



SAP Build Process Automation

Generated on: 2024-02-26 21:08:14 GMT+0000

SAP Build Process Automation | Cloud

PUBLIC

Original content: https://help.sap.com/docs/PROCESS_AUTOMATION/a331c4ef0a9d48a89c779fd449c022e7?locale=en-US&state=PRODUCTION&version=Cloud

Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

For more information, please visit the <https://help.sap.com/docs/disclaimer>.

Use SAP Build Process Automation

SAP Build Process Automation is an SAP BTP service that allows you to create, run, automate, and monitor your business processes on one interface using low-code/no-code capabilities.

You can subscribe to SAP Build Process Automation using either the standard (paid) or free plan, with active subscriptions added to your SAP BTP subaccount.

Follow our interactive guided boosters to **build** applications or use different platform services and features.

SAP Build Process X 🔍 All ▼

Extension Suite - Digital Process Automation (2)

»»»

Set up account for SAP Build Process Automation (Free)

Automated setup for SAP Build Process Automation in your account.

[Start](#)

»»»

Set up account for SAP Build Process Automation

Automated setup for SAP Build Process Automation in your account.

[Start](#)

For more information about subscribing to SAP Build Process Automation and accessing the service, see [Initial Setup](#).

And for more information about business processes and the role of citizen developers, take our SAP Learning journey:

[Utilize Low-Code/No-Code Applications and Automations for Citizen Developers](#).

Already subscribed to SAP Build Process Automation?

Once subscribed to SAP Build Process Automation, you can create, manage, and deploy projects using the Lobby.

For more information, see [Create and Manage Projects](#).

Create and Manage Projects

After subscribing to SAP Build Process Automation, you create and manage projects in the lobby.

This lobby includes projects you've created, those shared with you, or that you've imported from external sources or the store.

The screenshot shows the SAP Build interface. On the left is a sidebar with navigation links: Lobby, Connectors (Actions, Automation SDK), Store, Monitoring, and Control Tower. The main area has a purple header "Welcome to SAP Build" with the sub-header "Create apps, automate processes, and build business sites using productivity or no-code tools." Below this is a "Quick Start" section with three cards: "Access our SAP Build Learning Journeys" (Learning), "Create a Change and Innovation Approval Process" (Template), and "Create an Invoice Approval Process" (Template). The main content area shows "All Projects (2)" with a table:

Name	Versions	Type	Last Accessed	Members	Options
Supplier Approval and Creation in SAP S/4HANA Supplier Approval in SAP S/4HANA		Process Automation	Oct 26, 3:02 pm	Me	...
Job Offers Approvals Automates approval process of job offers pending in SAP SuccessFactors system.		Process Automation	Oct 19, 12:22 pm	Me	...

i Note

User principles are case insensitive. For example, “user@mail.com” is treated as the same entity as “USER@mAIL.cOM”. Therefore, sharing a Project/Package with them is transparent.

Project Types

With an SAP Build Process Automation subscription, you can create and manage the following project types:

- **Business Process Project** - Create, deploy, automate, and run digital business process by configuring process artifacts. Examples of business process projects include: Investment requests, invoice approvals, and sales order approvals.
- **Actions Project** - Embed external actions and capabilities into your business process projects by uploading an open API specification file in JSON format. Actions projects allow external systems and solutions to communicate with SAP Build Process Automation.

Create a Business Process Project

When creating a business process project, choose to start from scratch. This creates an empty project, allowing you to define your own content.

The screenshot shows the SAP Build "Create" dialog. At the top is a search bar "Search by Project name & description" with a magnifying glass icon. Next to it is a blue "Create" button with a red border, which is highlighted. To the right of the button are icons for refresh, undo, redo, and download. Below the search bar is a table with two columns: "Last Accessed" and "Members". Under "Last Accessed" is the date "Nov 11, 4:54 pm". Under "Members" is the text "Everyone" with a bookmark icon and three dots for more options.

Alternatively, you can take advantage of the **Quick Start** project options. This creates the project container and prompts you to add the relevant steps for the project.

For more information about business process projects and skills, see [Business Process Projects](#)

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

Create Actions Project

To create an actions project, choose **Connectors > Actions > Create > Choose an API Source** and then enter the necessary details.

For more information about Actions project, see [Create an Action Project](#).

Manage Existing Projects

For existing projects, choose **More Options** to see further project management abilities. These abilities depend on the permissions you hold and can include releasing, publishing, renaming, sharing, and deleting your projects.

Last Accessed	Members
Nov 8, 1:13 pm	2 members
Oct 27, 5:41 pm	Everyone

More Options

 Release

 Publish to Library

 Export

 Rename Project

 Manage Members

 Delete

For more information about your project management options, see [Manage Existing Projects](#).

Business Process Projects

In SAP Build Process Automation, you create a business process project using a combination of one or more process focused skills or building blocks. These skills are known as **artifacts**.

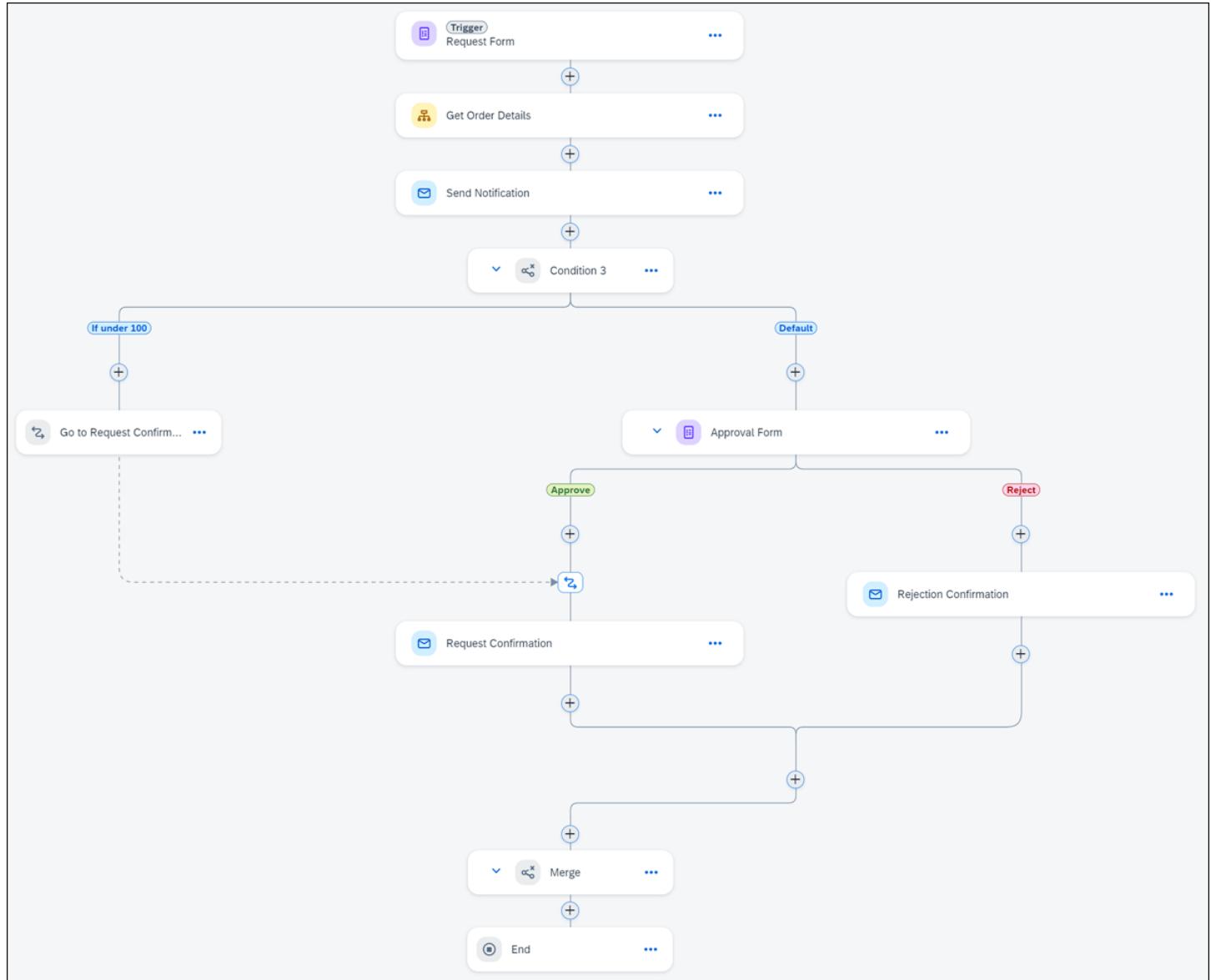
Artifacts help you to configure common steps or tasks needed for a process to run successfully. And the number and order of the artifacts you use depend on the process you're creating. As such, there are no mandatory artifacts needed in a business process project.

To help explain the artifacts available in SAP Build Process Automation, we'll divide them into two categories: Process and Automation

Process Artifacts

These include: Process editor, forms, approval forms, automations, decisions, and visibility scenarios

In this example, process artifacts have been used to create an investment request process:

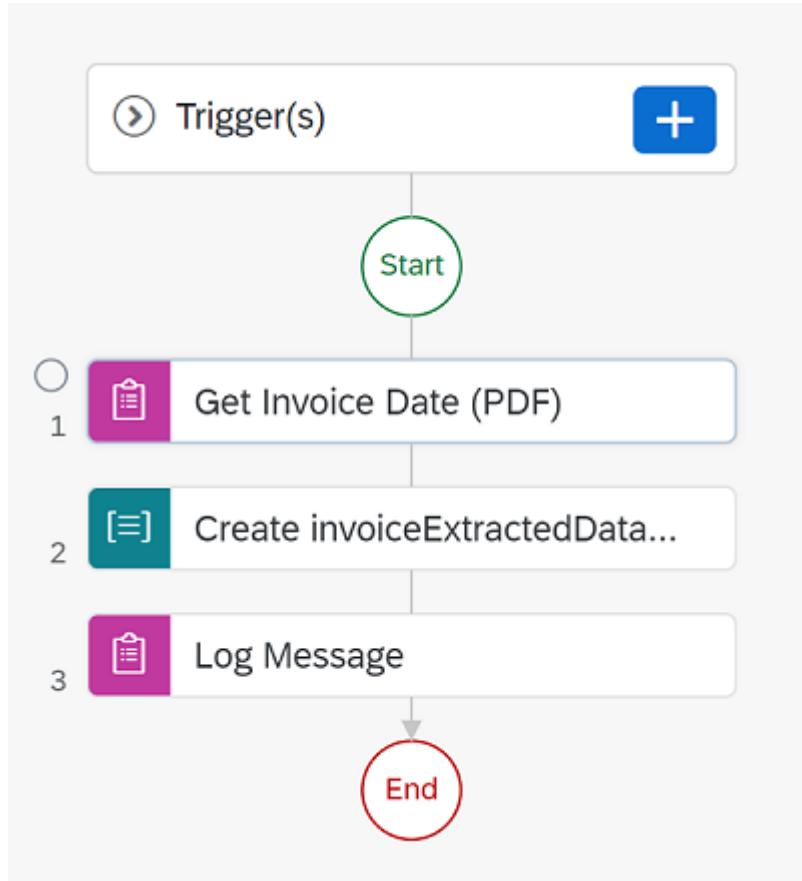


For an overview of available process artifacts, see [Process Artifacts](#).

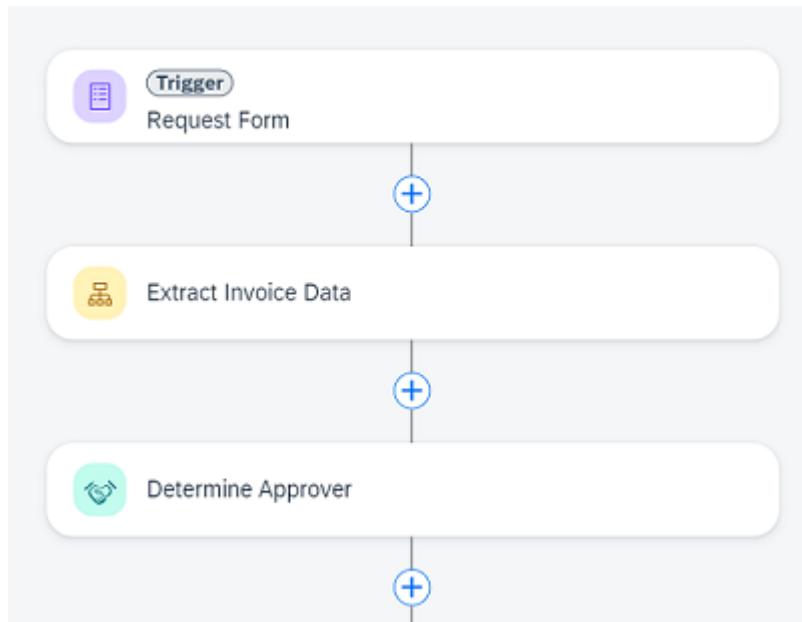
Automation Artifacts

These include: Alerts, applications, project launchers, user tasks, data types, files, and document templates

In this example, automation artifacts have been created to extract data from a PDF:



With this automation then added to a process:



For an overview of available automation artifacts, see [Automation Artifacts](#).

Process Artifacts

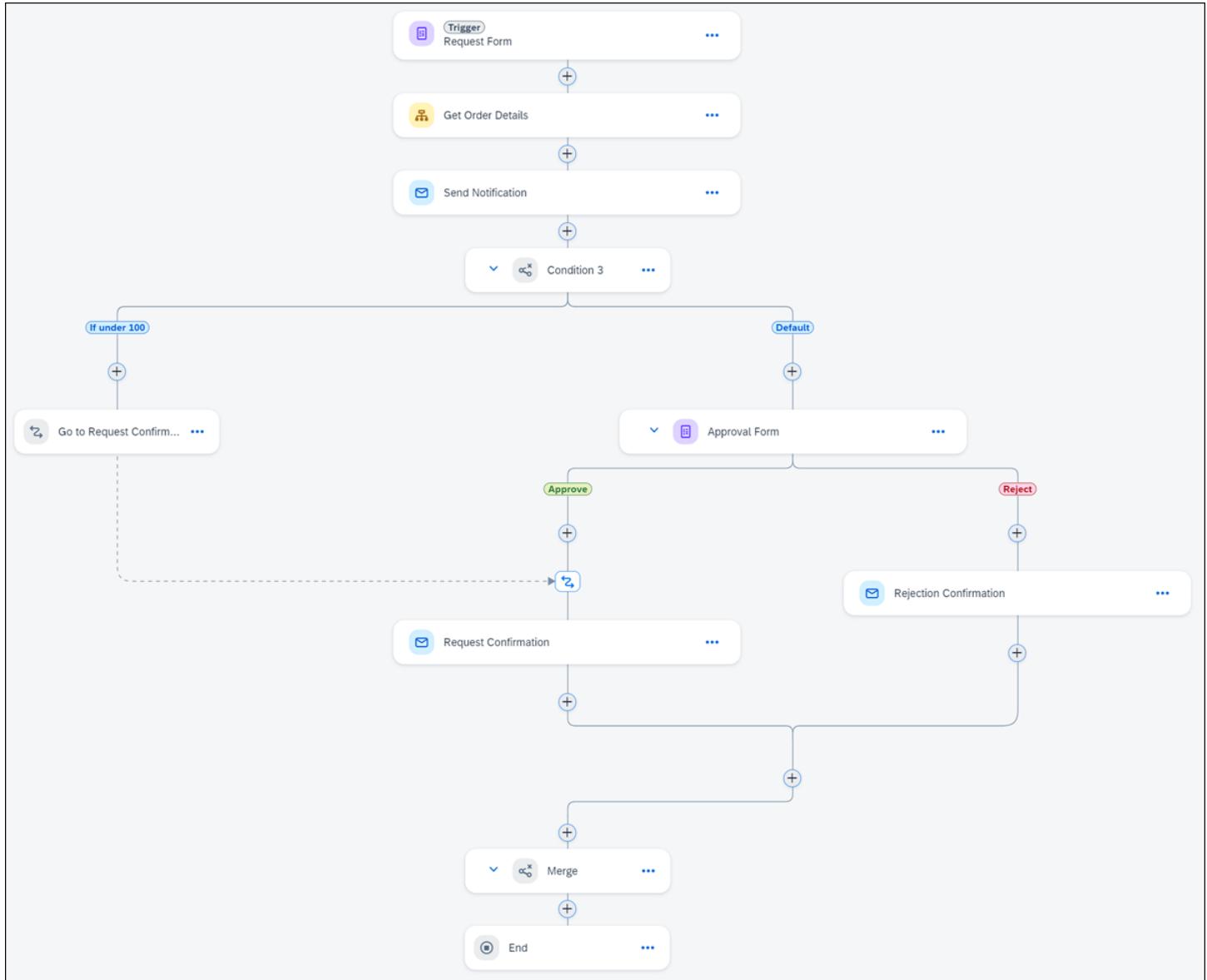
Process artifacts allow you to both create a multistage process and to configure how that process runs and is monitored.

Process Editor

This is a visual canvas on which you map out your business process from start to finish. Other process artifacts are then added to this canvas, with process controls and connectors used to decide how information flows when the process is running.

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

For example, the process editor was used to create this basic investment request process:



This canvas view is the ‘behind the scenes’ of the process and isn’t shown to a person participating in that process.

For more information about using the process editor, see: [Create a Business Process](#)

Forms

Interactive forms are created by adding text elements and input fields to a blank page. Forms can be both starting triggers for a live process or used as optional steps later in the same process.

In the following example, a request form has been created:

Published forms are often how someone submitting a request interacts with the process. Here they access a URL or a SAP Build Work Zone tile, enter their request details, then click 'Submit'.

For more information about creating and adding forms to your process, see [Create a Form](#)

Approval Forms

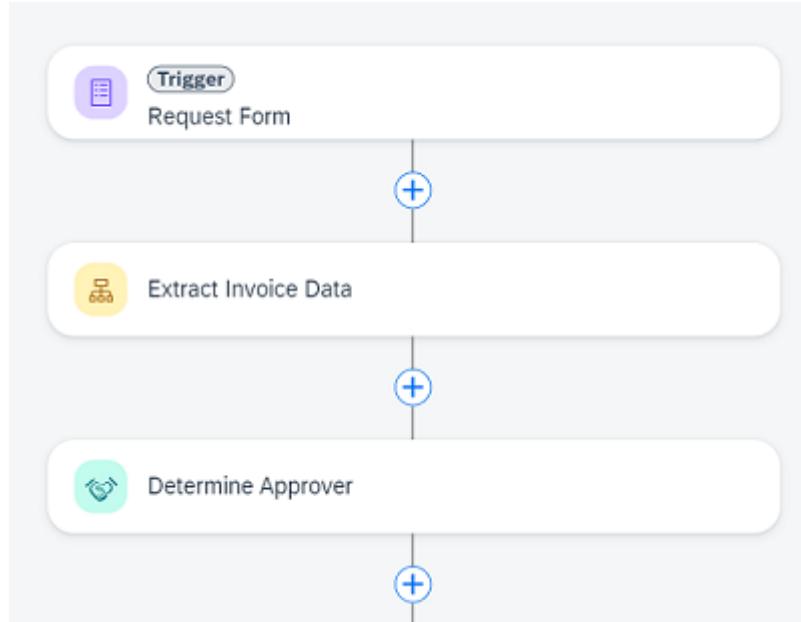
Based on the information submitted in the request form, approval forms share the request with those responsible for reviewing them. The assigned approvers for the process then decide whether to approve, query, or reject the request. Submitted requests appear as tasks in the approver's inbox, helping them to manage requests in one system.

For more information about creating and configuring approval forms, see [Create an Approval Form](#)

Automations

The use of intelligent bots to automate manual, repetitive tasks such as retrieving data from a spreadsheet or submitting information to a database. Automations can be both included in a business process, and configured using the process editor, or used as stand-alone projects. That's why you'll find them mentioned here, too.

In the following example, an automation that extracts invoice data has been added to a process:

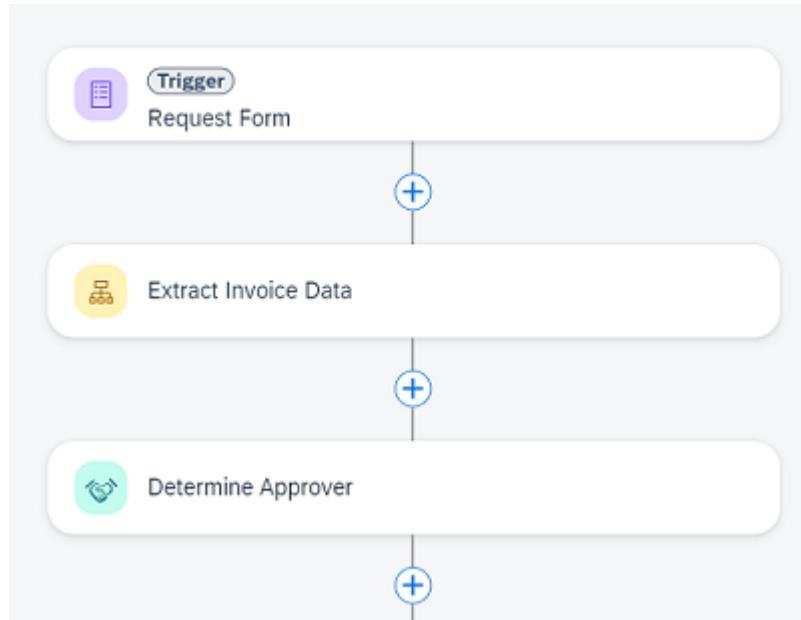


For more information about creating automations, see [Create and Design Automations](#)

Decisions

Effective processes respond and react to the information available. The way they behave in SAP Process Automation, and the direction the process moves, is based on adding and configuring process decisions. For example, if a request comes in from someone in Location A, then root that process to Location A's purchasing team.

In this example process, a decision determines who should approve the submitted request:



Using decisions, one process can effectively handle many scenarios at once.

For more information about creating and configuring process decisions, see [Create a Decision](#)

Visibility Scenarios

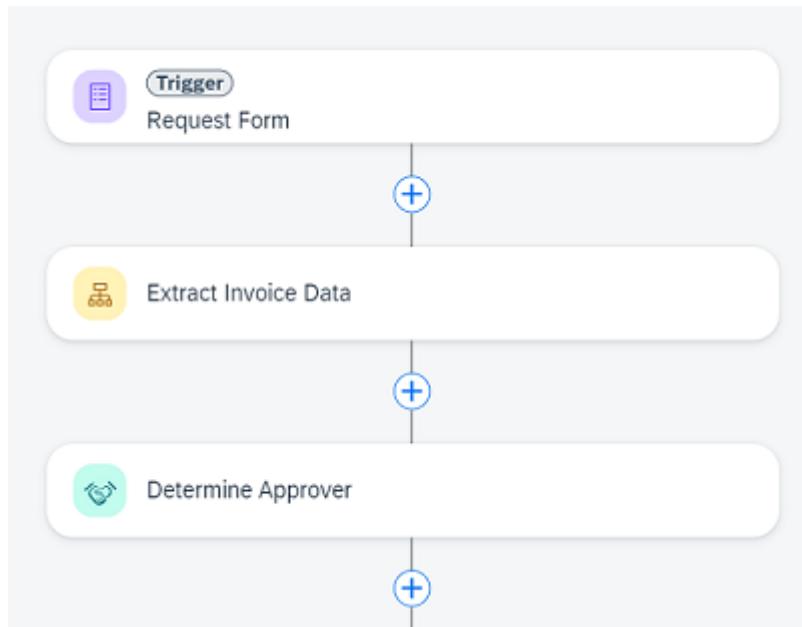
While not added to the process diagram, a visibility scenario allows you to configure and then monitor live versions of the process. By selecting the processes and choosing the type of information displayed, you have greater insights into the efficiency of your process (and react where necessary).

