

Node Guide: User Input Form

Overview

The **User Input Form** node allows you to present a structured, customizable form to the user and **capture their input into variables**. It supports a wide range of field types and provides powerful configuration for complex inputs like dropdowns, tables, and checkboxes.

What It Does

- Renders a fully configurable form interface to the user
 - Captures multiple inputs into mapped variables
 - Supports advanced fields like tables, dropdowns, files, and more
 - Offers preview and JSON import functionality for testing and configuration
-

Configuration

1. Form Name

- Can be **typed manually** or selected from a **variable** dropdown

2. Submit Button Label

- Can be **typed manually** or selected from a **variable** dropdown

3. Form Fields

You can configure multiple fields in the form. Each field must have:

- A **key/label**

- A **field type**
 - A **variable** to store the user's input
-

Supported Field Types and Configuration

Basic Field Types

Type	Description	Configuration
<code>string</code>	Single line text input	Key + Variable
<code>text</code>	Multi-line input	Key + Variable
<code>long</code>	Numeric input	Key + Variable
<code>date</code>	Date picker (calendar input)	Key + Variable
<code>file</code>	File upload field	Key + Variable

Selectable Field Types

`dropdown`, `checkbox`, `radio`

These types allow the user to select one or more options.

Option Source:

- **Static:** Manually define the options (label and value)
- **Variable:** Use a variable whose value holds an array of options

Table Field Type

This is the most advanced and flexible field. It supports **3 configurations**:

1. Manual Columns & Rows

- You define both columns and rows manually
 - Headers (column names) and rows are typed in manually
 - You can **import JSON** to create the table quickly
-

2. Static Columns + Rows from Variable

- Columns are **manually configured**
- Rows are fetched from a **variable** (dropdown)

You can:

- Define column headers and keys
 - Bind the row data to a JSON array stored in a variable
-

3. Columns & Rows from Variable

- Both column definitions and row data are **fetched from variables**
 - Useful for fully dynamic tables
-

Additional Table Configurations

Import JSON: You can paste valid JSON to generate column/row structure

- **Row Selection Type:**
 - **Single Selection:** Allows user to select only one row
 - **Multiple Selection:** Allows user to select multiple rows
 - **Map To:**
 - **FullObject:** The entire row (as a JSON object) is stored in the variable
 - **Key Field Only:** Only the value of a specific field from the row is stored
 - **Preview Table:** You can preview the table and test the setup
-

Inputs

- No initial input required
 - Triggers form UI for the user
-

Outputs

- Each form field's value is stored in the **configured variable**
 - Flow resumes after user submits the form
-

When to Use

Use this node to:

- Collect structured input from users (e.g., forms, surveys, onboarding)
 - Let users select items from a dynamic table
 - Capture file uploads, dates, or selection inputs in a clean UI
-

Example Use Case

Scenario: Collect user profile information

- **Form Name:** `{{profile_form_title}}`
- **Submit Button:** `Save Profile`

Fields:

- `Name` (string) → `user_name`
- `Email` (string) → `user_email`
- `DOB` (date) → `user_dob`
- `Hobbies` (checkbox, static options) → `user_hobbies`
- `Uploaded Resume` (file) → `resume_file`
- `Projects Table` (table with variable rows, static columns) → `selected_projects`

Summary

The **User Input Form** node is a powerful and flexible way to collect structured input in your agent workflow. With support for advanced fields like tables and selection types, it offers enterprise-level configurability while staying user-friendly.

Let me know if you need example JSON snippets or UI screenshots added!