or false

- **columnType**: e.g., text, number, date (defines how content is rendered)

Inputs

- **Summary**: Variable with plain text summary

- **Data**: Variable containing JSON table data in the required structure

Outputs

- No downstream data output

- Displays a **formatted table** in the interface with headers and rows

---

When to Use

Use this node when you need to:

- Display structured data (e.g., list of users, transaction logs, status reports)

- Show tabular results from a GET API or data aggregation

- Provide a readable summary with visual clarity

Example Use Case

Scenario: Display List of Pending Orders

You fetched order data from an API and stored it in a variable named `pending_orders_table`.

Configuration

- **Summary Variable**: `orders_summary` (e.g., "List of Orders Pending Delivery")

- **Data Variable**: `pending_orders_table`

```
{
  "TableData": {
    "Columns": [
      { "name": "orderId", "label": "Order ID", "field": "orderId", "align": "left", "sortable": true, "columnType": "text" },
      { "name": "customer", "label": "Customer", "field": "customer", "align": "left", "sortable": false, "columnType": "text" },
```

```
    { "name": "amount", "label": "Amount", "field": "amount", "align": "right", "sortable": true,
"columnType": "number" }
  ],
  "Rows": [
    { "orderId": "ORD123", "customer": "John Doe", "amount": "$150" },
    { "orderId": "ORD124", "customer": "Jane Smith", "amount": "$200" }
  ]
 }
}
```

Summary

The **Send Table Data** node is ideal for presenting dynamic, variable-driven data in a clean, tabular view. It works great for reports, summaries, and any structured display requirements within your agent workflows.