For more information about configuring visibility scenarios for your process, see [Configure a Visibility Scenario](#)

Automation Artifacts

Automation artifacts can be used to create and run an automation. These artifacts are created in the project overview area and can then be added to and configured in an existing automation. As such, they can't be used independently in your business process.

In this example, an artifact has been created to extract information from an invoice:

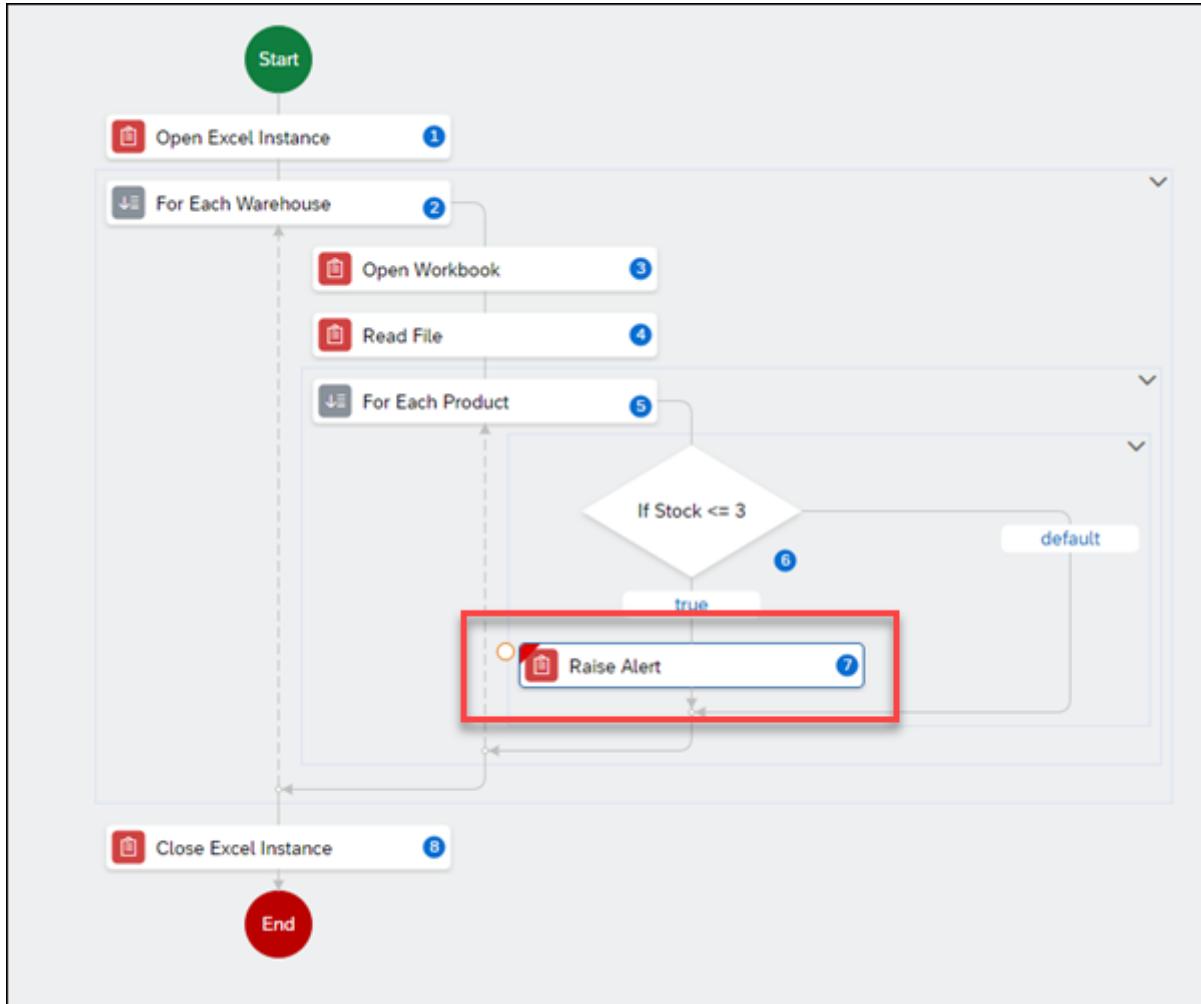


As you can see, an automation can be both a stand-alone project and included in a wider process. In both situations, there are several automation focused artifacts available in SAP Build Process Automation that help to build more complex, intelligent bots.

Alerts

These identify when specified business events occur while an automation is running. For example, when an item is out of stock or a limit has been reached. Alerts are then enhanced using alert handlers. These trigger an email to be sent to the people or system who need to know this information.

In this example, an alert has been created to notify those responsible when stock level has fallen under 3 units:



For more information about adding alerts to your automation, see [Create an Alert](#)

Applications

This artifact allows you to identify applications to control during a live automation, such as a website or software. Known as capturing and declaring applications, the automation can be configured to recognize a screen and interact with the elements displayed.

The screenshot shows the SAP Application Development interface with the 'Select Screen' feature open. On the left, a sidebar lists various open screens, including 'Candidates.xlsx - Excel'. The main area displays an Excel spreadsheet titled 'Candidates.xlsx - Excel' with data for four candidates: Paul Smith (Germany), Alice Norton (Canada), and John Barker (Spain). The 'Screen Details' panel on the right provides options for capturing the application, with 'Capture Application' selected. It also includes fields for 'Application Name' (set to 'Candidates.xlsx - Excel'), 'Application Identifier' (set to 'candidatesxlsxExcel'), and 'Technology' (set to 'Excel sheet listing candidates'). A note in the panel states: 'There might be a specific Excel activity for your use case. If so, you do not need to capture it.'

For more information about capturing applications with an automation, see [Capture and Declare Applications](#)

Project Launcher

Attended automations (those run under human supervision) can be started from a 'systray' agent or system tray agent on your device. By designing a project launcher, you configure which automations can be started (or launched) from that agent. These can either be run manually or automatically based on meeting certain factors.

In this example, the project launcher has been configured to run three automations from the systray:

The screenshot shows the SAP Fiori Project Launcher configuration screen. It has two main sections: 'Launch manually from the agent' and 'Launch automatically by events'.

Launch manually from the agent:

Automation name	Agent label
<input type="checkbox"/> Extract Data	Extract Data

Launch automatically by events:

Automation name	Agent label	Element	Event	Action
<input type="checkbox"/> Open Excel			0	<button>Add event</button>
<input type="checkbox"/> Populate Form			0	<button>Add event</button>

For more information about creating a project launcher, see [Create a Project Launcher](#)

User Task

User tasks take the form of messages received in a person's inbox while a process is running. When creating a user task, you define in a message a specific action that needs to be done by the recipient. For example – to enter personal information, attach a file, or choose from a list of options.

For more information about creating user tasks for your automations, see [Create a User Task](#)

Data Type

Most automations and business processes require data to be inputted, stored, and shared to successfully run. This data always depends on the specific process or those running it, however. As such, there are different data types for different circumstances. By manually creating data types, you can configure and use the data needed for your specific process.

In this example, three data types have been created to be used in an automation:

The screenshot shows the SAP Build Process Automation interface with the 'Investment Details' data type selected. On the left, there's a table with columns: Name, Type, Sample, List, and Required. It lists three fields: 'InvestmentType' (String), 'Country' (String), and 'BusinessUnit' (String). Each row has a 'New Child' button. On the right, there's a panel titled 'Data Type Details' with sections for 'General Information', 'Identifier', 'Description', and checkboxes for 'Data type is active' and 'Strict'.

Name	Type	Sample	List	Required	
InvestmentType	String		No	No	<button>New Child</button>
Country	String		No	No	<button>New Child</button>
BusinessUnit	String		No	No	<button>New Child</button>

Data Type Details

General Information

Name: *

Identifier:

Description:

Data type is active

Strict

For more information about creating and maintaining data types, see [Create a Data Type](#)

Files

Files can be both created and edited in SAP Build Process Automation (text, YAML, JSON, and XML) or uploaded and stored within a project. These files can then be integrated and used within an automation, too.

Document Template

Information can be extracted from a document while an automation is running. For the automation to do this, a document template can be created or uploaded to the project. This template is then used to identify the data available and extract it when directed to do so. For example – retrieving customer information from a frequently used invoice layout.

For more information about working with document templates, see: [Document Processing and Information Extraction](#)

Manage Existing Artifacts

In SAP Build Process Automation, you create a business process project using a combination of one or more process focused skills or building blocks. These skills are known as **artifacts**. Once created, you can manage your artifacts from your business process project overview page.

All Artifacts			Search...		Create	Import
Name	Description	Type	Last edited	Last edited	Created On	
alert	No value	Alert	21 days ago	tom.b...	November 18, 2022	...
autom...	No value	Automation	18 days ago	tom.b...	November 18, 2022	
Employ...	No value	Data Type	18 days ago	tom.b...	November 18, 2022	
form	No value	Form	21 days ago	tom.b...	November 18, 2022	
Process	No value	Process	4 days ago	tom.b...	November 18, 2022	...

Options

When managing your artifacts, you've the following options:

Artifact	Create	Read	Update	Delete	List	Rename	Activate / Deactivate	Copy / Paste / Duplicate	Import Inside
Action Project	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Alert	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	N/A
Application	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Partly (Excel files only)
Data Type	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	N/A
Decision	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Document Template	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
File	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Form	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	N/A
Approval Form	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	N/A
Process	Yes	Yes	Yes	Yes	Yes	Yes	No	No	N/A
Project Launcher	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Only Copy / Paste	N/A
User Task	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	N/A
Visibility Scenario	Yes	Yes	Yes	Yes	Yes	Yes	No	No	N/A

Using Artifacts Within Other Artifacts

Many artifacts can be used within existing artifacts, such as forms being used within a process.

i Note

If you modify or delete an artifact used within other artifacts, this can affect the validity of these artifacts.

The following table summarizes the available options:

	Action Project	Alert	Application	Automation	Data Type	Decision	Document Template	File	Form	Approval Form	P
Automation	Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes	N/A	N/A	N
Data Type	N/A	N/A	N/A	N/A	Yes	N/A	Opposite	N/A	N/A	N/A	N
Decision	N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A	N/A	N/A	N
Document Template	N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A	N/A	N/A	N
Form	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N
Approval Form	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N
Process	Yes	Yes (via an automation)	Yes (via an automation)	Yes	Yes	Yes	Yes	No	Yes (via an automation)	Yes	Y
Project Launcher	N/A	N/A	Yes	Yes	N/A	N/A	N/A	N/A	N/A	N/A	N
User Task	N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A	N/A	N/A	N
Visibility Scenario	N/A	N/A	N/A	N/A	N/A	Yes	N/A	N/A	Yes	Yes	N

Create a Business Process

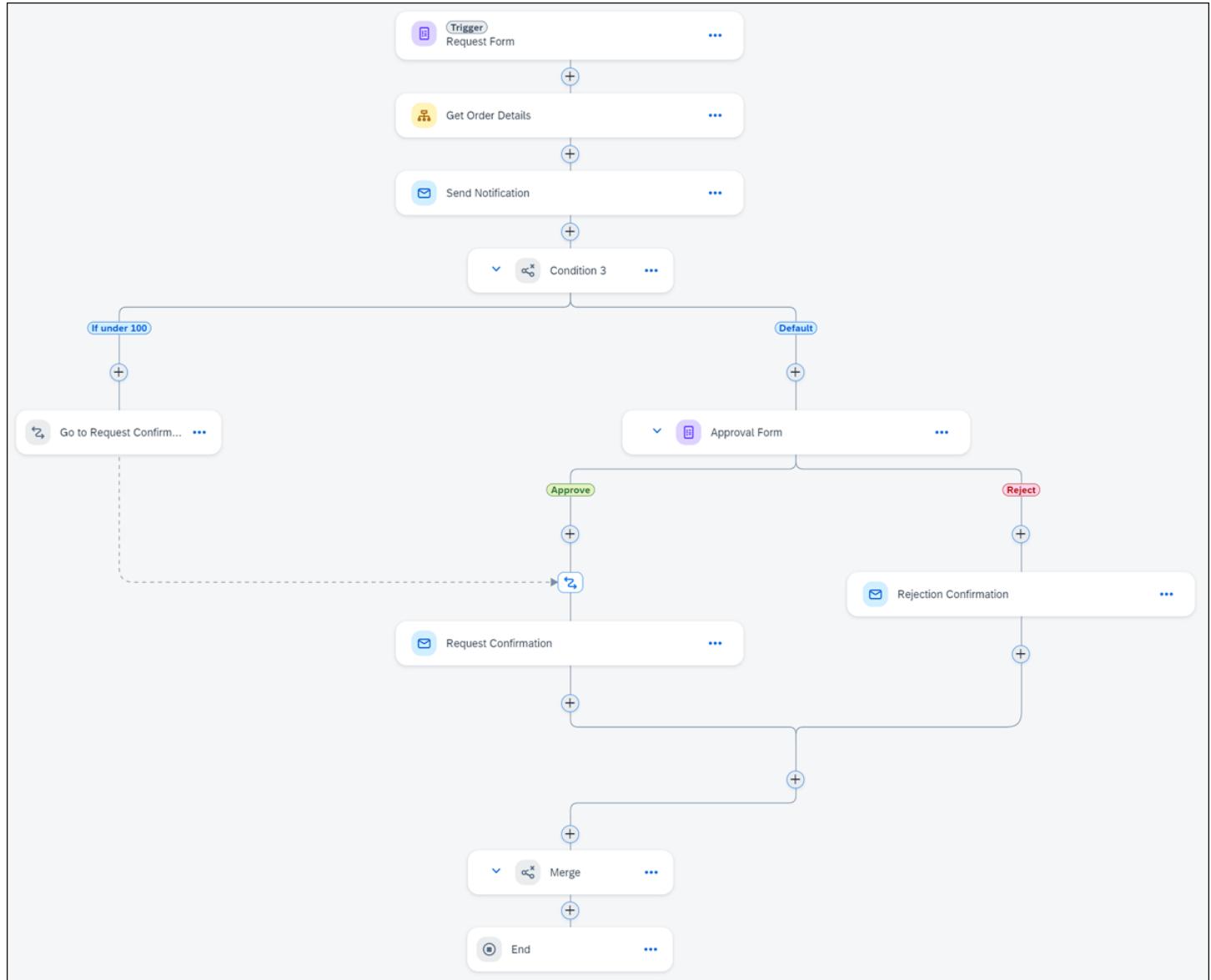
With SAP Build Process Automation, you can visually create a business process version using a combination of artifacts (such as forms and decisions) and process controls (such as branches, conditions, and mail notifications).

Context

A business process is started by defining a trigger, an event that indicates to your SAP Build Process Automation tenant to start a process instance.

Process triggers can be a form, such as a request form, an API call, where an external system starts the process, or an event.

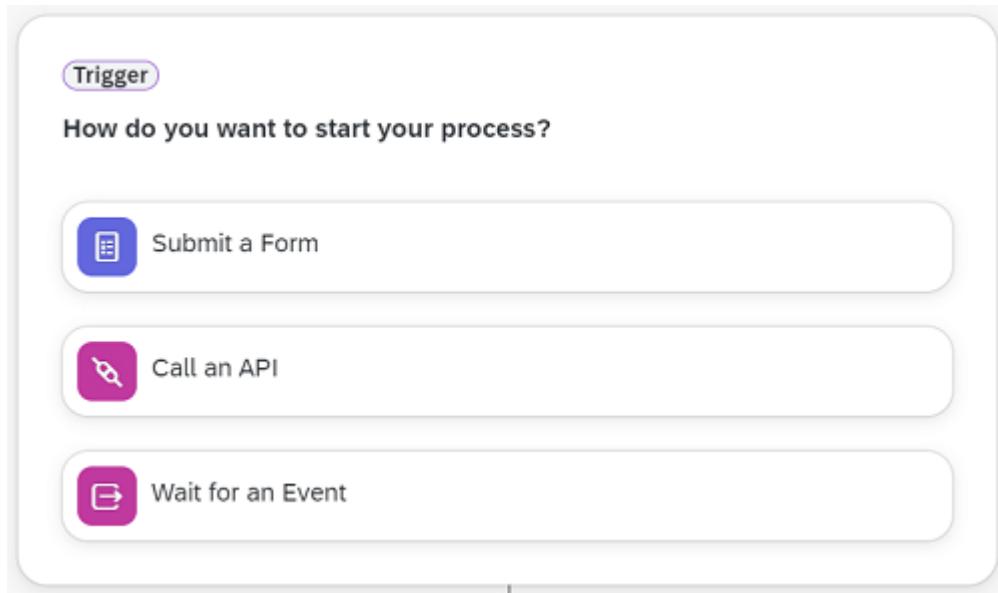
In this example, the process is triggered when a request form is submitted:



The process then continues with a combination of [Process Artifacts](#) (such as forms, approval forms, and decisions) and optional process controls.

Procedure

1. To create your project and process in the **Lobby**, choose **Create > Build an Automated Process > Business Process**.
 2. Enter a name and description for your project, and choose **Create**.
 3. Enter a name, identifier, and optionally a description for your process.
 4. Choose **Create** again.
- The process editor loads.
5. Configure your process trigger, choosing between:
 - o API trigger - [Configure an API Trigger to Start a Process](#)
 - o Form trigger - [Configure Settings for Forms and Approval Forms](#)
 - o Event trigger - [Create Event Triggers](#)

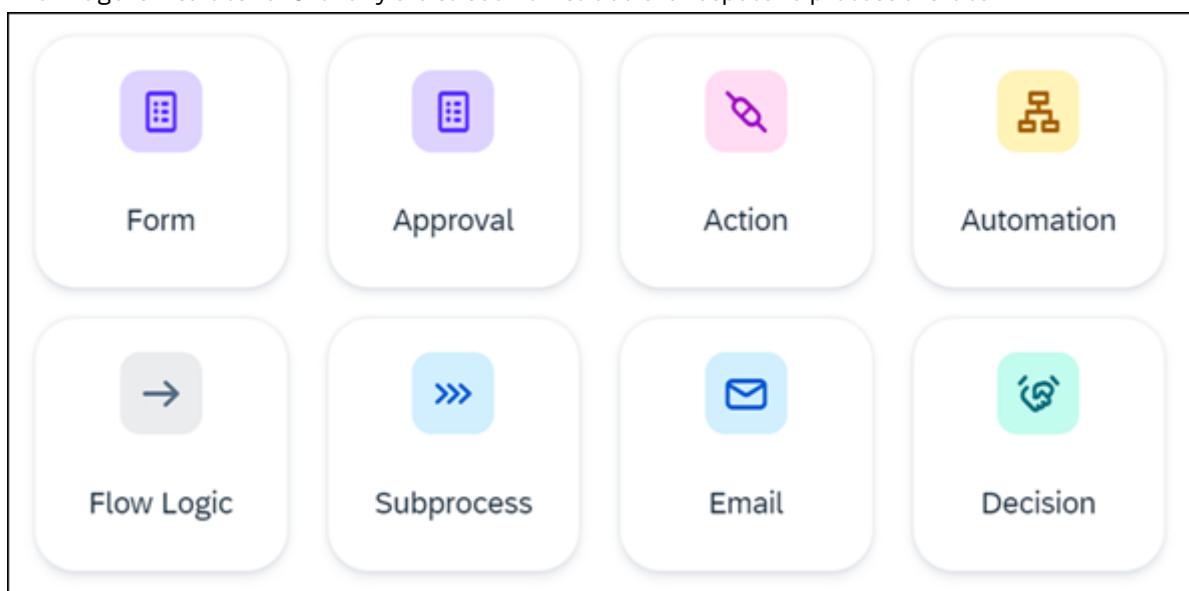


6. Click **+** and select your required process artifacts or process controls. Sometimes, you get to select again whether to create a new artifact or reuse an existing one. Repeat this step as necessary to continue building your process.

i Note

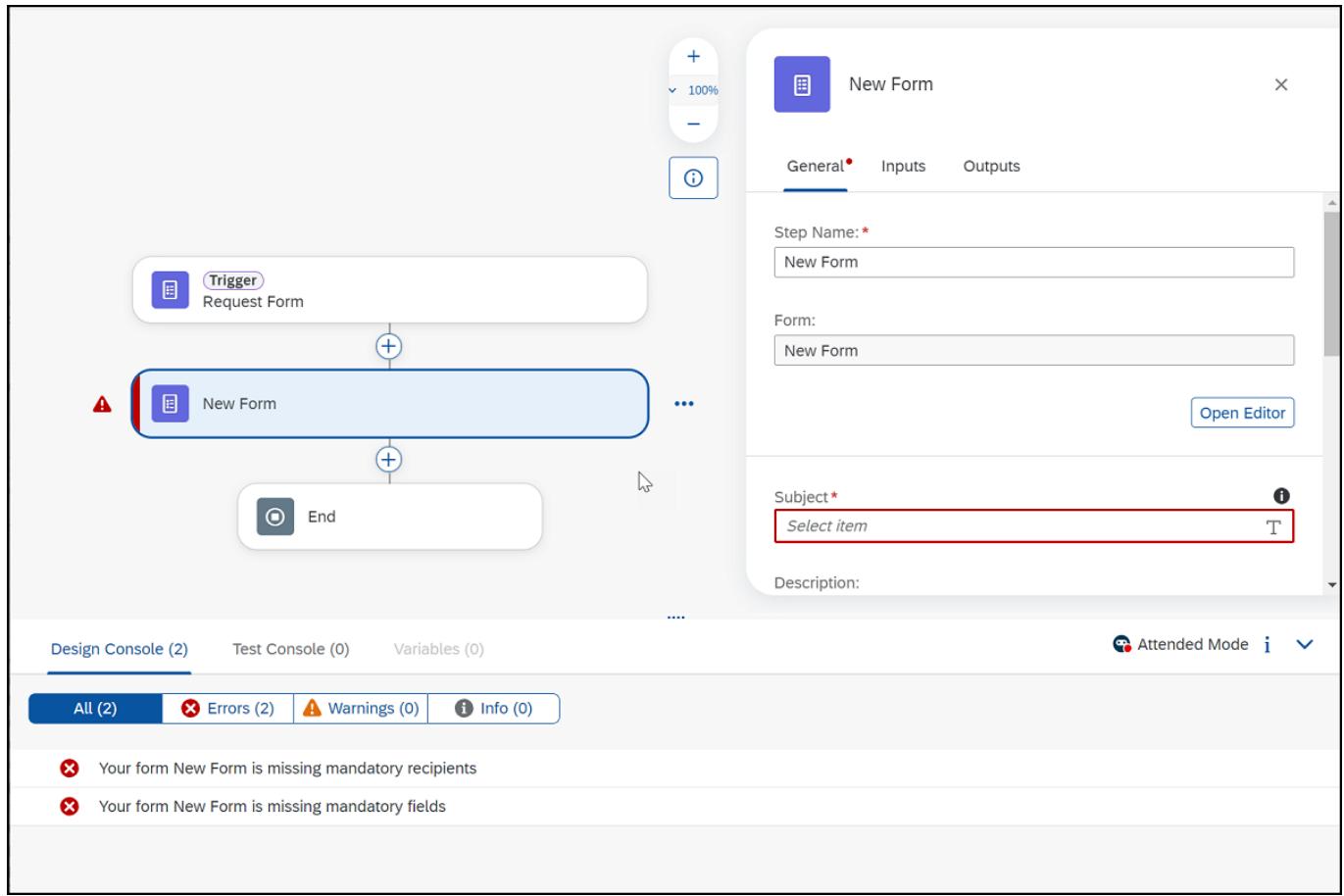
You can change the order of your process steps using drag and drop. The selected step is automatically connected to the process.

The image is interactive. Click any tile to see how to add the respective process artifact.



7. Review and correct all process errors, indicated by a red icon on the impacted artifact.

In this example, the form artifact is missing a subject. The design console displays the error details.



8. Once the process is completed and all errors are corrected, save your process.

The project is ready to be released and deployed.

Related Information

[Release a Project](#)

[Deploy a Project](#)

Configure Process Variables

The outputs of any step that you add to your process can be consumed as inputs for the subsequent steps in a process. In addition, you can create custom variables on a project level that aren't bound to a particular step.

Context

If you add actions or subprocesses to your process, their data types become available as step input to any later step in the process. For data types that contain deep structures, such as business partner information, this saves you much repetitive work.

Using custom variables, you can make any information available at the global level at any stage in the process. For more information on the use cases, see [Use Cases for Custom Variables](#).

Procedure

- For the input variables, add an action to your process. See [Add Actions to a Process](#).
- Click anywhere on the canvas and open the side panel.
- On the **Process Details** side panel, choose **Variables** > **Process Inputs**.

4. To configure process input variables, next to **Process Inputs**, choose **Configure**.

a. Choose **Add Input**, and enter the following data:

- **Name and Identifier:** For example, `businessPartner`.
- **Type:** Select any data type from the list:
 - The basic data types
 - All data types of the current project
 - All date types of any dependent project. These dependencies are, for example, created when adding actions or subprocesses to the project.

The screenshot shows the 'Configure Process Inputs' dialog. At the top, there is a message: 'Some inputs might be bound to other processes and deleting them can cause errors.' Below this is a table titled 'Inputs' with columns: Name *, Identifier *, Type *, Required, and List. In the 'Name' field, 'emaillist' is entered. In the 'Identifier' field, 'emaillist' is also entered. In the 'Type' column, a dropdown menu is open, showing 'String' selected. A checkbox labeled 'Required' is checked. A checkbox labeled 'List' is checked. To the right of the table is a vertical scroll bar. Below the table is a dropdown menu with several categories and items listed:

- Basic**
 - String
 - Number
 - Boolean
 - Date
 - DateTime
- Current Project**
 - Table List Object
 - WF data type
- petstoreactionproject**
 - ApiResponse
 - Category
 - get_findPetsByStatus_200_output_schema
 - get_findPetsByTags_200_output_schema
 - get_getOrderById_200_output_schema
 - get_getPetById_200_output_schema

 At the bottom right of the dialog are 'Apply' and 'Cancel' buttons.

- To create a list, select **List**.
- To make this variable a required entry, select **Required**.

5. Next to **Process Outputs**, choose **Configure** and add outputs in the same way as you added inputs.

6. To configure custom variables, choose **Configure** next to **Custom Variables**. Then, choose **Add Variable**, and enter the following data:

- **Name and Identifier:** For example, `businessPartner`.
- **Type:** Select any basic data type from the list.
- To create a list, check **List**.

7. Choose **Apply**.

8. Save your changes.

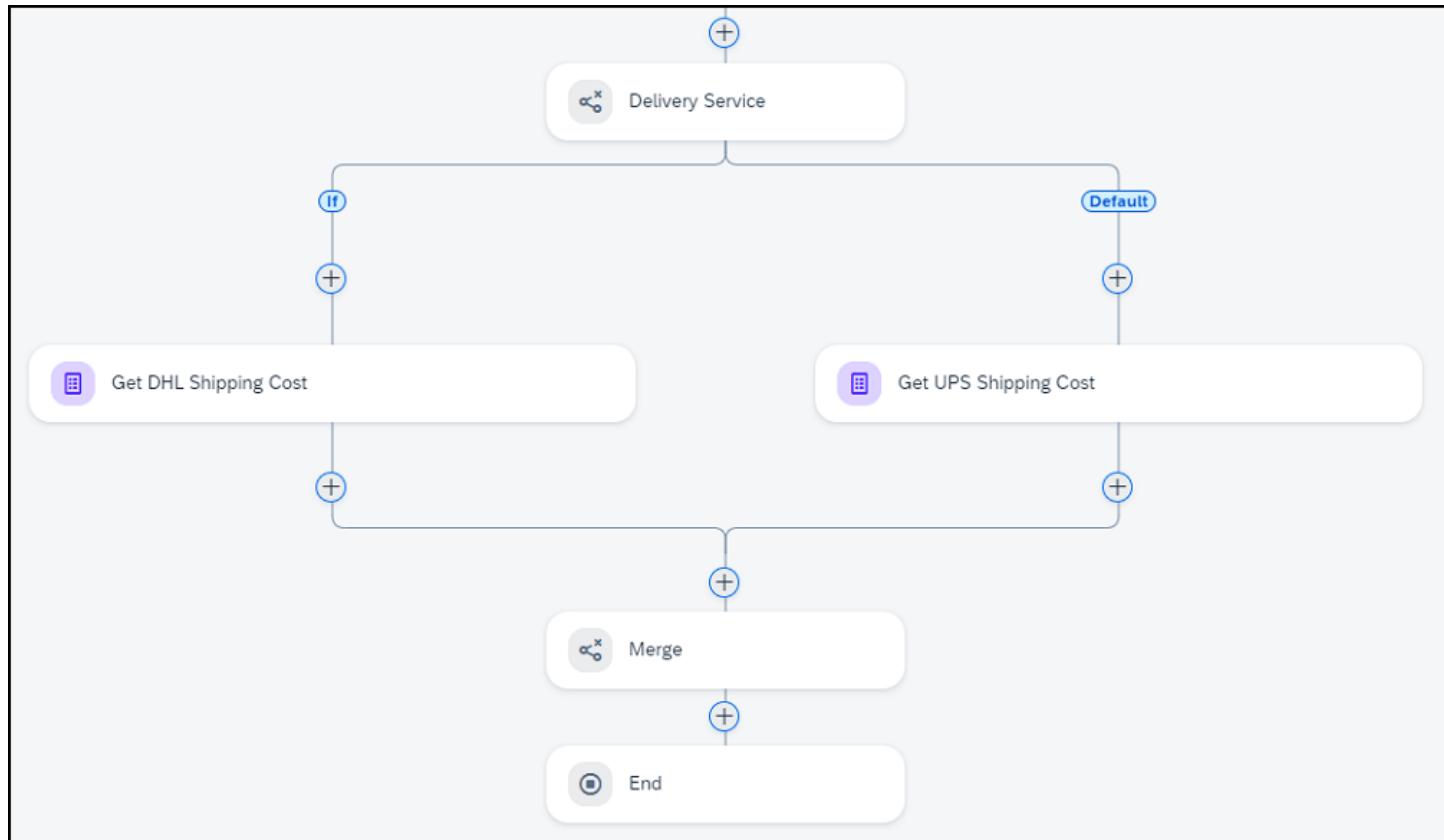
Use Cases for Custom Variables

Using custom variables, you can make any information available at the global level at any stage in the process.

Using custom variables, you can make any information available at the global level at any stage in the process. Custom variables have the following use cases:

Independency of Branch Outcome

Make the step that follows a condition branching independent of the outcome of the conditional branches. Let's look at an example: We have a condition where the branches have steps for two shipping companies. The **Final Approval** form needs to cover both possible outcomes to map the **Shipping Cost** field. So, we map an independent custom variable instead of the specific process output from the steps in the branches.



The following shows the order, in which the custom variable is inserted into the process.

1. We define the **shipping cost global** custom variable for the process.

Process Details

- General Variables Visibility
- > Process Inputs
- > Process Outputs [Configure](#)
- > Custom Variables [Configure](#)

T bike model global *
shipping cost global *
bike price global *

2. In the two steps in the branches, we use the new custom variable on the [Outputs](#) tab.

The screenshot shows a process flow in the SAP Fiori process builder. It starts with a trigger 'Equipment Request Form' leading to a 'Condition' step. The condition splits the flow into two parallel branches: 'Get DHL Shipping Cost' and 'Get UPS Shipping Cost'. Both branches merge back into a single path that leads to 'Final HR Approval' and finally ends. On the right, there is a 'Process Content' panel showing the steps and a 'Form' panel for 'Get DHL Shipping Cost'. In the 'Form' panel, under the 'Outputs' tab, a custom variable '# Shipping Cost' is being mapped to the 'Shipping Cost' field. The 'Set Custom Variables' section indicates that the value will be updated after the step is executed.

3. In the [Final Approval](#) form, we can now map the [Shipping Cost](#) field to the custom variable as well, and make it independent of the condition branches.

This screenshot shows the same process flow as the previous one, but with a change in the 'Final HR Approval' step. The 'Final HR Approval' step now has its 'Shipping Cost' field mapped to the custom variable '# Shipping Cost' via the 'Outputs' tab. The 'Set Custom Variables' section in the 'Form' panel for 'Final HR Approval' also shows the mapping of '# Shipping Cost' to 'Shipping Cost'.

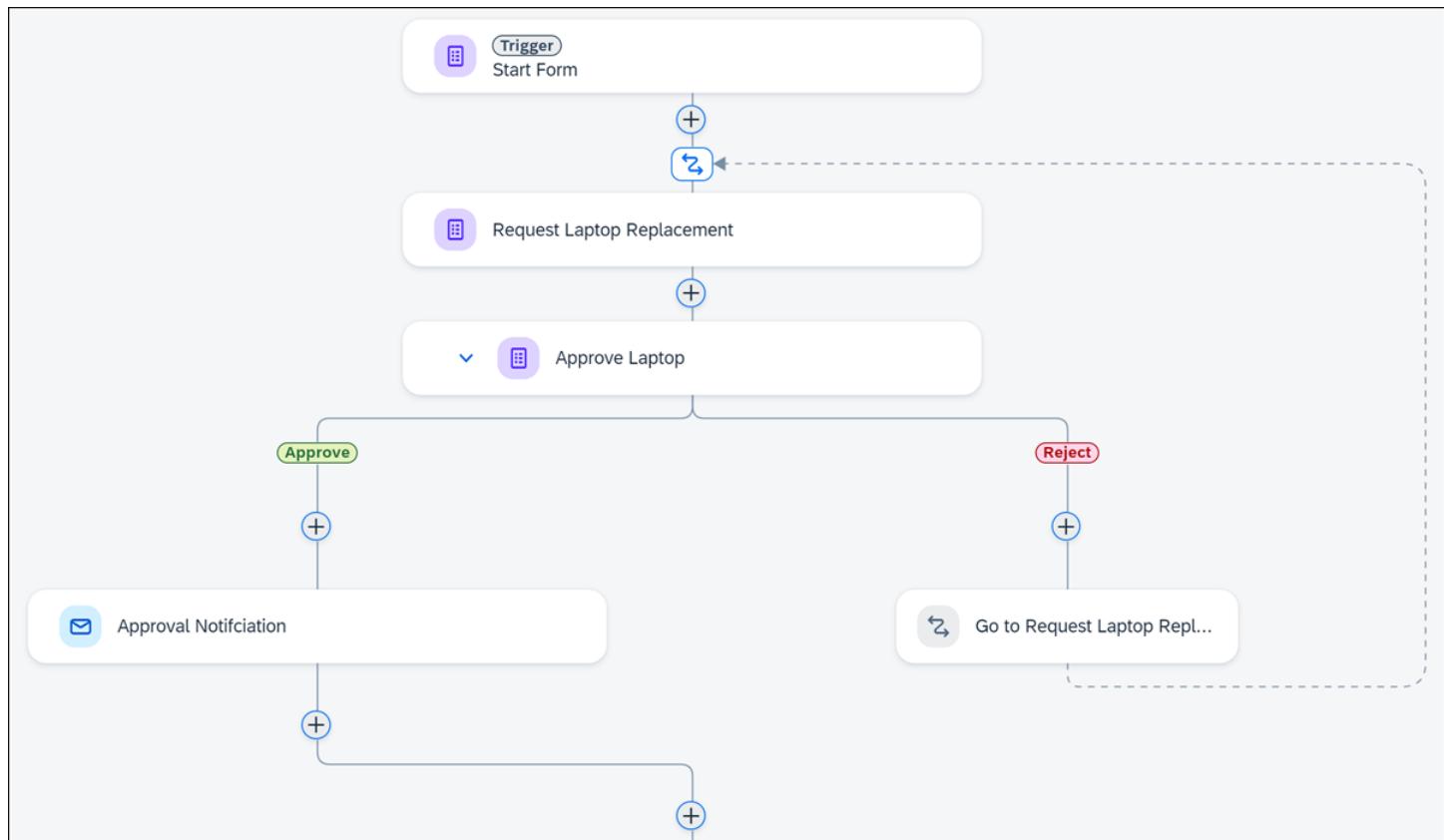
4. At runtime, the final approval form receives an input for the shipping cost, no matter which branch is used.

Rework Request

Send a request back for rework while preserving the already-entered values and the comments added by the parties involved.

Let's assume that a step needs rework before the approver accepts it. To send the process back to the previous step that needs rework, you can define a custom variable that is already available before the approval step outcome is available. In this way, the This is custom documentation. For more information, please visit the [SAP Help Portal](#)

previous step can anticipate the result of the approval step and use the correct variable.



One Field, Multiple Processes

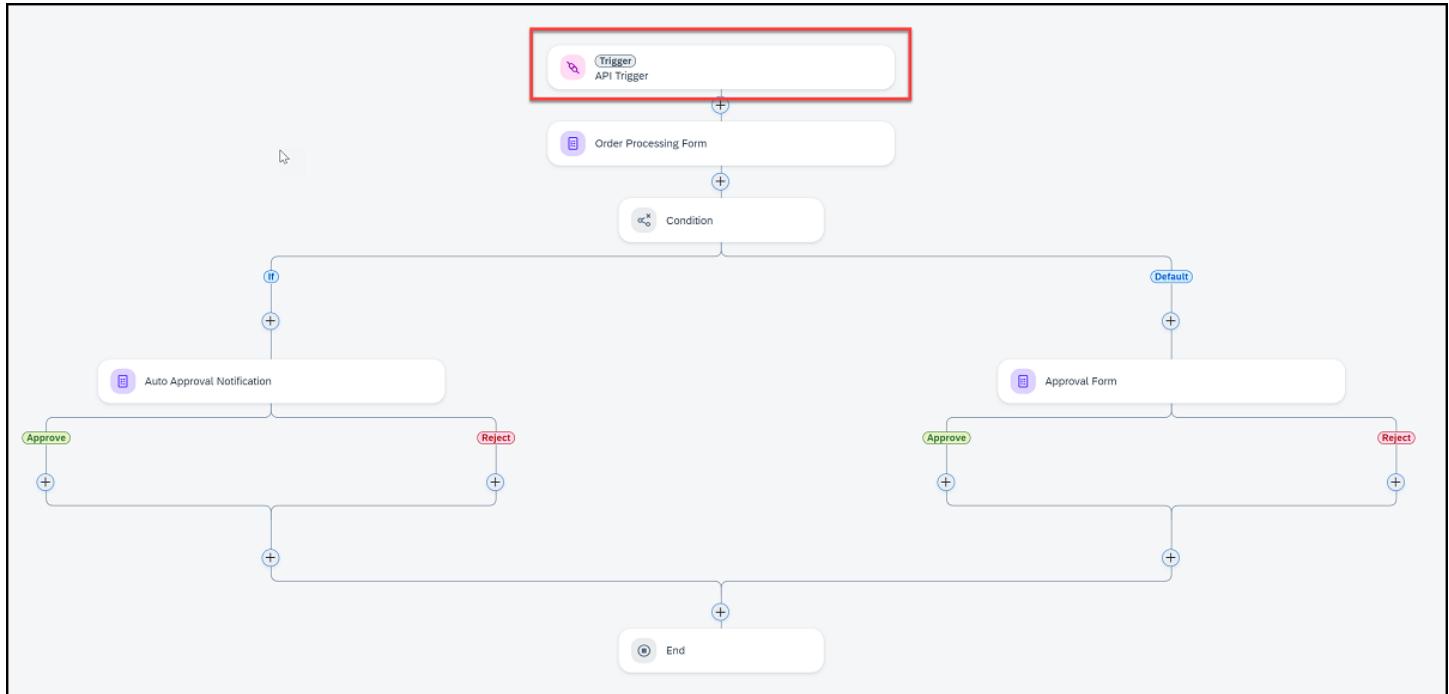
Multiple process steps edit the same field. This field is immediately available and doesn't need to be mapped and carried forward.

Configure an API Trigger to Start a Process

You can start an instance of your process using an API call, with the inputs for the call configured in the process editor. These inputs can then be used as input fields in your process, for example, in a form or approval form.

Context

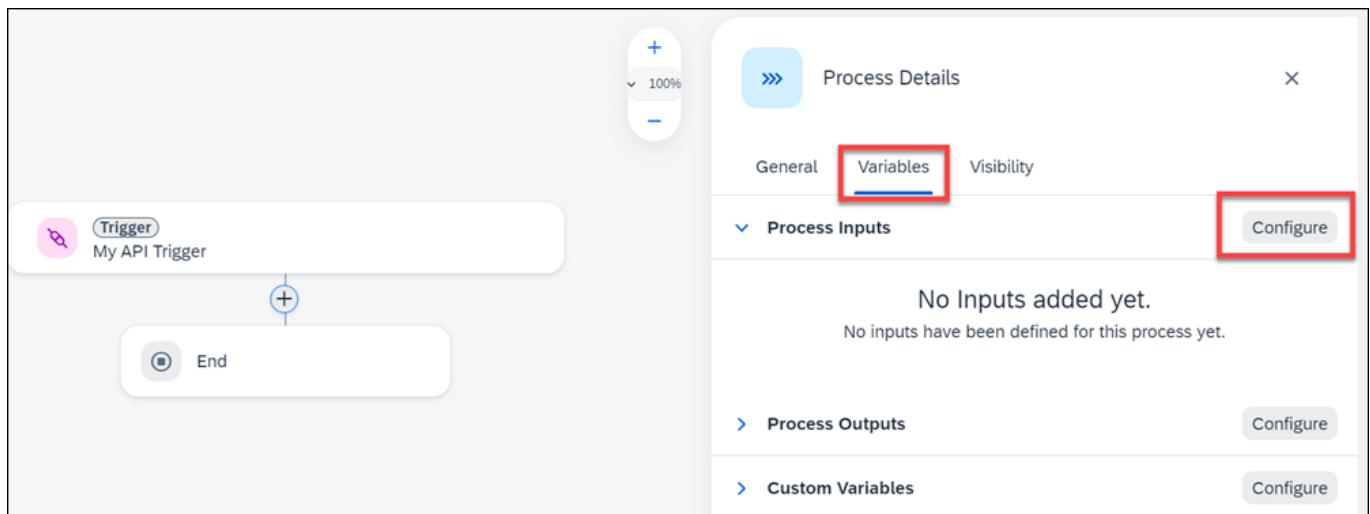
In this example, an API call is used to trigger a sales order process:



You can display such API triggers in the same overview as the automation triggers. For deployed processes, go to [Monitor](#) > [Manage](#) > [Triggers](#) from the lobby.

Procedure

1. On the **Trigger** tile in the process editor, choose **Add a Trigger**. Then, select **Call an API**.
 2. Enter a name for your trigger and a description. The identifier is generated based on the name.
 3. Click anywhere on the canvas to display the **Process Details** side panel.
 4. Choose  **Variables**  **Process Inputs**  **Configure** .



5. Click **Add Input** and then enter a **Name**, **Identifier**, and **String**, and specify if this is a **Required Input**.
Repeat this step to add all necessary inputs.

Configure Process Inputs

Some inputs might be bound to other processes and deleting them can cause errors.

Inputs

Name *	Identifier *	Type *	Required	List
No Inputs added.				

Add Input

Apply Cancel

6. Choose **Apply**.

Configure Process Inputs

Some inputs might be bound to other processes and deleting them can cause errors.

Inputs

Name *	Identifier *	Type *	Required	List
Cost Center	costCenter	String	<input checked="" type="checkbox"/>	<input type="checkbox"/> Delete
External Sourcing	externalSourcing	String	<input checked="" type="checkbox"/>	<input type="checkbox"/> Delete
Investment Value	investmentValue	String	<input checked="" type="checkbox"/>	<input type="checkbox"/> Delete

Apply Cancel

The inputs are displayed in the process settings side panel.

Process Details

General Variables Visibility

Process Inputs

- Cost Center *
- External Sourcing *
- Investment Values *

Configure

7. Once a form has been added to your process, you can then map these inputs to the form fields:

Process Content

Search

Process Inputs

- Cost Center
- External Sourcing
- Investment Values

Process Metadata

Trigger My API Trigger

My First Form

General Inputs Outputs

Cost Center
Select item T

External Sourcing
Select item T

Investment Value
Select item T

8. Save your changes.

9. With the API call set as your process trigger, you can now test the call using Postman. See [SAP Community - How to Test Your API Call Trigger Using Postman](#).

i Note

We support the ISO 8601 format for date and time:

YYYY-MM-DD (2023-03-16)

hh:mm:ss (15:33:16)

Create Event Triggers

Prerequisites

- Your SAP S/4HANA Cloud system is available and connected with SAP Build Process Automation, and is able to receive all required notification events. For more information, see [Enable the Consumption of SAP S/4HANA Cloud Events](#).
- You have set up an Event Mesh service instance in your SAP BTP cockpit subaccount. For more information, see [Setting Up SAP Event Mesh in BTP Cockpit](#).
- You have created your queue and your queue subscriptions. For more information, see [Manage Queues](#).
- You have created a webhook subscription to subscribe to an event queue. For more information, see [Manage Queue Subscriptions](#).
 - You have created a service key. For more information, see [Create a Service Key for the SAP Build Process Automation Instance](#).
 - You have fulfilled the **Webhook URL** by using the URL from the instance of the **SAP Build Process Automation plan standard**  **Service Key**  **api** endpoints and adding `internal/be/v1/events`. As a result, your webhook URL should be as follows:
`https://spa-api-gateway-bpi-eu-prod.cfapps.sap.hana.ondemand.com/internal/be/v1/events`.
 - You have inserted the **Client ID**, the **Client Secret**, and the **Token URL** by adding `oauth/token` to the latter. You can copy and paste these credentials from the service key.

Context

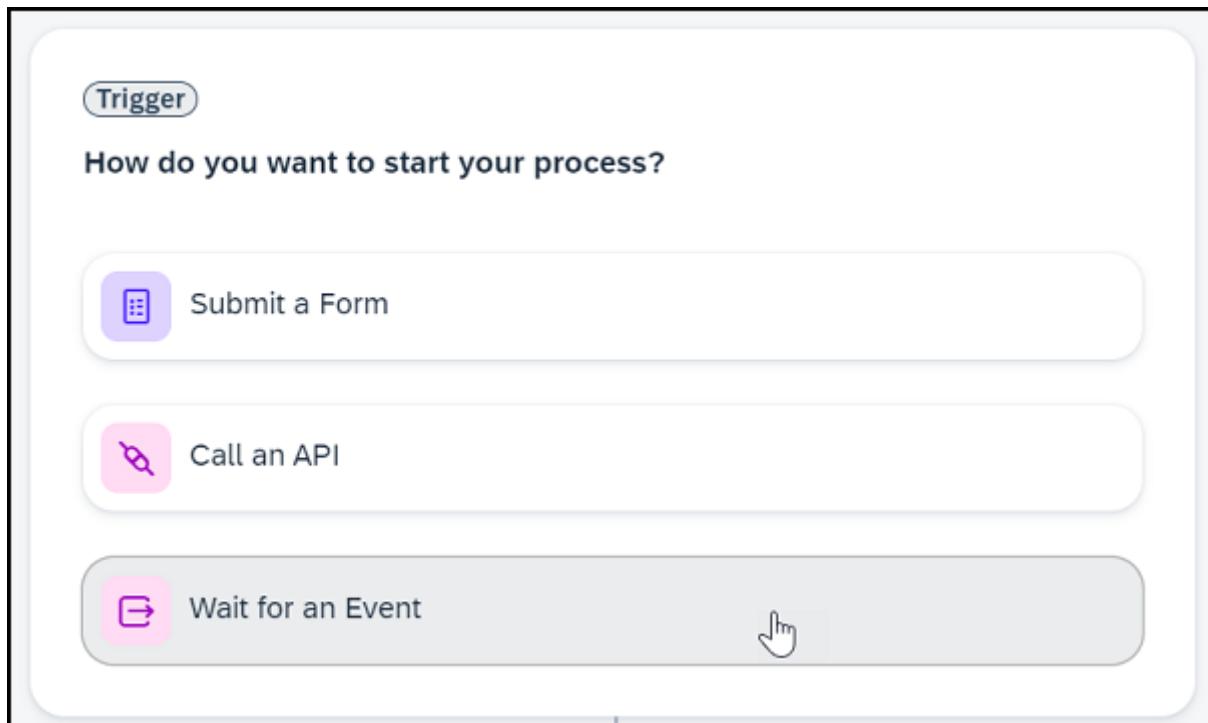
The event triggers onboard and listen to back-end events emitted from an external source system, and reacts to such events by triggering artifacts such as processes and automations.

All public events published on SAP Business Accelerator Hub for [SAP S/4HANA Cloud](#) and [SAP S/4HANA](#) are supported.

You already have created a project and can now add a process as an artifact that has to be started by an event trigger.

Procedure

1. On the **Trigger** tile in the process editor, choose **Add a Trigger**. Then, select **Wait for an Event**.



2. Enter the following data:

Field	Description
General	
Name	<p>Enter the name of your event trigger.</p> <p>i Note The event trigger name cannot be longer than 64 characters.</p>
Executes	Set by default to the current process name and is read-only.
Add Description	<p>Enter the description of your event trigger.</p> <p>i Note The description cannot be longer than 1024 characters.</p>
Event Configuration	
Event Object	Select an event object from the predefined list with consolidated event groups. You can use the fuzzy search to match your choice easier.
Event	Select an event that's available for the event object you have chosen at the previous step.

Create Event Trigger

General

Name: *	Executes: *
<input type="text" value="Enter a name"/>	<input style="background-color: #e0f2e0; border: 1px solid #c0e0c0; padding: 2px 10px; width: 150px; height: 20px; border-radius: 5px; vertical-align: middle;" type="text" value="» MyProcess"/> ▼
Description:	
<input style="width: 100%; height: 40px; border: 1px solid #c0e0c0; border-radius: 5px; padding: 5px; vertical-align: middle;" type="text" value="Enter a description"/>	

Event Configuration

Event Object *	Event *
<input style="width: 150px; height: 20px; border: 1px solid #c0e0c0; border-radius: 5px; padding: 2px 10px; vertical-align: middle;" type="text" value="Select Event Object"/> ▼	<input style="width: 150px; height: 20px; border: 1px solid #c0e0c0; border-radius: 5px; padding: 2px 10px; vertical-align: middle;" type="text" value="Select Event"/> ▼

Create **Cancel**

3. Choose **Create**.

A success message for creating the event trigger appears.

Related Information

[Manage Event Triggers](#)

Manage Event Triggers

You can edit, enable or disable, and delete event triggers using the different options of SAP Build Process Automation.

Related Information

[Create Event Triggers](#)

Manage Event Triggers from the Project Overview

You can edit, enable or disable, and delete event triggers using the option of project **Overview**.

Procedure

1. To manage your event trigger, choose **Overview > Triggers > ... (More)**.

Artifacts (27) **Triggers (1)** Dependencies (4)

Triggers Search... Trigger Start

Name	Executing
User Starts Work	Trigger Start

...

Edit
Deactivate
Delete

2. Select one of the following options:

- **Edit:** To change the event trigger's **Name** or **Description**
- **Deactivate:** To not deploy the trigger
As a result, the event trigger becomes grayed out.
- **Activate:** To redeploy the event trigger
- **Delete:** To delete the event trigger artifact

3. Save your changes.

Manage Event Triggers from the Process Editor

You can edit, enable or disable, and delete event triggers using the option of process editor.

Procedure

1. To manage your event trigger, go to Overview and choose... (**More**).

2. Choose one of the following actions:

- **Edit:** To change the event trigger's **Name** or **Description**



- **Delete:** To delete the **Event** trigger artifact

3. Save your changes.

Manage Event Triggers from the Triggers View

You can edit, enable or disable, and run event triggers using the option of the **Triggers** view.

Procedure

- In SAP Build, choose **Control Tower**, then choose the **Environments** tile.
- Select your environment and choose the **Unattended Triggers** tab.
- Search for your trigger using the Type, Time range validity, Project, and Attributes filters.

The screenshot shows the SAP Build interface. On the left, there's a sidebar with links like Lobby, Connectors, Actions, Events, Automation SDK, Store, Monitoring, and Control Tower (which is currently selected). The main area is titled 'Control Tower / Environments / Environment'. At the top, there are tabs: Projects, **Unattended Triggers** (which is highlighted with a red box), Attended Triggers, Agent Management, Alert Handlers, Variables, API Keys, and More. Below the tabs are search fields for Type, Time range validity, Project, and Attributes, along with a 'Search...' button. A table below shows one item: 'Test trigger' (Trigger icon), 'Auto 2' (Executing icon), 'Second... V1....' (Project & Version), 'Over a month ago' (Last Updated), and 'Attributes' (button). There are also 'Add Trigger', 'Edit', and 'Delete' buttons at the top right of the table.

- Choose ... (**More Options**) for the selected event trigger and select one of these options.

- To pause the trigger action, choose **Disable**.

As a result, the event trigger becomes grayed out.

- To execute the event trigger processes, choose **Enable**.
- To change the event trigger **Name** or **Description**, choose **Edit**.
- To execute the event trigger in order to initiate event delivery with the user-provided data immediately, choose **Run Now**.
 - If you choose **Run Now**, the **Run My Event Trigger** pop-up window appears. The **Event** and **Executes** fields are predefined with the event trigger data.
 - In the **Test with following data** field, a structured message header must be provided, including the event data with the respective inputs. Ensure the following properties are correctly configured, as in the example below:

Sample Code

```
{
  "type": "sap.s4.beh.salesorder.v1.SalesOrder.Created.v1",
  "specversion": "1.0",
  "source": "internal",
  "id": "0794ef45-7741-1eea-b7be-ce30f48e9a1d",
  "time": "2023-12-04T06:21:52Z",
  "datacontenttype": "application/json",
  "data": {
```

```

    "SalesOrder": "3016330",

    "EventRaisedDateTime": "2023-08-08T10:16:27.919Z",

    "SalesOrderType": "TA",

    "SalesOrganization": "1010",

    "DistributionChannel": "10",

    "OrganizationDivision": "00",

    "SoldToParty": "1000360"

}

}

```

- Ensure the correctness of the message header properties:

Property	Description
"type"	Represents the type of the event associated with the trigger. Update it using the same structure but with the corresponding event object and event.
"specversion"	The default value is "1.0" and must not be altered.
"source"	The default value is "internal" and must not be altered.
"id"	Random identifier with a maximum length of 64 characters. It must be unique. If not unique, then event will not appear as a separate record in Event monitoring UI.
"time"	Date-time in the ISO 8601 format.
"datacontenttype"	Specifies the type of content. The default value is "application/json" and must not be altered.
"data"	Represents the parameters of events utilized in the process definition, including corresponding input values. <p>i Note</p> <p>You should not provide any properties that are not defined in the process definition.</p> <p>In case of such, the validation will be activated, and then you will not be able to execute the trigger.</p>

- If the event trigger is disabled, the **Run Now** option is grayed out.

5. Save your changes.

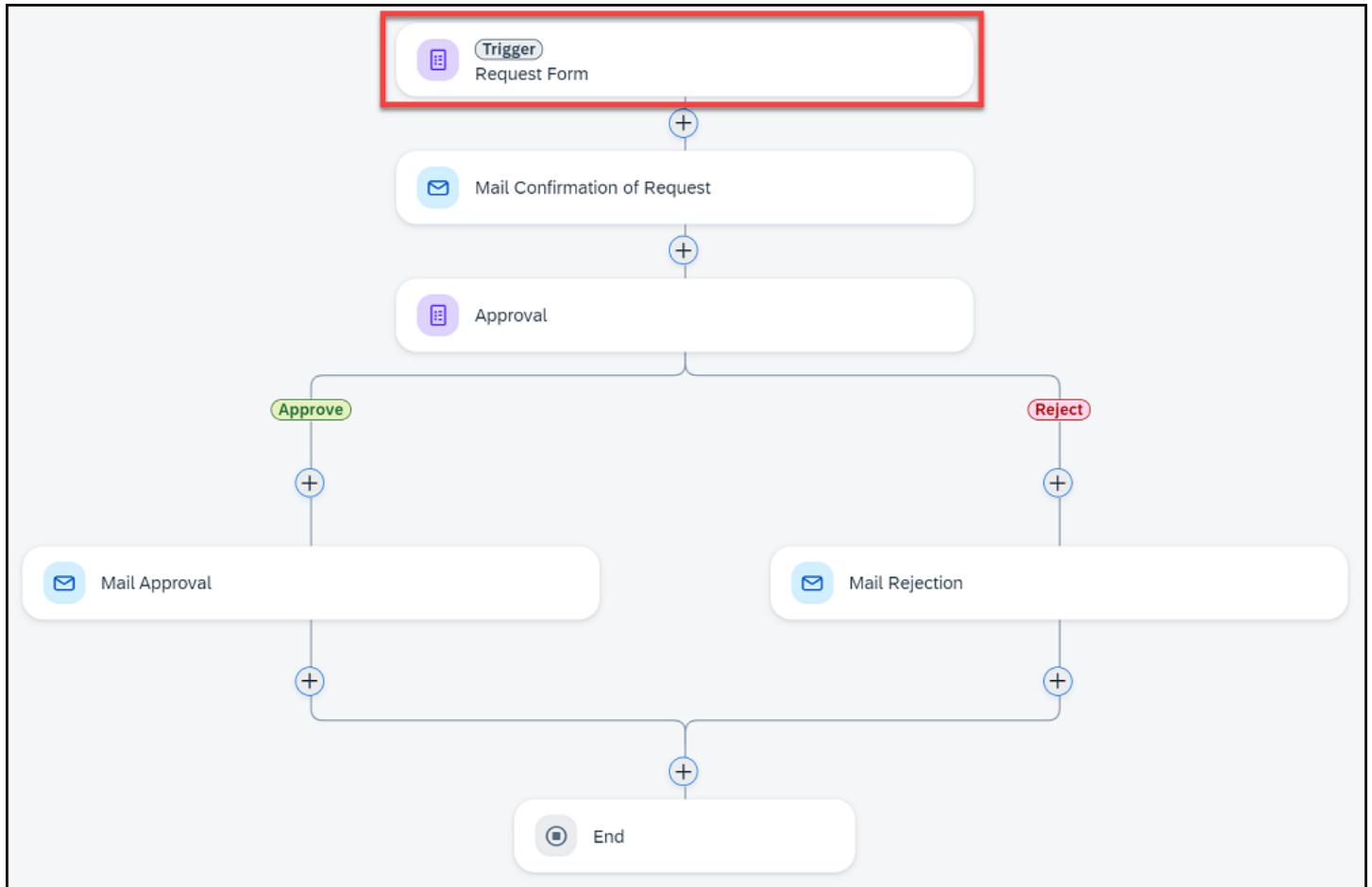
To monitor the business events, acquired from connected systems, see [Acquired Events](#).

Create a Form

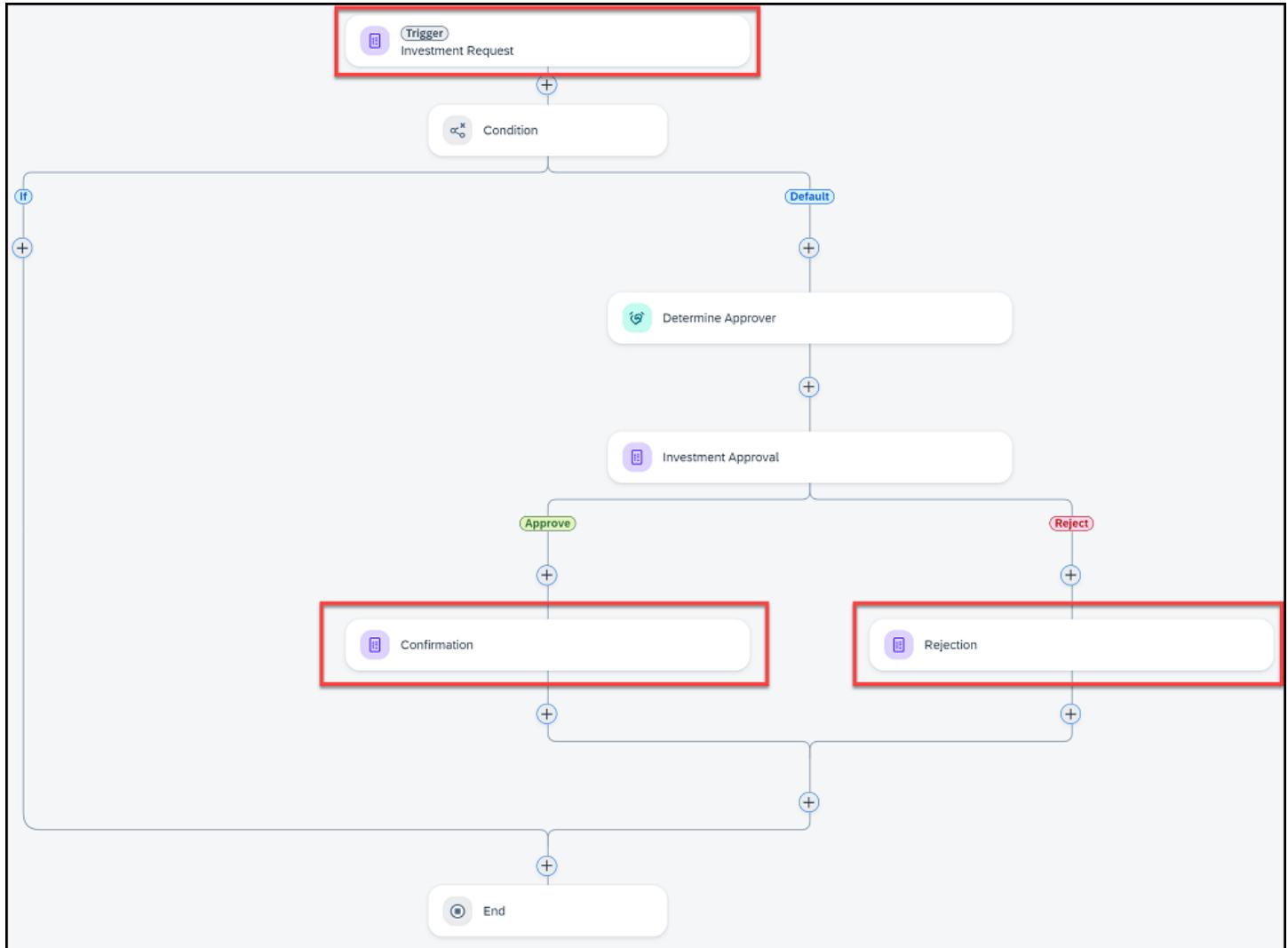
You can create interactive forms in SAP Build Process Automation, allowing you to capture and share information during a running process. Forms can then be a start trigger for a process and added as additional steps in the same process.

Context

In the following simple request process, a request submission form starts the process running. The requestor can access the published request form using a direct link or using a configured file in their SAP BTP.

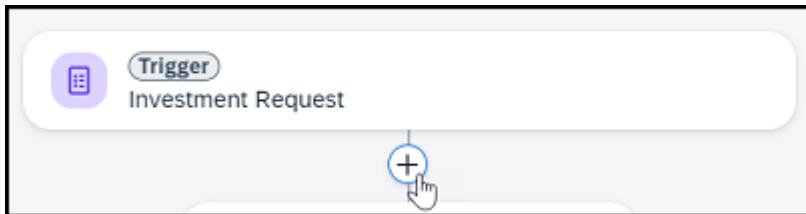


And in the following investment approval request, a request submission form also starts the process running. The requestor can access the published request form using a direct link or using a configured file in their SAP BTP. Then additional confirmation and rejection forms are included in the process, which will be displayed to the recipients as a task in their inbox.



Procedure

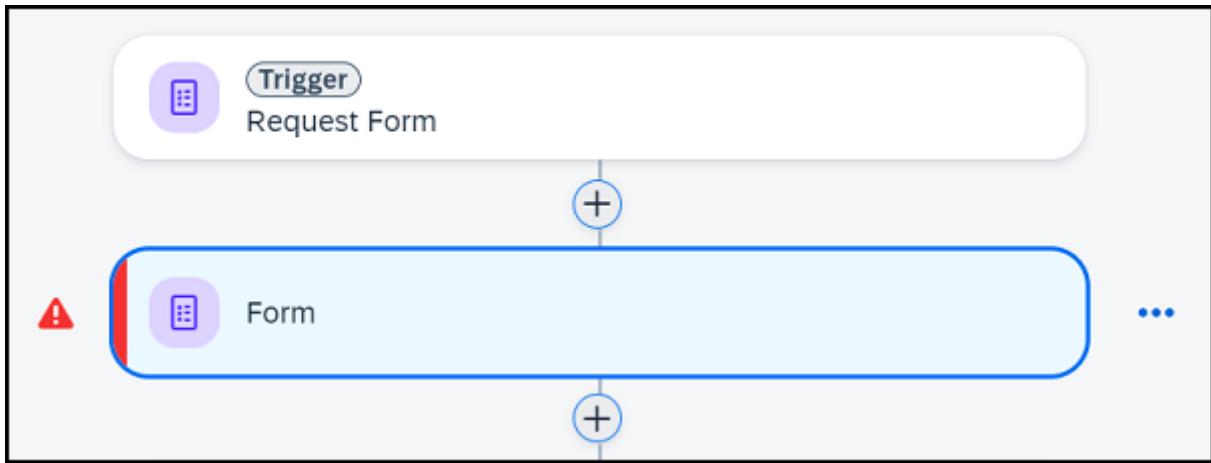
1. To create another form in the business process editor, choose **Form > Blank Form**.



2. Enter a **Name**, **Identifier** (used as an internal reference), and an optional **Description**.

3. Choose **Create**.

The form is created and added to the process editor canvas.



4. Double click and open the **Trigger** or **Form** element on the process editor canvas.
5. Drag and drop **Layout** and **Input** fields to the form as required. For an overview of form fields, see [Form Input Fields](#).

The screenshot shows the SAP Build Process Automation interface. The top navigation bar includes 'SAP Build Process Automation', 'Request Project' (which is the active tab), and 'Editable'. Below the navigation is a toolbar with icons for Overview, Request Process, Request Form, Release, and Save. The main content area is titled 'Request Form' and contains the following fields:

- Name ***: Text input field with placeholder 'Enter Text'.
- Office Location ***: Text input field with placeholder 'Enter Text'.
- Item Required ***: Text input field with placeholder 'Enter Text'.
- Enter the Reason**: Text input field with placeholder 'Enter Text'.
- Order Amount**: Text input field with placeholder 'Enter a Value' and a '#' symbol.
- Delivery Date**: Text input field with placeholder 'Select a Date'.

A red box on the left side of the screen highlights the sidebar where various input field types are listed, such as H1, H2, AA, T, E, etc.

6. To add a table, drag and drop the **Table** element to the form and enter a name.
- a. To add a column, choose **+** (Plus). For an overview of column types, see [Form Tables](#).

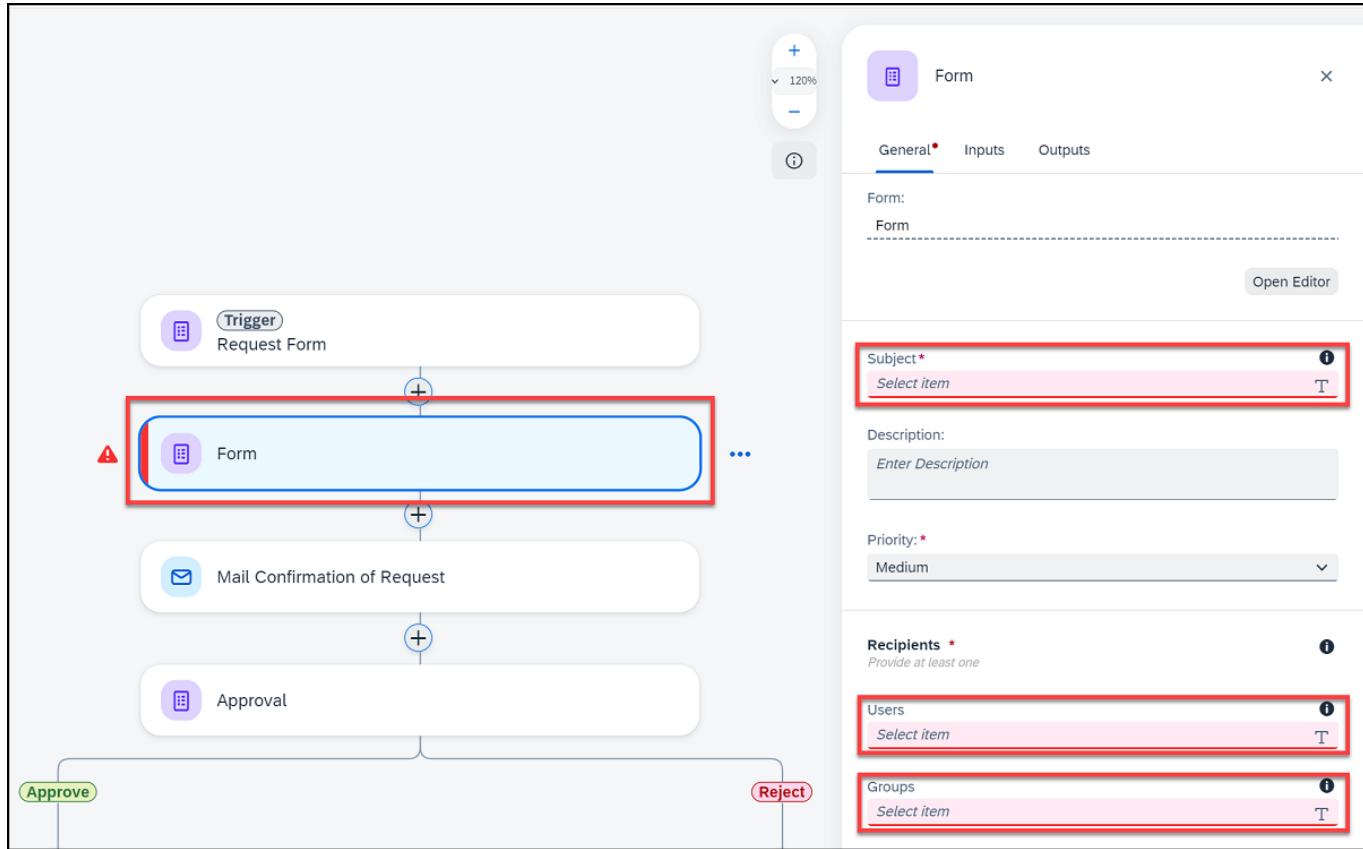
The screenshot shows the configuration of a table named 'Line Item Details'. The table has four columns with headers: 'Description *', 'Quantity *', 'Amount', and 'Expected Delivery Date'. Each column contains a text input field with placeholder text ('Enter Text', 'Enter a Value', 'Enter a Value', 'Select a Date') and a corresponding unit indicator ('T', '#', '#', 'Select'). A red box highlights the '+' button in the bottom right corner of the table header, which is used to add new columns.

7. **Optional:** Customize the default **Submit** outcome button and use **Next**, **Send**, or a **Custom** text. Choose the cogwheel icon, and select the required text from the **Button Title** dropdown.

If you select **Custom**, enter your own text. This text is then only available in the language in which you entered it. The predefined labels are translated. If several labels are translated using the same term in the target language, then the application only shows one label.

8. Save your changes.

9. Return to the process editor canvas and click on the form title, opening the form settings side panel.



10. Configure the **Form Settings** as required. See [Configure Settings for Forms and Approval Forms](#).

11. Save your changes.

Form Input Fields

You can add form input fields to both a standard form and an approval form, allowing you to design and manage the information displayed to the process participant.

Available Input Fields

In the form editor, you can directly edit the form name. You can drag and drop the following input fields:

Field	Available Values or Options
Headline 1	Enter text. You can move the element up and down.
Headline 2	
Paragraph	Enter text. You can move the element up and down.
Text (maximum 256 characters)	<ul style="list-style-type: none"> Minimum character limit Maximum character limit Read Only Required Description

Field	Available Values or Options
Text Area (maximum 1700 characters)	<ul style="list-style-type: none"> • Minimum character limit • Maximum character limit • Read Only • Required • Description
Dropdown	<p>Use one of the following options to define the dropdown entries:</p> <ul style="list-style-type: none"> • Use data from a data set by selecting an action from the library. When a data source is value help enabled, then the dropdown also shows the search icon and a popup for the search entry. See Add and Use Data Sources in Form Input Fields. ◦ Required ◦ Description • Enter text to define the options manually. ◦ Required ◦ Read Only ◦ Multiple Selection ◦ Description
Choice	<ul style="list-style-type: none"> • Enter text for the selection option entries • Vertical Alignment • Read Only • Required • Description
Checkbox	<ul style="list-style-type: none"> • Read Only • Required • Description
Number	<ul style="list-style-type: none"> • Fixed decimal places • Minimum value • Maximum value • Read Only • Required • Description
Date	<ul style="list-style-type: none"> • Based on the user's location • Allow selection of: All Dates, Only Past Dates, and Only Future Dates • Read Only • Required • Description

Field	Available Values or Options
File Upload / Attachment Requires an active SAP Document Management Service subscription. See Configure SAP Document Management Service for Process Attachments .	<ul style="list-style-type: none"> • Maximum number of files • Required • Read Only • Description
Tables	See Form Tables
Link	<p>Enter a link address of type HTTPS, HTTP, or mailto.</p> <p>Enter a link text to display instead of the URL, a label, and a description that appears in a question mark tool tip.</p> <p>Your input is the default link content. If you use a form that contains a link, the link content can be overwritten by other process content because the link field is available for mapping.</p>

Input Field Settings

You can directly edit input field texts. To move the field up or down or to delete it, use the **Menu** icon.



The **Configuration** section, let's you define more settings. Depending on the field type, you can, for example, mark the field as:

- **Required** - The participant must input data here to progress the active process.
- **Read Only** - The participant can see information entered earlier in the process, but can't edit it.
- **Multiple Selection** - The participant can see select more than one entry.

The **Input Validation** section, let's you define whether and which validation is used for text fields and text areas. When you select an option, the regular expression for that selection is displayed. You can copy it to create a more sophisticated custom expression that you enter as a custom validation. You can directly test the validation you defined using the **Test Text** field. For examples on regular expressions, see, for example, the [Mozilla cheat sheet](#).

The following validation options are available:

- **Letters** - The participant can enter letters only.
- **Numbers** - The participant can enter numbers only.
- **Letters and Numbers** - The participant can enter any character.
- **Alphanumeric** - The participant can use latin letters (a-z, A-Z) and arabic numbers.
- **Custom Validation** - You can enter a regular expression that exactly defines what the participant can enter. For example, you can require three letters, one number....

Add and Use Data Sources in Form Input Fields

You can add data sources to your form input fields, allowing process participants to select from information managed in external systems. This removes the need to manually add fields and information when creating a form.

Prerequisites

Before you can add and use data sources in your forms, make sure that the data sources have been configured in your library.

See [Configure Data Sources for Form Input Fields](#).

i Note

- Only JSON files are supported and the file size is limited to 5 MB.
- Open API specification files with versions 2.x.x and 3.x.x of JSON type are supported.
- Must include a **GET** request with an array.
- Must not have mandatory input parameters defined.
- To make an action available as a data source, define its output by setting the **Main Output Array** tag.
- To use the search feature for a data source, the action must have the \$search parameter defined.

Context

Adding data sources for form input fields allows process participants to select from information shared from external systems. These fields can then be mapped across process steps, ensuring that process data is consistent.

As an example, a dropdown field in the following form is configured to allow process participants to select from customer data shared from SAP S/4HANA. In this case, they can select the customer's country, ID, address, and the company name.

New Dropdown ⋮ X

▽ Data to display

From data set **Manual definition**

Data Source *

Get entities from Custom...

Destination Variable *

Data_S4HANA

Available Data *

ID	CompanyName	S...	⊕
<input type="checkbox"/>	Fax string		
<input type="checkbox"/>	City string		
<input type="checkbox"/>	Phone string		
<input type="checkbox"/>	Region string		
<input checked="" type="checkbox"/>	Address string		
<input checked="" type="checkbox"/>	Country string		
<input checked="" type="checkbox"/>	CustomerID string		
<input type="checkbox"/>	PostalCode string		
<input checked="" type="checkbox"/>	CompanyName string		

i Note

You can only use dropdown fields.

The list shows the labels of the output values that are defined for the action. These labels can be changed in the action editor.

Procedure

1. In the form editor, drag and drop a **Dropdown** input field to the canvas.
2. In the side pane under **Data to display**, choose **From data set**.

New Dropdown : X

▼ Data to display

From data set Manual definition

Option 1

Option 2

Configuration

Required

Read Only

Multiple Selection

3. Select a library under **Data Source**.

New Dropdown : X

▼ Data to display

From data set Manual definition

Data Source *

X

Configuration

Required

Add Description

4. Once you've found the **Action** to use, choose **Add**.

Browse library

Supplier

Action Type	Projects
GET X	X

🔗 Action

Get entities from Suppliers

Project: Northwind actions for Forms Value... Add

5. Select or create a **Destination Variable**. Destination variables are used to connect the action to an external system.
6. Select the **Available Data** that you want to display to the process participant in this field.

New Dropdown ⋮ X

⌄ Data to display

From data set Manual definition

Data Source *

Get entities from Suppliers X Delete

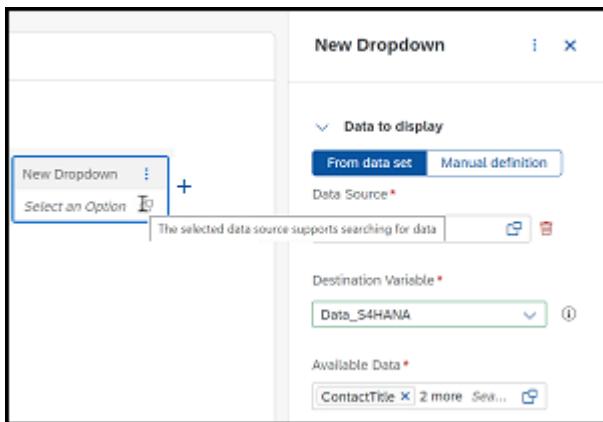
Destination Variable *

Data_S4HANA i

Available Data *

CompanyName X 2 more S... X

If the the \$search parameter is defined, the icon in the dropdown changes:



7. Save your changes.

Form Tables

As with form input fields, you can add tables to both a standard form and an approval form, allowing you to design and manage the information displayed to the process participant.

You define the columns of the table, and the process participant adds content with as many rows as needed.

If you create a form based on a form with a table, then you can map those fields as input of the new form using [Bind List](#).

i Note

You must bind the table to a data type list of type **Object** and the **Required** constraint, even for single-column tables of type text with string input. Lists with other data types, for example, strings, numbers, or booleans don't work.

The table in the new form is set to read only by default but you can change that setting.

The following general rules apply:

- The number of columns isn't limited.
- The number of rows can't be predefined.
- The **Read Only** setting applies to the entire table and can't be used for individual columns.

The following columns elements are available:

Column	Available Values or Options
--------	-----------------------------

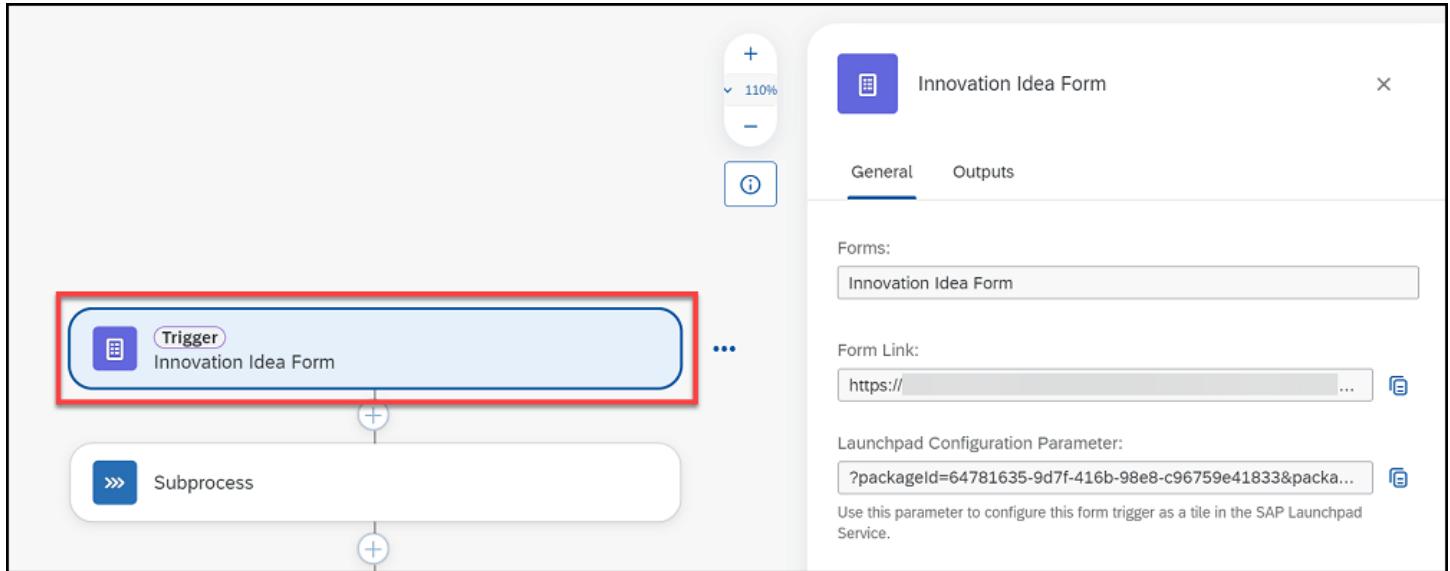
Column	Available Values or Options
Text (maximum 256 characters)	<ul style="list-style-type: none"> • Minimum character limit • Maximum character limit • Required • Description
Checkbox	<ul style="list-style-type: none"> • Required • Description
Number	<ul style="list-style-type: none"> • Fixed decimal places • Minimum value • Maximum value • Required • Description
Date	<ul style="list-style-type: none"> • Based on the user's location • Allow selection of: All Dates, Only Past Dates, and Only Future Dates • Required • Description
Dropdown	<p>Use one of the following options to define the dropdown entries:</p> <ul style="list-style-type: none"> • Use data from a data set by selecting an action from the library. When a data source is value help-enabled, then the dropdown also shows the search icon and a popup for the search entry. See Add and Use Data Sources in Form Input Fields. • Enter text to define the options manually. <ul style="list-style-type: none"> ◦ Required ◦ Description <p>i Note</p> <p>If you use a dropdown in the table, you can no longer set the whole table to read only.</p>

Configure Settings for Forms and Approval Forms

You can configure the settings for your forms using the process editor, allowing you to define how the form is displayed to the process participant.

Settings for Start Trigger Forms

You can configure your process start trigger settings by clicking on the **Trigger** tile.



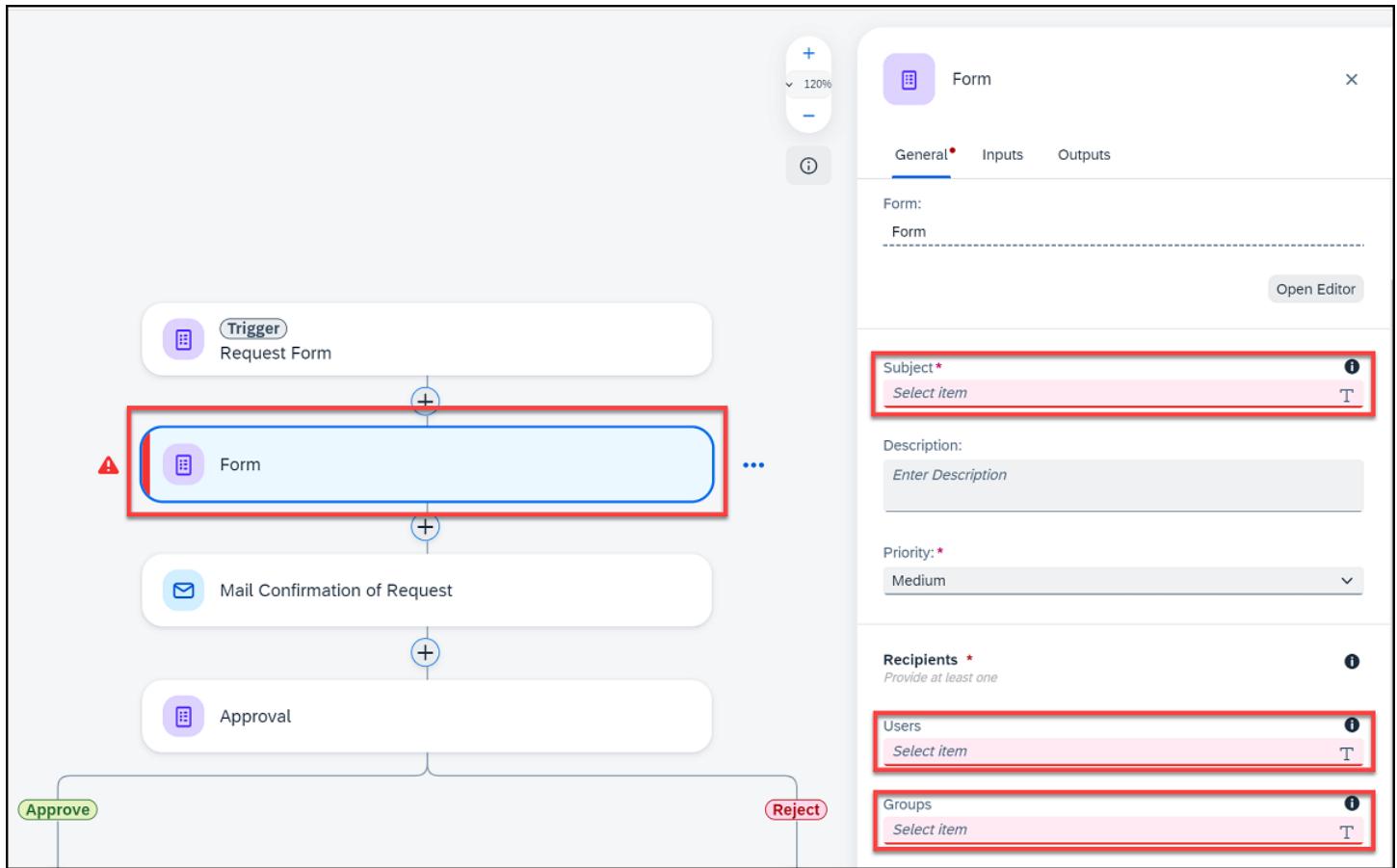
You have the following configuration options:

Field	Description
Forms	Select the form you want to use as the trigger for your process. Only forms available in the same business process project can be selected.
Form Link	Available for deployed processes only, this provides a direct link to the published form. The process participant can access this link, enter the relevant information, and then click submit.
Launchpad Configuration Parameter	Available for deployed processes only, this provides the parameters needed to create a tile in SAP Build Work Zone. Clicking the tile opens the form and from there the process participant can enter the relevant information and then click submit.
Outputs	This section provides an overview of the process metadata included in the trigger form. This is a read-only section.

Additional Settings for Forms or Approval Forms

You can configure additional form or approval form settings for forms visible on the process editor canvas only, rather than all forms in the project. These form settings allow you to configure how the form information is displayed as a task in the inbox and who that task is sent to.

To configure the form settings, click on the form element on the process editor:



You have the following configuration options:

Field	Description
Step Name	This is the name under which the form displays in the process canvas.
<Name of the form>	This is the name you entered for the form when creating it.
Subject	This is the subject of the task received in the inbox, allowing the participant to identify the item. In the following example, this is displayed as: Purchase Requisition - Approval .
Description	This provides the description displayed within the received task, giving the participant more information about the process. In the following example, this is displayed as: Please approve or reject following request .
Priority	This defines the priority of the task received in the inbox, giving an indication of when a response is needed. In the following example, this is displayed as High .
Recipients	You can assign process metadata and context fields, for example, lists of users or the user who started the process. Alternatively you can enter specific email addresses of users or groups who should receive the task. For more detailed guidelines, see Guidelines for Specifying Recipient Users .
Allow Forwarding	Select to allow the recipient to forward the task to another user.

Field	Description
Due Date	<p>You can set a due date for the task received in the inbox, giving a deadline for the response from the time that the task arrives. Due dates can be set to minutes, hours, days, months, and yearly durations.</p> <p>Choose one of the following due date options:</p> <ul style="list-style-type: none"> • No Due Date • Static Duration <p>Enter an integer between 1 and 999, and select the unit (minutes, hours, days, months, and years).</p> <ul style="list-style-type: none"> • Dynamic Duration <p>Select a number input between 0 and 2147483647 from the process content.</p> <ul style="list-style-type: none"> • Reference Date <p>Select a date input (UTC + 0) from the process content.</p>
Inputs	You can map the process metadata fields here, ensuring that the correct information persists throughout the process.
Outputs	This section provides an overview of the process metadata included in the form. This is a read-only section.

Example of a Task

The following is an example of a task received to the inbox, with the fields taken from the form settings configured using the process editor.

The screenshot shows the SAP Fiori task inbox interface. On the left, there is a sidebar titled "All Tasks (1)" with a search bar containing "tom." and a refresh button. A single task card is visible, titled "Purchase Requisition Approval" with a priority level of "High". The main content area is titled "PR Approval Form" and contains the following details for the task:

Purchase Requisition - Approval

Please approve or reject following request.

Requestor Name: Tom

Material: IPHONE-12

Quantity: 1

Requestor Comments: Can you please approve this request?

Manager Comments:

Guidelines for Specifying Recipient Users

When you specify recipient users for a user task, consider the following:

i Note

Carefully consider the impact that the changes described here might have on your overall scenario. Changing certain settings after productive use has started can have a negative impact on scenarios that are incompatible with the change. If

applicable, use mechanisms that restrict the impact to the specific scenario.

- Evaluate whether you can use "recipient groups" instead of "recipient users" because there are limits as to how many recipient users may be specified. If you can, you must configure the assignment of users to groups in the identity management-related function of the platform or the identity management back-end systems. This has, for example, the benefit that the assignment of a task to a certain user can be removed using these central identity management functions instead of in the workflow definition and related locations. This usually improves compliance with company and legal requirements. For example, removing an assignment typically becomes effective as soon as authentication tokens expire.
- If you cannot avoid specifying user names using constants or expressions, make sure that you apply the necessary lifecycle actions on the respective events to achieve compliance. For example, use the administrative REST APIs of SAP Build Process Automation, to remove recipient users when they should no longer be assigned to a task because they left their department or the company. Also, ensure that user interfaces that allow configuration of user IDs apply appropriate validation on the task recipients.
- Ensure that the case and spelling of the user ID matches the respective fields of the authentication tokens exactly. It is important that lower or upper case is also considered, because SAP Build Process Automation matches them as is. SAP Build Process Automation must also consider case sensitivity for user names that look like email addresses. There is no metadata that indicates whether user names are actual email addresses or whether case sensitivity is irrelevant. For this purpose, check your identity management system and the related configuration of SAP Business Technology Platform.
- Check the **User ID Source** and its related settings. Consider using "E-Mail" as the configured value, because this might improve consistency of user names in a scenario.
- Ensure that the identifiers, as validated against the user database, are provided. Do not rely, for example, on user names as entered directly by the user.
- Evaluate whether you can disable the creation of "shadow users". In certain constellations, this prevents users from logging in with user names that do not correspond to the canonical identifier, but use a different case.
- Evaluate whether you can configure that user inputs are automatically converted to the expected case. If you can, see the documentation of the SAP Cloud Identity Services feature **Apply Function to Subject Name Identifier** in [Convert Subject Name Identifier to Uppercase or Lowercase](#) or the respective configuration of your custom identity management system.

Configure a Start Trigger Form with SAP Build Work Zone

Using this trigger, you can start a business process.

Prerequisites

You have set up SAP Build Work Zone. See [Configure SAP Build Work Zone for SAP Build Process Automation](#) and particularly [Configure SAP Build Work Zone Content](#).

Procedure

- Start the Site Manager:
 - In your SAP BTP Cockpit subaccount, choose **Services** **Instances and Subscriptions**.
 - Search for your SAP Build Work Zone application and choose the **Go to Application** icon next to the app name.
- Open **Content Manager** **My Content**.
- Create a local copy of the **Process Trigger** app by selecting the **Process Trigger** item and choosing **Create a Local Copy**.

4. To configure the local copy, choose **Edit**

- Enter a title.
- On the **Navigation** tab, enter a parameter named **uri**.

To do so:

- In the process editor of SAP Build Process Automation, select the trigger and on the Trigger Settings panel, copy the configuration parameter URL.
 - Paste the configuration parameter URL into the **Default Value** field.
- On the **Translation** tab, edit any translated texts as needed.
 - Save your changes.

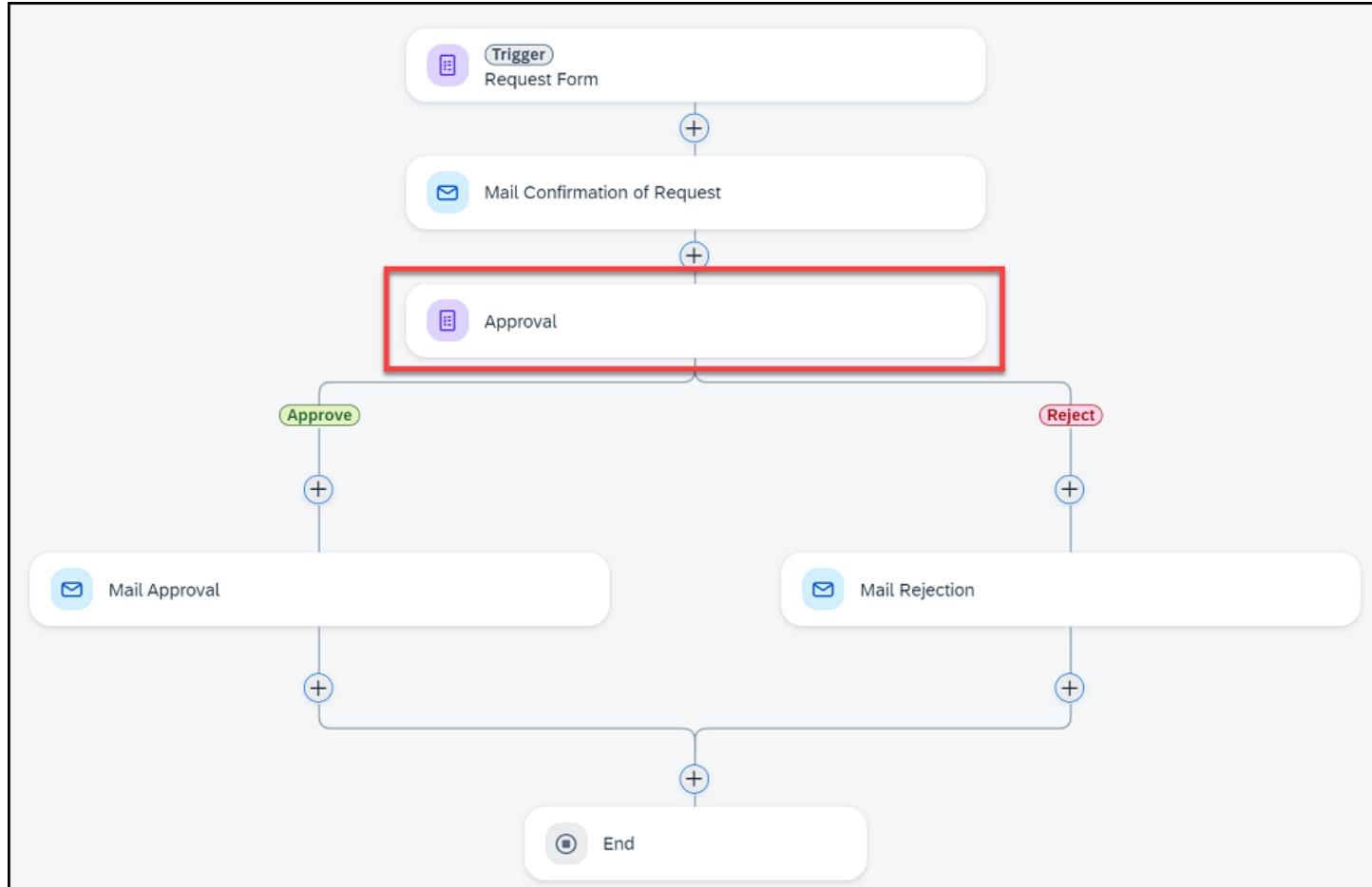
5. Assign the app to your users using standard means of SAP Build Work Zone.

Create an Approval Form

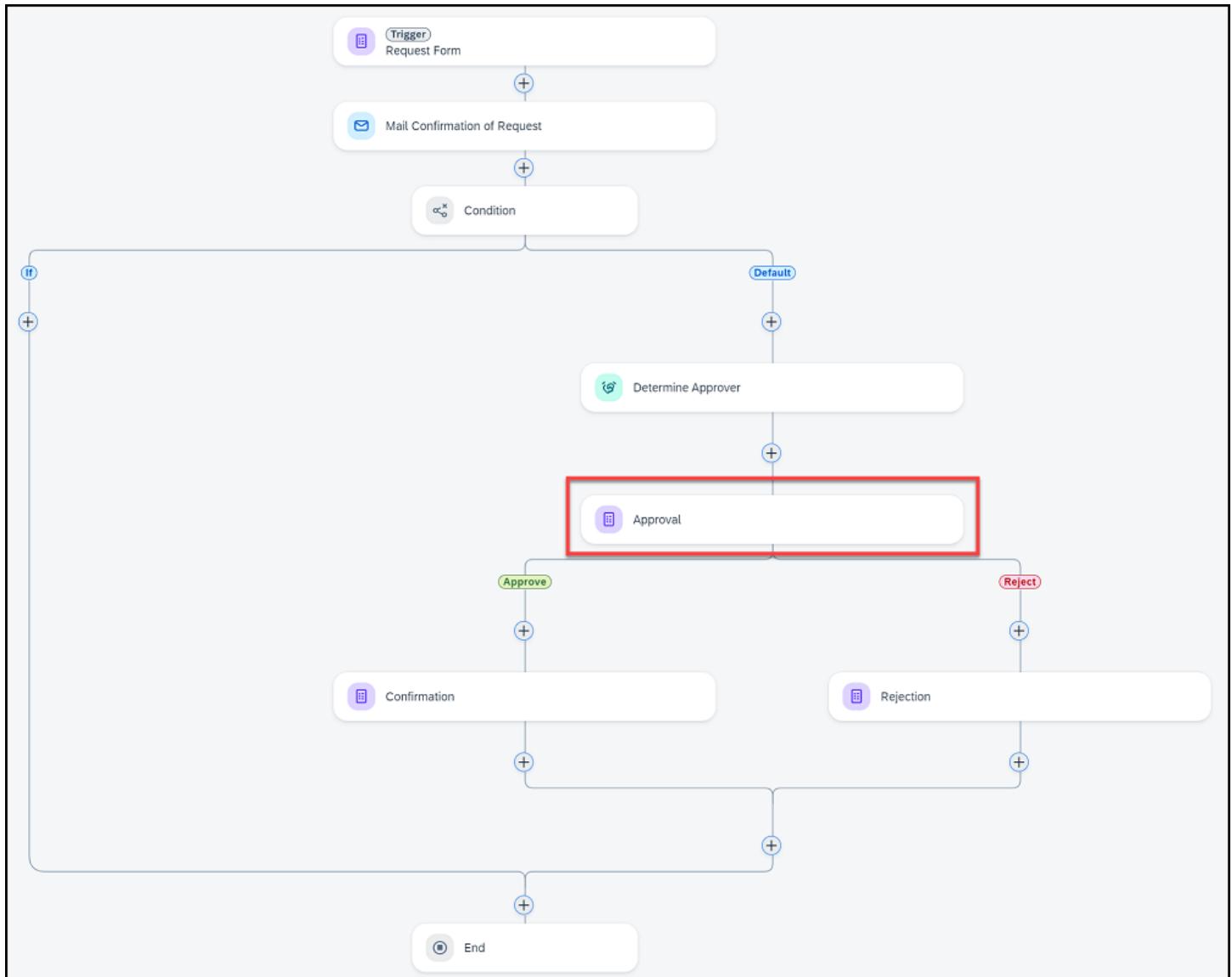
Approvals are an important part of business processes, whether performed manually or automatically approved based on process conditions. With SAP Build Process Automation, you can manage approvals by creating and adding an approval form to a business process.

Context

In the following simple request process, an approval is needed for all submitted requests:



And in the following investment approval request, an approval is only needed when an investment value is higher than a specified amount:



When a process is running, the approver receives a task in their inbox containing the request information. They can then choose to accept, query, or reject the request. Their decision then determines how the process proceeds.

The following is an example of a task received to the inbox, with the fields taken from the form settings configured using the process editor.

All Tasks (1)	
tom.	X
REFRESH	
Purchase Requisition Approval High	
PR Approval Form <h2>Purchase Requisition - Approval</h2> <p>Please approve or reject following request.</p> <p>Requestor Name Tom</p> <p>Material IPHONE-12</p> <p>Quantity 1</p> <p>Requestor Comments Can you please approve this request?</p> <p>Manager Comments <input type="text" value="Enter text"/></p>	

Procedure

1. From the process editor canvas, click **+** and select **Approval - New Approval Form**.
2. Enter a **Name**, **Identifier**, and optionally a **Description**.
3. Choose **Create**.

The approval form element is now added to your process editor canvas.

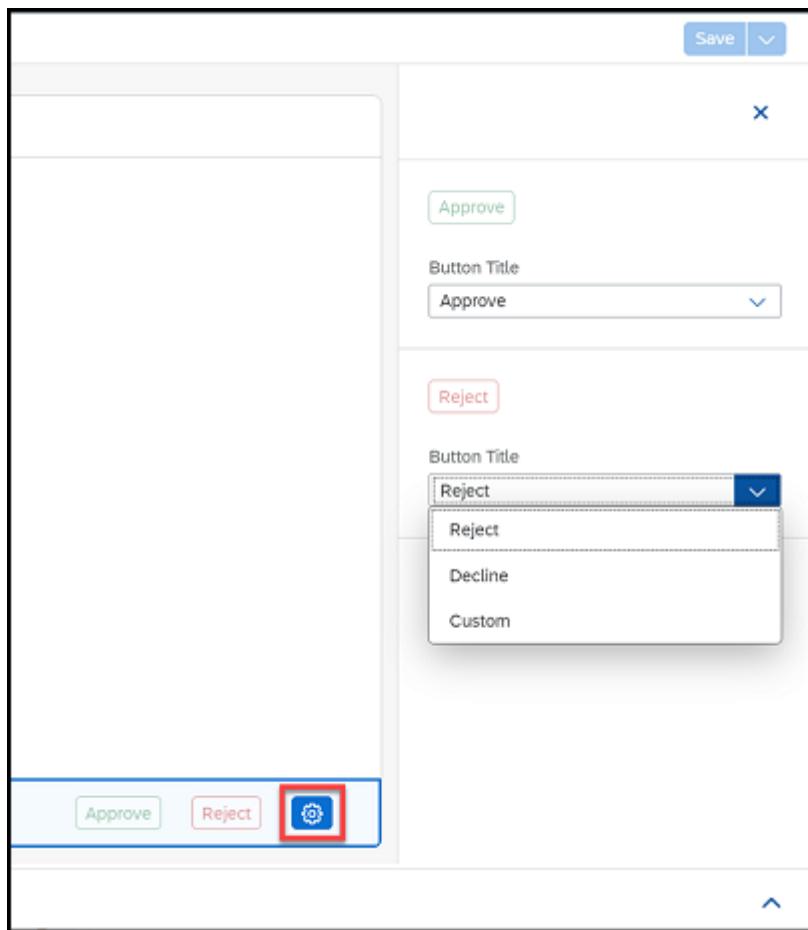
4. Open the form editor by double-clicking the approval form element on the canvas.

5. Configure your approval form as required using **Layout** and **Input Fields**.

For more information about input fields, see [Form Input Fields](#).

6. **Optional:** Customize the default outcome buttons. You can adapt the default **Approve** button to **Accept** or a **Custom** text, and the default **Reject** button to **Decline** or a **Custom** text. Choose the cogwheel icon, and select the required text from the **Button Title** dropdown.

If you select **Custom**, enter your own text. This text is then only available in the language in which you entered it. The predefined labels are translated. If several labels are translated using the same term in the target language, then the application only shows one label.



7. Save your changes.

The approval form has now been saved to your process.

8. Navigate back to the process editor.

9. Click on the approval form element on the canvas, opening the approval settings menu.

10. Configure your approval settings as needed.

For more information about configuring approval settings, see [Configure Settings for Forms and Approval Forms](#).

11. Save your changes.

The configured approval form is added to your business process.

Import an SAPUI5 Form

Import an SAPUI5 form and use it in your process in SAP Build Process Automation. This allows you to create SAPUI5 apps and use them as task UIs in your process.

Prerequisites

- You've created an SAPUI5 application. For more information about creating an application, see [Creating a Custom Task UI](#).
- You've added the namespace "sap.bpa.task" to the manifest.json for your application and you've defined the inputs, outputs, and outcomes for the application. For more information, see [Technical Information for Adapting the SAPUI5 Application](#).
- You've adapted the Component.js file and the App.view.xml file. For more information, see [Technical Information for Adapting the SAPUI5 Application](#).
- You've deployed your application. For more information, see [Build and Deploy](#) in the documentation for SAP Business Application Studio.

Context

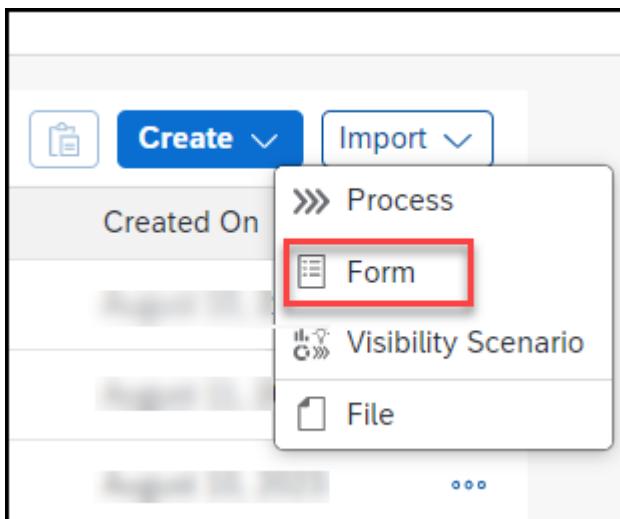
When you import a SAPUI5 form into SAP Build Process Automation, you can use this SAPUI5 app in the process editor in the same way as you use any other form. You can map the inputs and outputs of your form to the rest of the process and you can also use the outputs of the previous task in your form.

i Note

You cannot use an SAPUI5 form as a start form.

Procedure

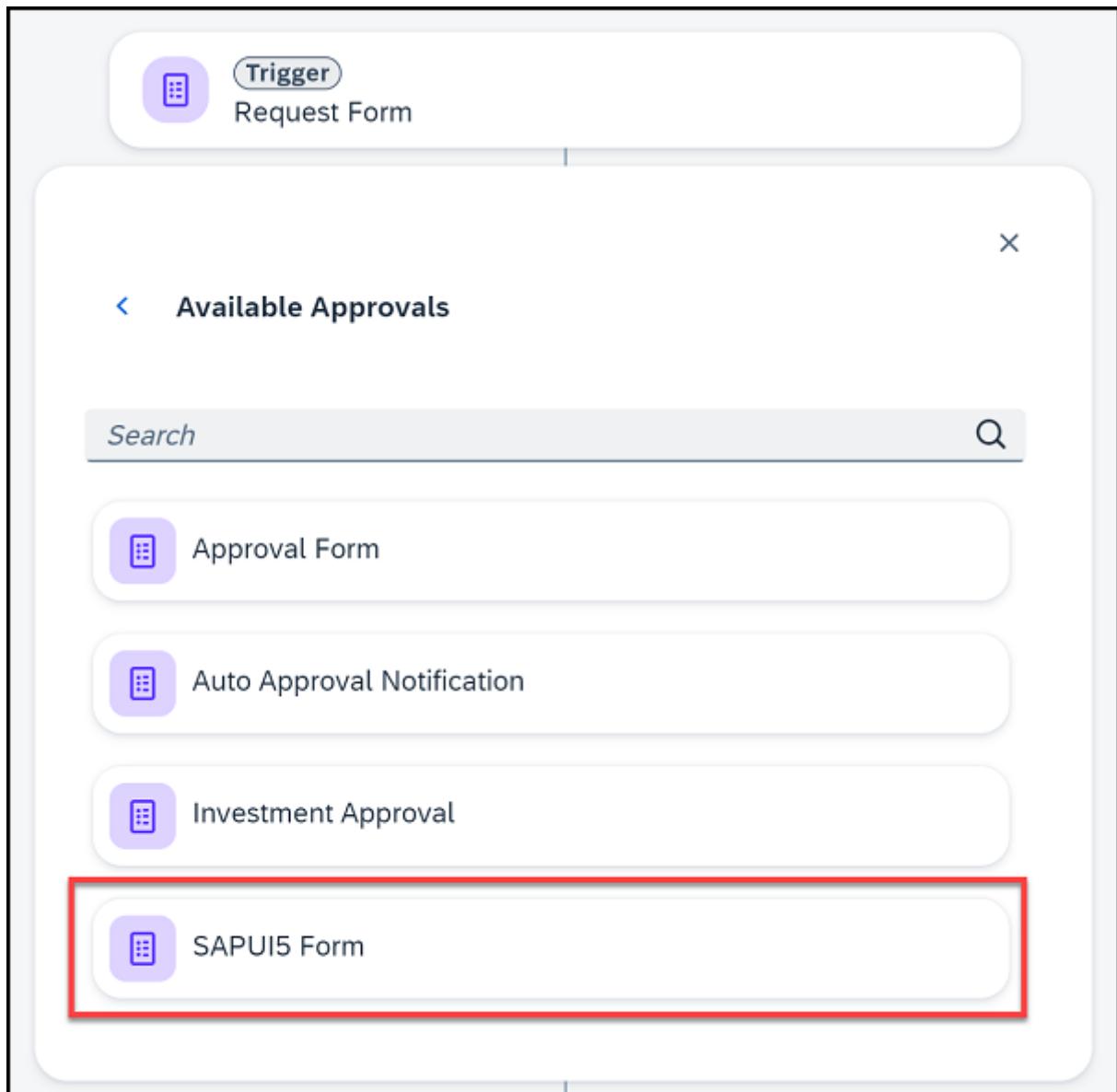
1. In the Overview tab of your project, choose **Import > Form**.



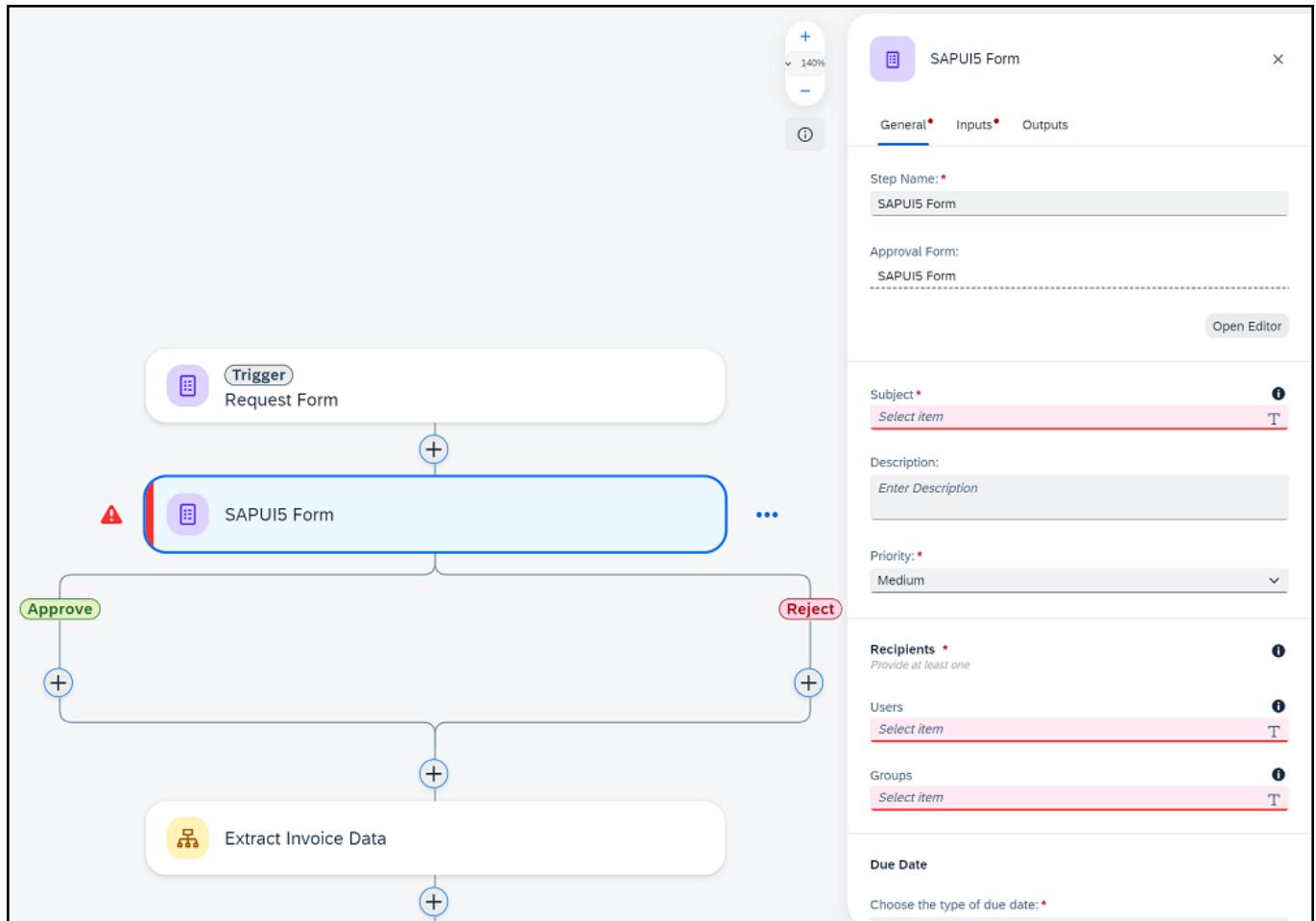
2. In the **Import Form** screen, enter the **Application ID** and the **Application Version** from the manifest.json.

The form is imported and the manifest URL is displayed. You can now add the form to your process.

3. From the process editor canvas, choose **+**, then **Form** or **Approval** (depending on the category of the form) and choose your SAPUI5 form.



4. Choose the form element on the canvas to open the form settings.



5. Configure your form settings as required. For more information, see [Configure Settings for Forms and Approval Forms](#).

6. Save your changes.

The configured SAPUI5 form is added to your business process.

Technical Information for Adapting the SAPUI5 Application

To use SAPUI5 applications as forms in SAP Build Process Automation, you need to add the "sap.bpa.task" namespace to the manifest.json file. You also need to adapt the Component.js file and the App.view.xml file in accordance with the manifest.json. The examples shown here are all compatible with each other.

Configuring the manifest.json File

Add the namespace "sap.bpa.task" to the manifest.json for your application. The following is an example:

❖ Example

```
{
  ...
  "sap.bpa.task": {
    "_version": "1.0.0",
    "outcomes": [
      {
        "id": "approve",
        "label": "Allow"
      }
    ]
  }
}
```

```

},
{
  "id": "reject",
  "label": "Deny"
}
],
"inputs": {
  "$schema": "http://json-schema.org/draft-07/schema",
  "title": "input",
  "type": "object",
  "required": ["newText"],
  "properties": {
    "newText": {
      "type": "string",
      "title": "Textfield",
      "description": "Description for Textfield"
    }
  }
},
"outputs": {
  "$schema": "http://json-schema.org/draft-07/schema",
  "title": "output",
  "type": "object",
  "required": ["newText"],
  "properties": {
    "newText": {
      "type": "string",
      "title": "Textfield",
      "description": "Description for Textfield"
    }
  }
},
"category": "approval"
}
}

```

The namespace must have the following properties:

- The `_version` must be the namespace version.
- The `category` can be either `standard` or `approval`, depending on the type of form you want to use in SAP Build Process Automation. The outcomes depend on the category you choose.
- The outcomes can be an array of objects that have an `id` and a `label`. The `id` can be `submit`, `approve` or `reject`. You can enter any text for the `label`.
 - If the `category` is `standard`, the outcome must be `[{id:"submit", label: "AnyLabel"}]`
 - If the `category` is `approval`, the outcome must be `[{id:"approve", label:"SomeLabel"}, {id:"reject", label: "SomeLabel2"}]`.

i Note

The outcomes you define in the `manifest.json` determine the actions that the application has to create for [My Inbox](#).

- The inputs must be a draft-7 JSON schema and define the context object you receive with the [Workflow API for Cloud Foundry](#). For more information, see the [Inputs and Outputs supported by SAP Build Process Automation Forms](#) section below.
- The outputs must be a draft-7 JSON schema and define the context object you have to send with the [Workflow API for Cloud Foundry](#). For more information, see the [Inputs and Outputs supported by SAP Build Process Automation Forms](#) section below.

Configuring the Component.js File

Once you've created your SAPUI5 application, you need to adapt the Component.js file.

Replace the definition of the Component.js file, which is the second parameter of UIComponent.extend, using the following example:

❖ Example

```
{
  metadata: {
    manifest: "json",
  },
  /**
   * The component is initialized by UI5 automatically during the startup of the app and can
   * @public
   * @override
   */
  init: function () {
    // call the base component's init function
    UIComponent.prototype.init.apply(this, arguments);

    // enable routing
    this.getRouter().initialize();

    // set the device model
    this.setModel(models.createDeviceModel(), "device");

    this.setTaskModels();
    const rejectOutcomeId = "reject";
    this.getInboxAPI().addAction(
      {
        action: rejectOutcomeId,
        label: "Deny",
        type: "reject",
      },
      function () {
        this.completeTask(false, rejectOutcomeId);
      },
      this
    );
    const approveOutcomeId = "approve";
    this.getInboxAPI().addAction(
      {
        action: approveOutcomeId,
        label: "Approve",
        type: "approve",
      },
      function () {
        this.completeTask(true, approveOutcomeId);
      },
      this
    );
  }
}
```

```

        label: "Allow",
        type: "accept",
    },
    function () {
        this.completeTask(true, approveOutcomeId);
    },
    this
);
},
setTaskModels: async function () {
    // set the task model
    var startupParameters = this.getComponentData().startupParameters;
    this.setModel(startupParameters.taskModel, "task");

    // set the task context model
    var taskContextModel = new sap.ui.model.json.JSONModel(
        this._getTaskInstancesBaseURL() + "/context"
    );
    this.setModel(taskContextModel, "context");

    // parse Date objects and set in own model
    await taskContextModel.loadData(this._getTaskInstancesBaseURL() + "/context");
},
_getTaskInstancesBaseURL: function () {
    return (
        this._getWorkflowRuntimeBaseURL() +
        "/task-instances/" +
        this.getTaskInstanceID()
    );
},
_getWorkflowRuntimeBaseURL: function () {
    var ui5CloudService = this.getManifestEntry("/sap.cloud/service").replaceAll(".", "")
    var ui5ApplicationName = this.getManifestEntry("/sap.app/id").replaceAll(".", "");
    var appPath = `${ui5CloudService}.${ui5ApplicationName}`;
    return `/ ${appPath}/api/public/workflow/rest/v1`;
},
getTaskInstanceID: function () {
    return this.getModel("task").getData().InstanceID;
},
getInboxAPI: function () {
    var startupParameters = this.getComponentData().startupParameters;
    return startupParameters.inboxAPI;
},
completeTask: function (approvalStatus, outcomeId) {
    this.getModel("context").setProperty("/approved", approvalStatus);
    this._patchTaskInstance(outcomeId);
},
_patchTaskInstance: function (outcomeId) {

```

```

const context = this.getModel("context").getData();
var data = {
  status: "COMPLETED",
  context: {...context, newText: context.newText || ''},
  decision: outcomeID
};

jQuery.ajax({
  url: `${this._getTaskInstancesBaseURL()}`,
  method: "PATCH",
  contentType: "application/json",
  async: true,
  data: JSON.stringify(data),
  headers: {
    "X-CSRF-Token": this._fetchToken(),
  },
}).done(() => {
  this._refreshTaskList();
})
},
}

_fetchToken: function () {
  var fetchedToken;

  jQuery.ajax({
    url: this._getWorkflowRuntimeBaseURL() + "/xsrf-token",
    method: "GET",
    async: false,
    headers: {
      "X-CSRF-Token": "Fetch",
    },
    success(result, xhr, data) {
      fetchedToken = data.getResponseHeader("X-CSRF-Token");
    },
  });
  return fetchedToken;
},
}

_refreshTaskList: function () {
  this.getInboxAPI().updateTask("NA", this.getTaskInstanceID());
},
}

```

Configuring the App.view.xml File

If you want to use a text input, replace the App element in your app.view.xml file as follows:

❖ Example

```

<App id="app">
  <Input id ="input" value = "{context>/newText}"></Input>
</App>

```

Inputs and Outputs supported by SAP Build Process Automation Forms

The following schema shows all property definitions a SAP Build Process Automation form can output or handle as input.

↳ Sample Code

```
{
  "$schema": "http://json-schema.org/draft-07/schema",
  "title": "<\"input\" or \"output\">",
  "type": "object",
  "required": [],
  "properties": {
    "newText": {
      "title": "New Text",
      "description": "",
      "type": "string",
    },
    "newTextArea": {
      "title": "New Text Area",
      "description": "",
      "type": "string",
    },
    "newNumber": {
      "title": "New Number",
      "description": "",
      "type": "number"
    },
    "newDate": {
      "title": "New Date",
      "description": "",
      "type": "string",
      "format": "date"
    },
    "newChoice": {
      "title": "New Choice",
      "description": "",
      "type": "array",
      "uniqueItems": true,
      "items": {
        "type": "string",
        "enum": ["Option 1", "Option 2"]
      }
    },
    "newChoice2": {
      "title": "New Choice",
      "description": "",
      "type": "string",
      "enum": ["Option 1", "Option 2"]
    },
    "newDropdown": {
      "title": "New Dropdown",
      "description": ""
    }
  }
}
```

```
"type": "string",
"enum": ["Option 1", "Option 2"]
},
"newCheckbox": {
    "title": "New Checkbox",
    "description": "",
    "type": "boolean"
},
"newTable": {
    "title": "New Table",
    "description": "",
    "type": "array",
    "items": {
        "$ref": "#/definitions/newTableItems"
    }
}
},
"definitions": {
    "newTableItems": {
        "type": "object",
        "required": [],
        "properties": {
            "newText": {
                "title": "New Text",
                "description": "",
                "type": "string"
            }
        }
    }
}
}
```

Update the xs-app.json File

In the `xs-app.json` in your UI module folder (the same folder that also contains the webapp), you must replace the following code snippet:

☰ Sample Code

```
{  
  "source": "^/bpmworkflowruntime/(.*)$",
  "target": "/public/workflow/rest/$1",
  "service": "com.sap.spa.processautomation",
  "endpoint": "api",
  "authenticationType": "xsuaa"
},
```

Use it with the following code snippet instead. Make sure that it is the first entry of the routes array.

```
{  
  "source": "^/api/(.*)$",  
  "target": "$1",  
  "service": "com.sap.spa.processautomation",
```

```
  "endpoint": "api",
  "csrfProtection": true,
  "authenticationType": "xsuaa"
},
```

Troubleshooting

When you build your application, if you receive an error message stating that your UI5 CLI installation is outdated, update the `@ui5/cli` version in the package `.json` file of the UI module. You can use versions from version 3 onwards, for example `"^3.4.0."`.

Add Flow Controls

You can add and configure controls to your business process.

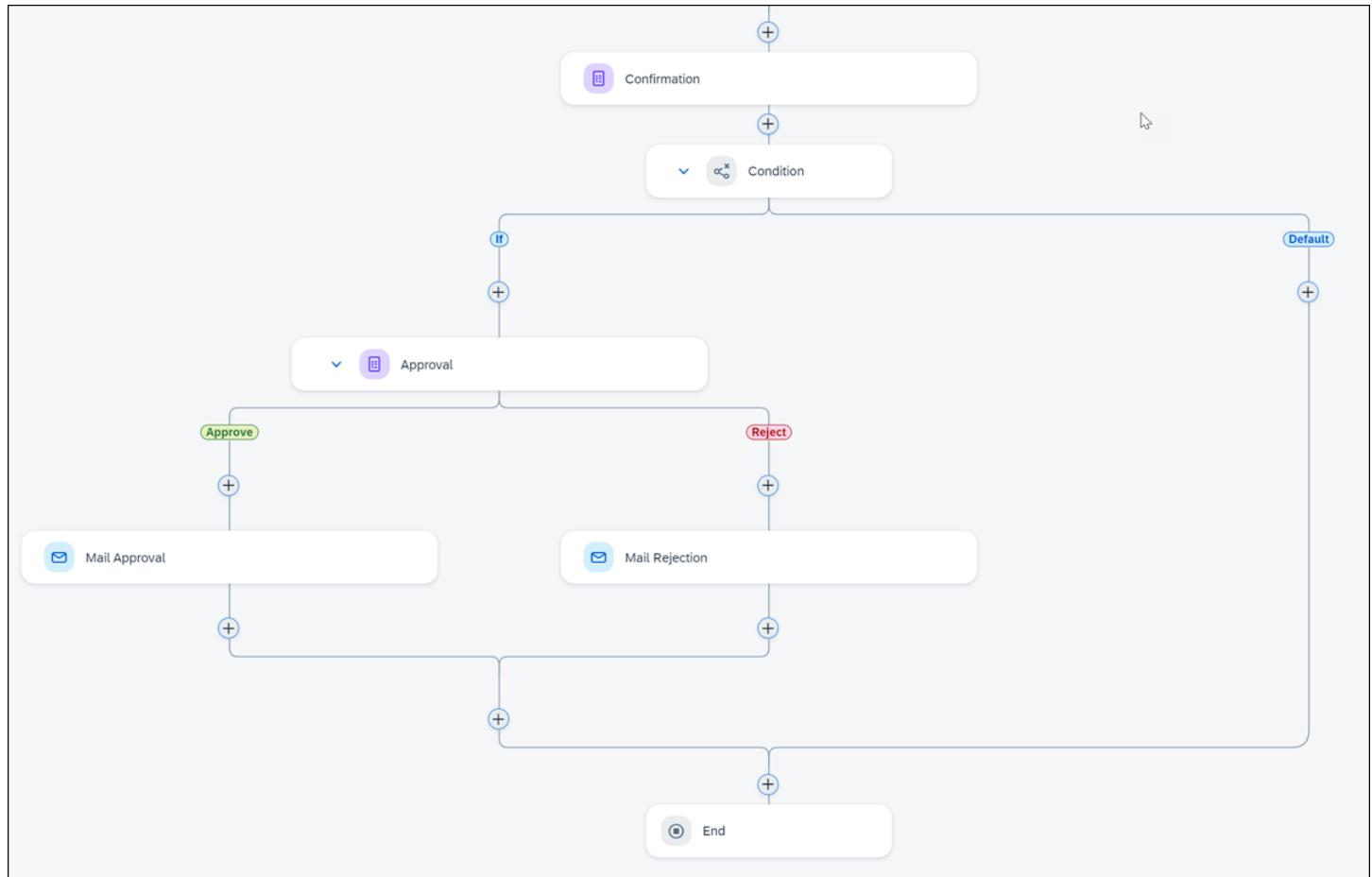
- [Add and Configure Process Conditions](#)
- [Add a Branch to a Process](#)

Add and Configure Process Conditions

You can add and configure conditions to a business process, enabling you to route a running process based on applying IF or DEFAULT rules to the process context.

Context

In the following example, this process condition checks if the value of a submitted request is over 1000. If yes, then the request is sent for approval. If no, then the request is considered as a default request, automatically approved, and a confirmation mail is sent.



For this example, the condition is configured like this:

Edit Branch Condition

Satisfies:

All Any of the following:

is greater than

Add Clear All Add Group

Summary:

Apply Cancel

You can also create more detailed conditions by using the **All** or **Any** feature.

In the following example, the process must include **Any** of the following three values for the process to be sent for approval:

- Value of the request is greater than 1000.
- Office location is Potsdam.
- The item requested isn't a laptop.

Edit Branch Condition

Satisfies:

All **Any** of the following:

» Order Amount ×	is greater than	1000	X
» Office Location ×	is equal to	Potsdam	X
» Item Required ×	is not equal to	Laptop	X

Add Add Group

Summary:

if value of » Order Amount is greater than 1000
or value of » Office Location is equal to Potsdam
or value of » Item Required is not equal to Laptop

Apply Cancel

You can also apply both the **All** and **Any** rules to groups of rules, such as in the following example:

In this example, the process must include **any** of the first group conditions AND **all** of the second group of conditions for the request to be approved.

Edit Branch Condition

Satisfies:

All **Any** of the following:

» Order Amount ×	is greater than	1000	X
» Office Location ×	is equal to	Potsdam	X
» Item Required ×	is not equal to	Laptop	X

All Any Remove Group

» Name ×	is equal to	Maya	X
» Delivery Date ×	is earlier than	» Delivery Date ×	X

Add

Summary:

if value of » Order Amount is greater than 1000
or value of » Office Location is equal to Potsdam
or value of » Item Required is not equal to Laptop
or value of » Name is equal to Maya
and value of » Delivery Date is less than » Delivery Date

Apply Cancel

Procedure

This is custom documentation. For more information, please visit the [SAP Help Portal](#)

- From the process editor canvas, choose **►+ (Plus) ► Flow Login ► Condition**.

The condition is added to the process and the side panel is displayed.

- Optional:** Enter a **Condition Name** and a **Branch Name** to customize how the condition is displayed in the process.
- Choose **Open Condition Editor**.
- Configure a minimum of one condition to be satisfied.

For example, the office location must be Potsdam:

Edit Branch Condition

Satisfies:

All Any of the following:

Office Location is equal to Potsdam

Clear All X

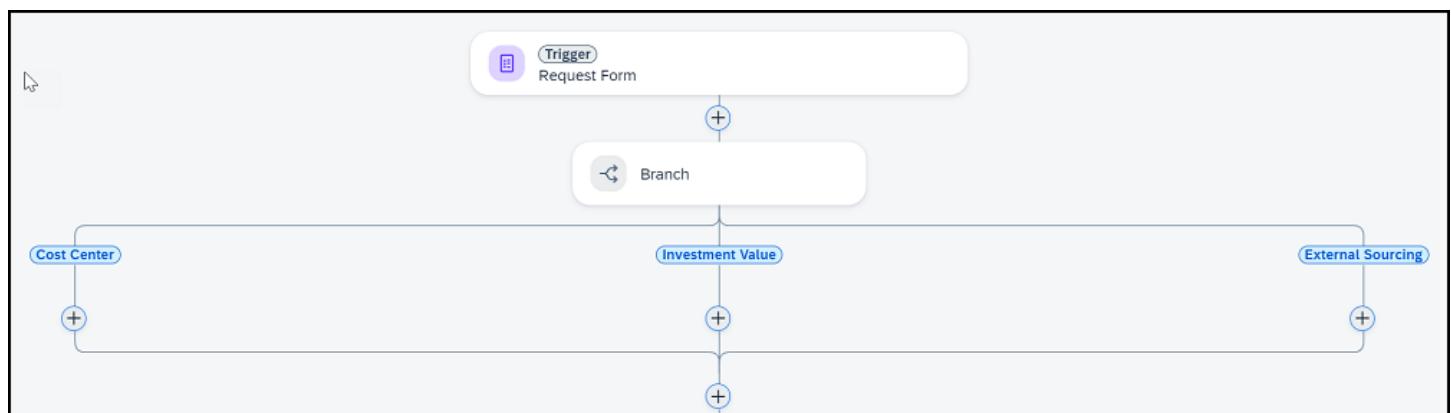
- Choose **Apply**.
- Save your changes.

Add a Branch to a Process

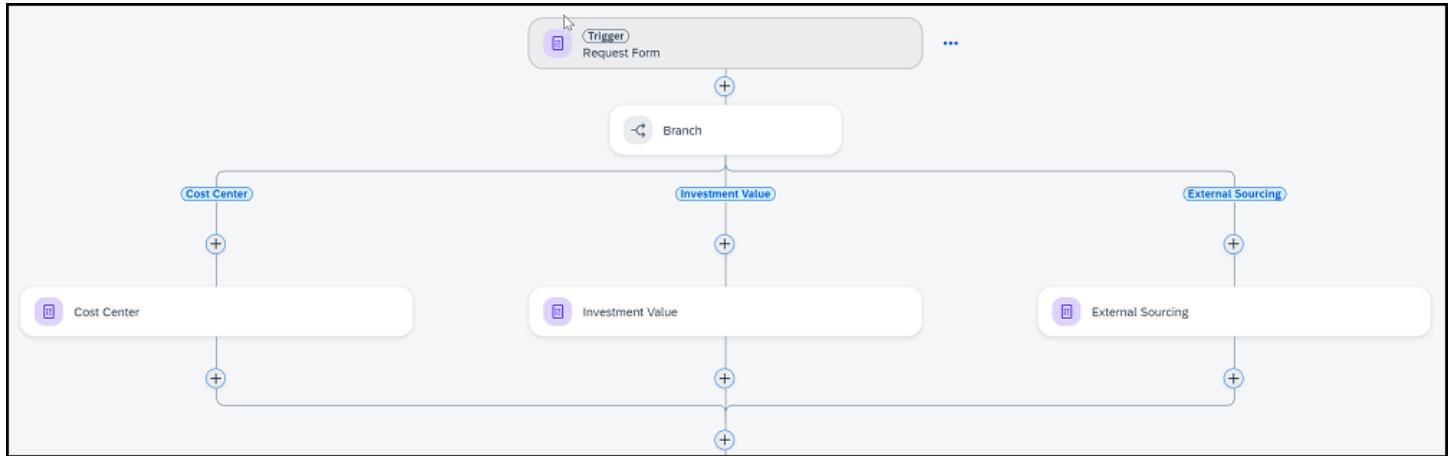
You can add a maximum of 10 parallel branches to your business process, enabling you to run multiple process tasks at the same time. These tasks run concurrently, with the process progressing once all tasks are complete.

Context

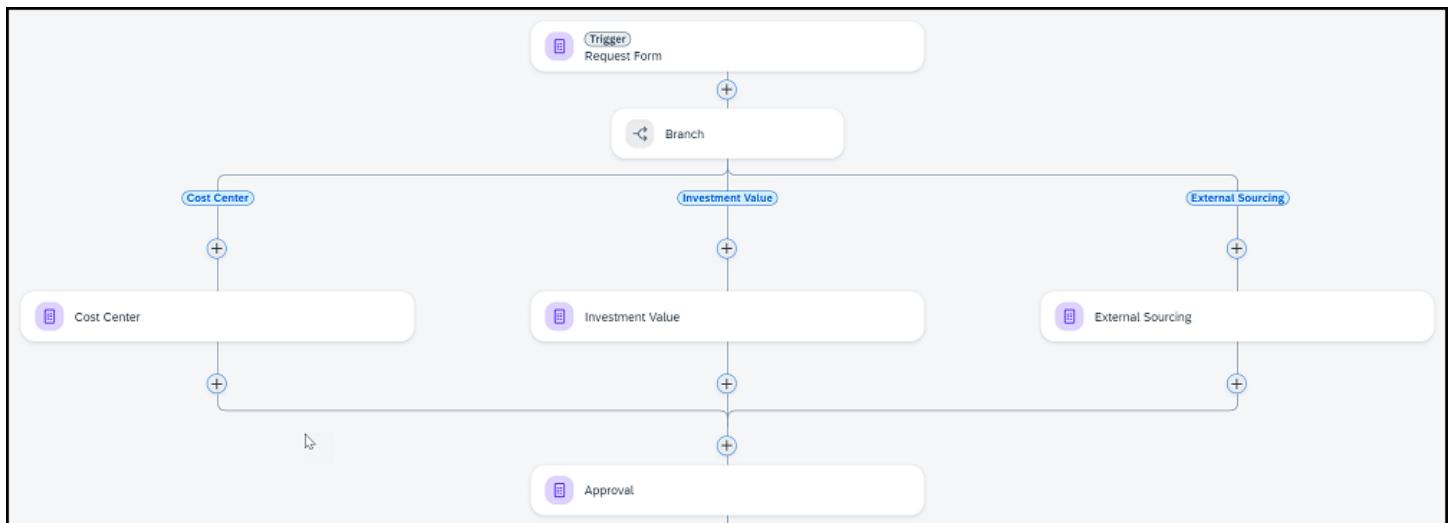
In the following example, three branches have been added to an investment request process and labeled accordingly:



These branches are then enhanced by adding process steps, in this case three additional forms:



An approval form is then added to the process. This approval is only triggered once all three branches have successfully processed.



You can monitor the status of branch tasks in a running process using the [Execution Log](#) in the [Monitoring](#) area.

In this example the process has reached the branch:

EXECUTION LOG

Do

Task "4711 IT costs" available

Aug 5, 2022, 2:08:32 PM

Instance ID: [5339c8fe-14b7-11ed-8f7c-eeee0a9bd7bc](#)

Recipients:

Initiator:

Do

Task "Yes" available

Aug 5, 2022, 2:08:31 PM

Instance ID: [5291a219-14b7-11ed-8f7c-eeee0a9bd7bc](#)

Recipients:

Initiator:

Do

Task "500.000 EUR" available

Aug 5, 2022, 2:08:31 PM

Instance ID: [527fc7c3-14b7-11ed-8f7c-eeee0a9bd7bc](#)

Recipients:

Initiator:

Do

"Branch" reached

Aug 5, 2022, 2:08:31 PM

Do

completed the task "5e6d3522-14b3-11ed-8f7c-eeee0a9bd7bc"

Aug 5, 2022, 2:08:31 PM

Do

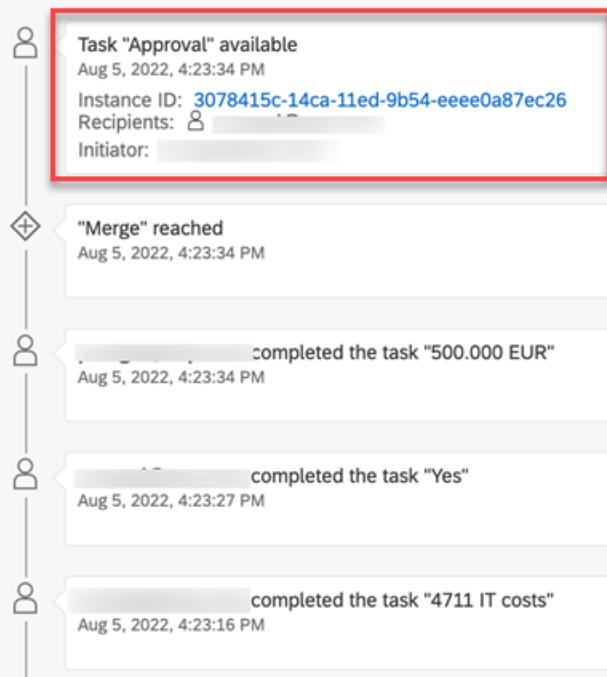
Task "5e6d3522-14b3-11ed-8f7c-eeee0a9bd7bc" available

Aug 5, 2022, 1:40:13 PM

Instance ID: [5ea3fc9e-14b3-11ed-8f7c-eeee0a9bd7bc](#)

And in this example all branch tasks have successfully processed and the approval has triggered:

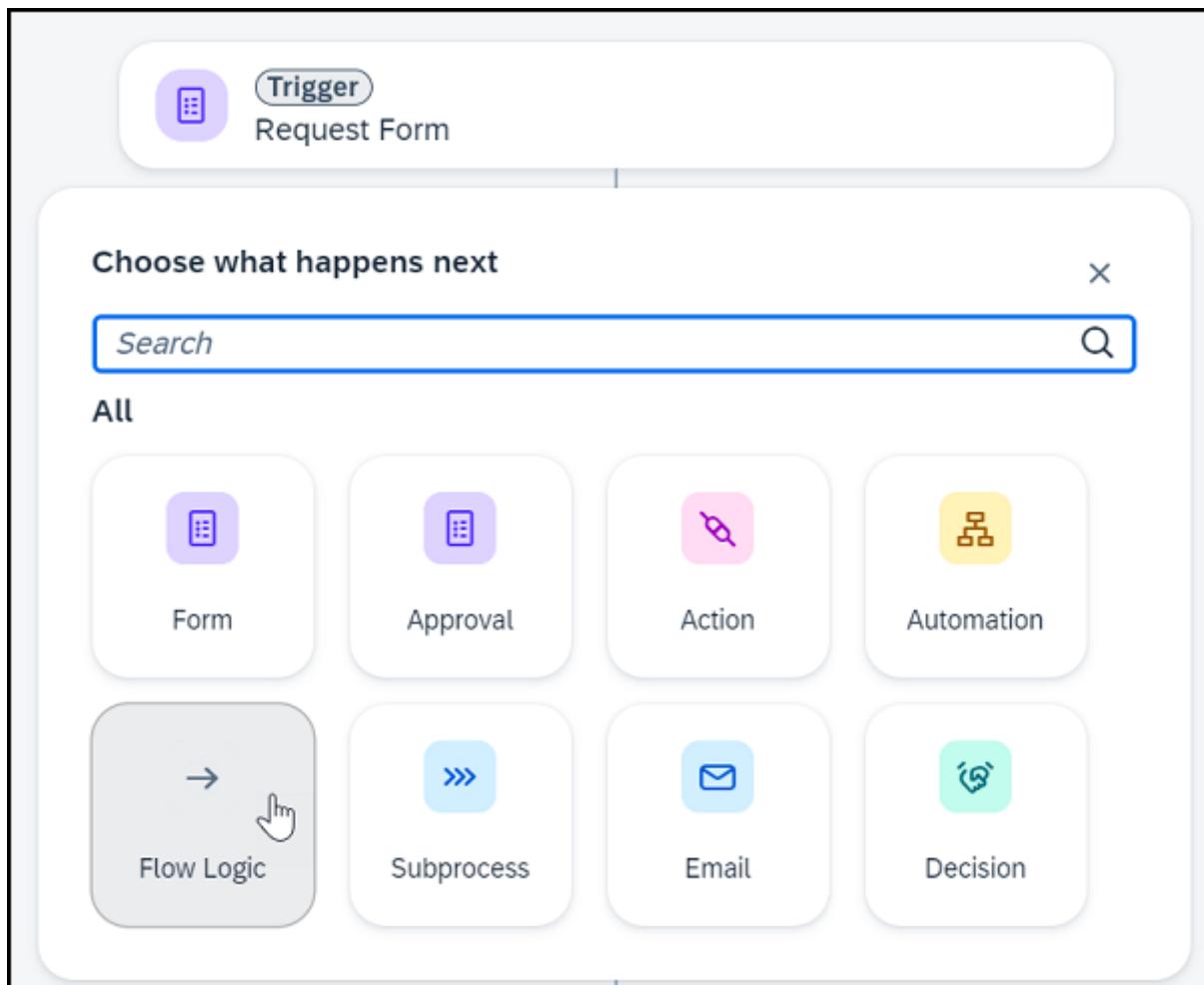
EXECUTION LOG



Procedure

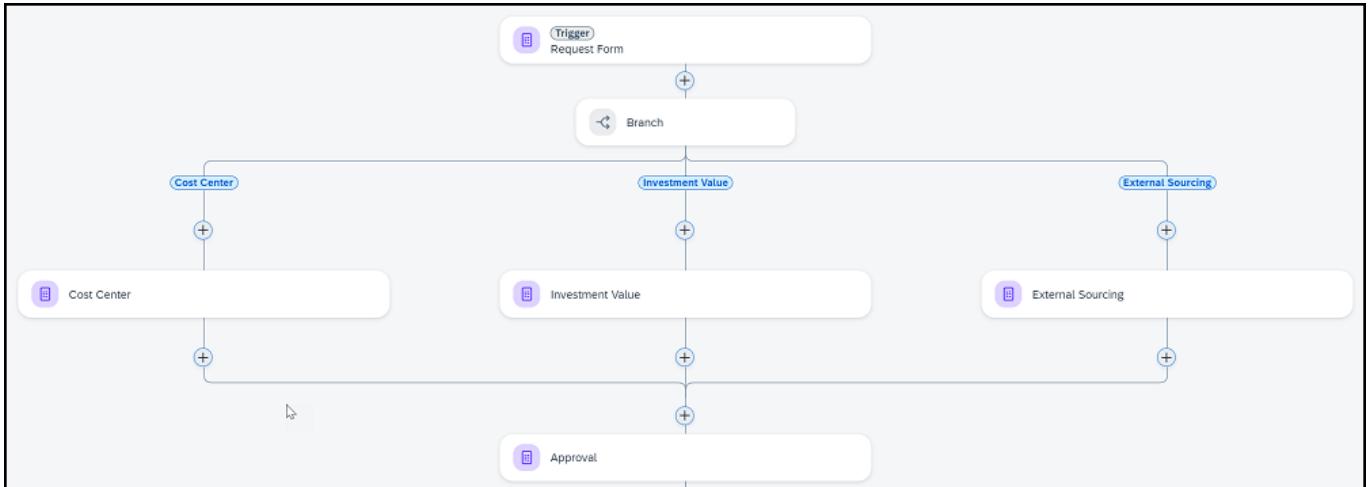
- From the process editor, click **+** and select **Controls - Branch**.

From the process editor, choose **Flow Logic**.

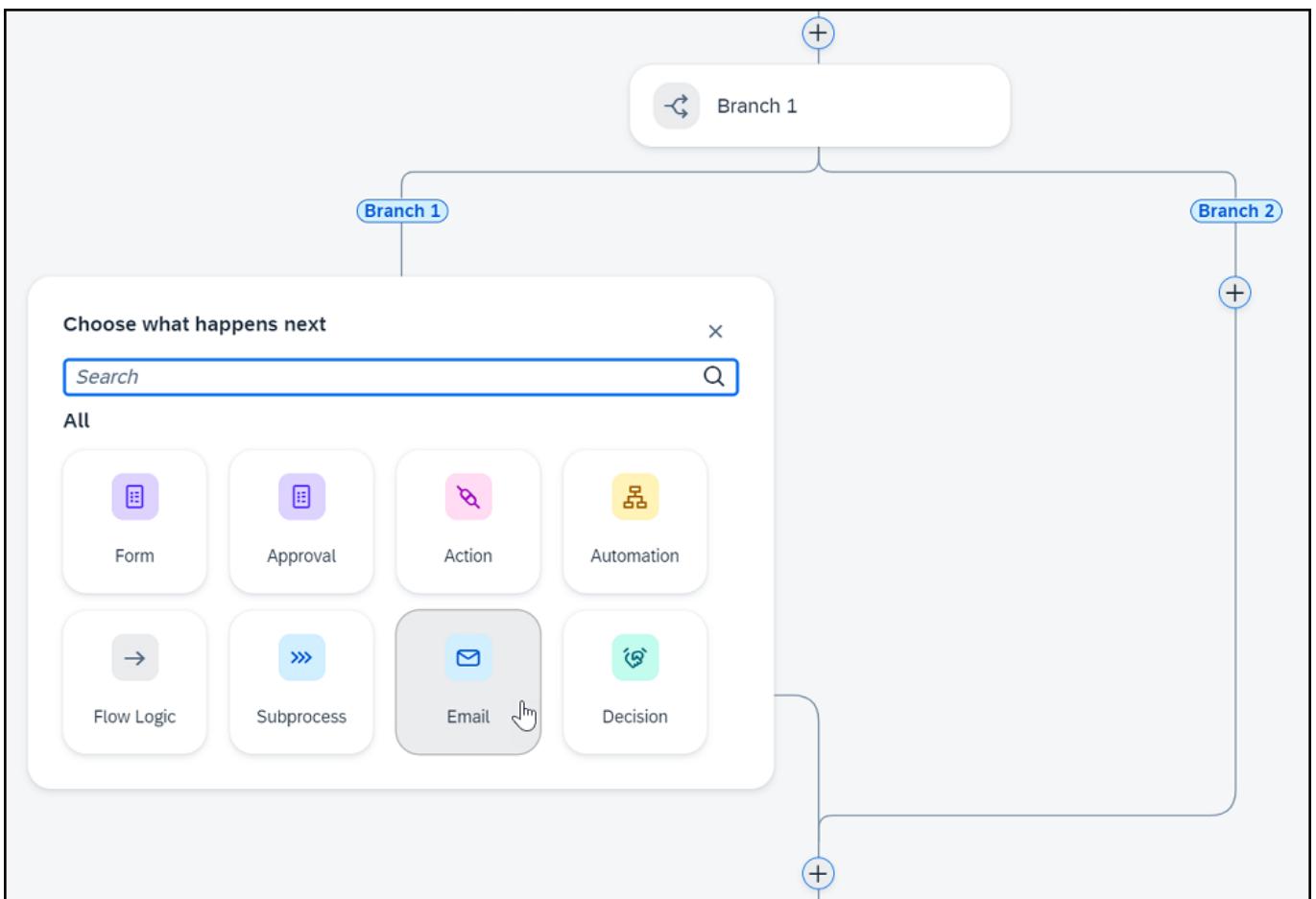


Two branches are added to the process and the branch settings are displayed.

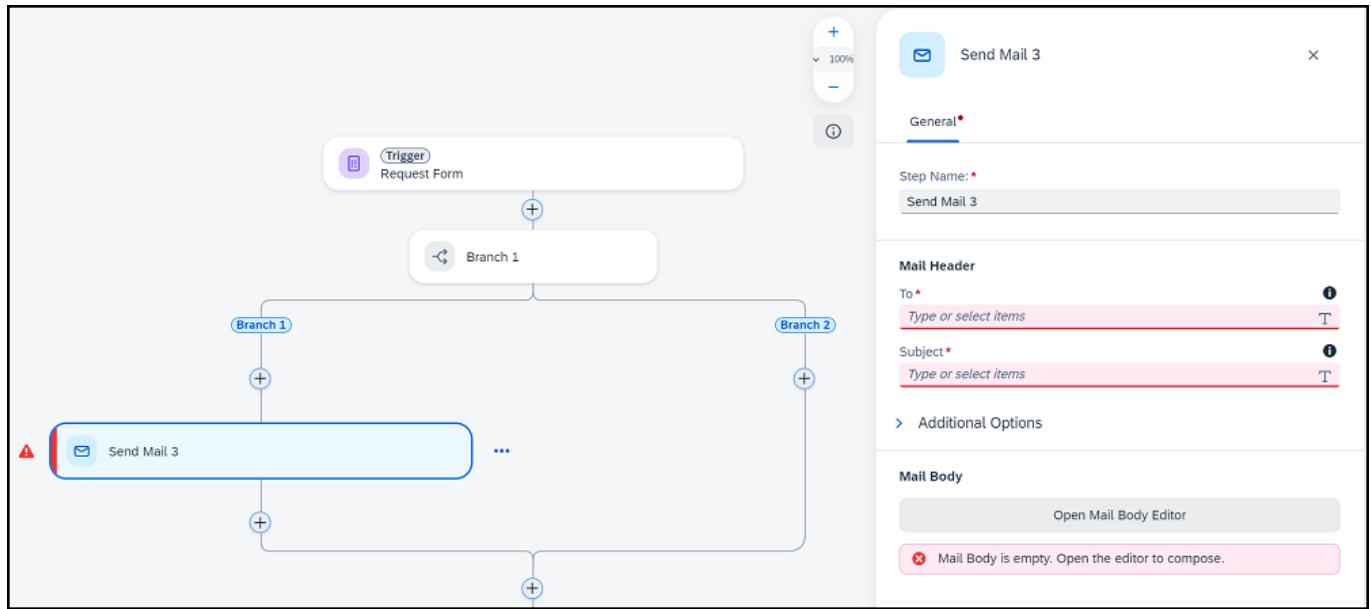
2. Configure the branch accordingly, adding an optional **Branch Name** and additional branches using **Add branch**.



3. Add process steps to each branch using **+** and selecting the required step. In this example, a mail notification is added to the branch.



4. Configure the process step, ensuring that all errors are corrected. In this example, the mail notification is missing the **Mail Header** information.



- Once all branches are configured, save your changes.

The branches have been added to the process and will be processed concurrently when the process is running.

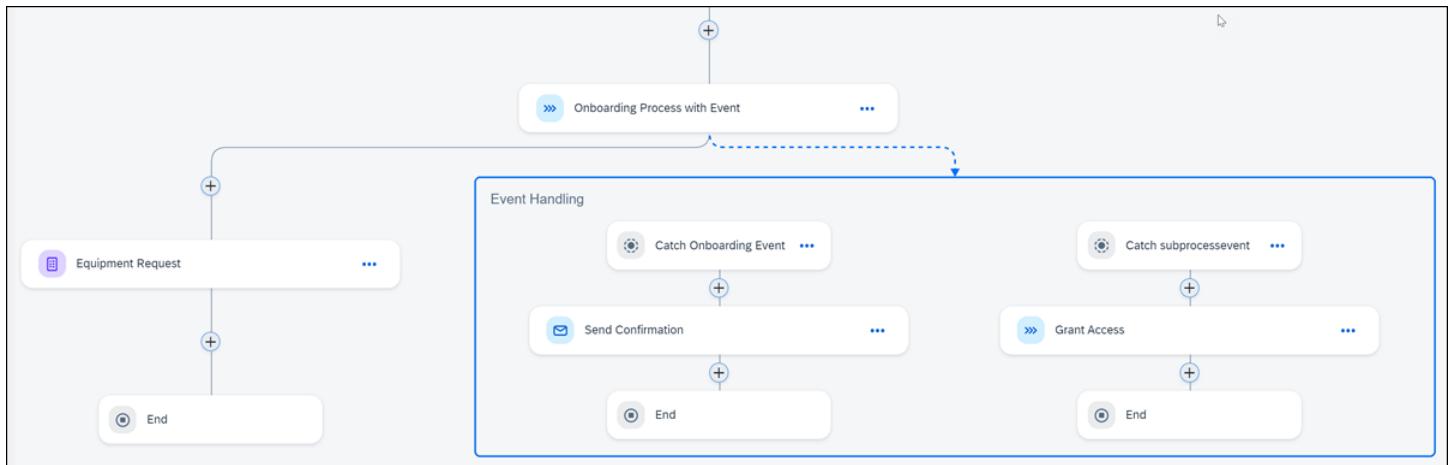
Raise an Escalation Event in Process Editor

You can raise and configure escalation events from anywhere in a business process.

Context

For an escalation event, you only define its name without any action attached. If the process reaches that event, it throws an escalation to its parent process. If you configure event handling for this event in the parent process, then the escalation is caught and you can send, for example, a notification.

In the following sample, we have a parent process with a subprocess that catches two raised events.



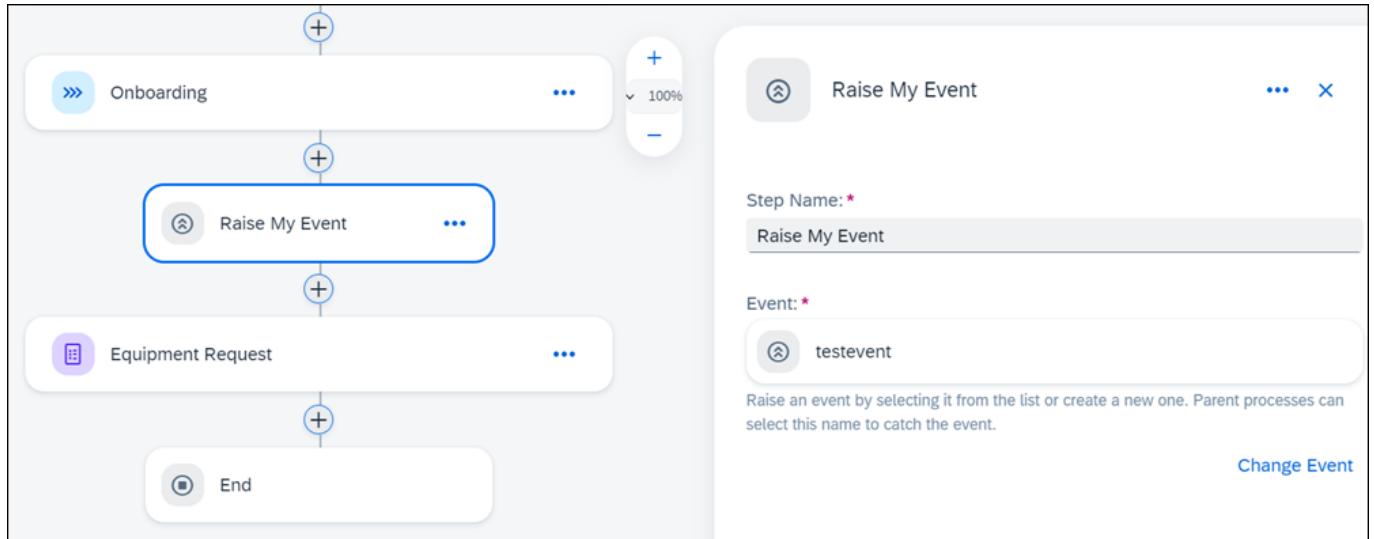
Procedure

- From the process editor canvas, choose **+ (Plus) > Flow Controls and Events > Raise Event**.
- On the **Raise Event** tile, choose **Create Event**.
- Enter a name for the event and choose **Create**.

The event is added to the process.

4. **Optional:** On the side panel, you can change the event by choosing **Change Event**.

You can then select an event from the list that opens or create another event from there.



5. Save your changes.

Handle Events Using the Parent Process

You can catch events from a business process or workflow.

Context

The parent process can catch an event. Not catching an event is not considered an error.

The following types of events can be raised by a subprocess or a workflow:

- Escalation event: This event type can be caught by its parent process.
- Timer events: This event type can be caught by its parent process or a user task, for example, an approval.

Handle Escalation Events

Procedure

1. From the step menu of the process or workflow, choose **Options**, and then choose **Handle Event**.

If your process does not contain a raised event yet, choose **Open Subprocess** to add an event to it.

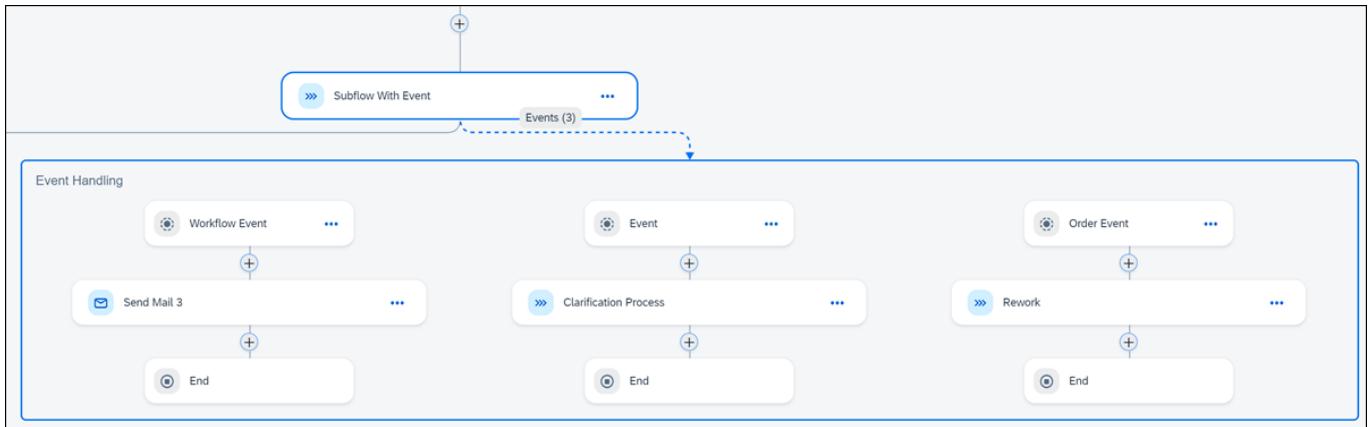
2. On the **Event Handling** tile, choose **Catch Event**.

3. In the catch flow that is created, choose **+** (plus) to add the step to catch the event.

You can catch multiple events, and they are then displayed next to each other. The number of handled events is displayed if you click the process or workflow. If you then click the number, the **Event Handling** tile opens.

4. Configure the event on the side panel.

5. Save your changes.



Handle Timer Events

Procedure

1. From the step menu of the process or workflow, choose **Options**, and then choose **Handle Event**.

If your process does not yet contain a raised , choose **Open Subprocess** to add an event to it.

2. On the **Event Handling** tile, choose **Timer Event**.

3. In the catch flow that is created, choose **+ (plus)** to add the step to catch the event.

You can catch multiple events and they are then displayed next to each other. The number of handled events is displayed if you click the process or workflow. If you then click the number, the **Event Handling** tile opens.

4. Configure the event on the side panel.

Select whether the timer is based on the task creation date or a reference date:

- **Task Creation:** Set the toggle to **On Task Creation** to start the event handling directly or to **After Task Creation** to start the event handling only after the completion of the wait duration that you define.
- **Reference Date:** Select a date from the process context list, for example, the sales order date.

5. Save your changes.

Handle a Timer Event with a User Task

You can catch timer events from a user task, for example, an approval or a form.

Procedure

1. From the step menu of the user task, choose **Options**, and then choose **Handle Event**.

2. On the **Event Handling** tile, choose **Timer Event**.

3. In the catch flow that is created, choose **+ (plus)** to add the step to catch the event.

You can catch multiple events that are then displayed next to each other. The number of handled events is displayed if you click the process or workflow. If you then click the number, the **Event Handling** tile opens.

4. Configure the event on the side panel.

Select whether the timer is based on the task creation date, the due date, or a reference date:

- **Task Creation:** Set the toggle to **On Task Creation** to start the event handling directly or to **After Task Creation** to start the event handling only after the completion of the wait duration that you define.

- **Due Date:** Start the event handling when the due date of the user task arrives.

If you select this option for a user task that has no due date defined, you get a notification telling you to update the user task first before you can use this option.

- **Reference Date:** Select a date from the process context list, for example, the sales order date. The event handling is started on that date.

5. Save your changes.

Add Actions to a Process

You can embed external skills and capabilities into your SAP Build Process Automation projects using actions. Action projects can either be created in the lobby, allowing you to upload an Open API specification file, or imported from the store to your library.

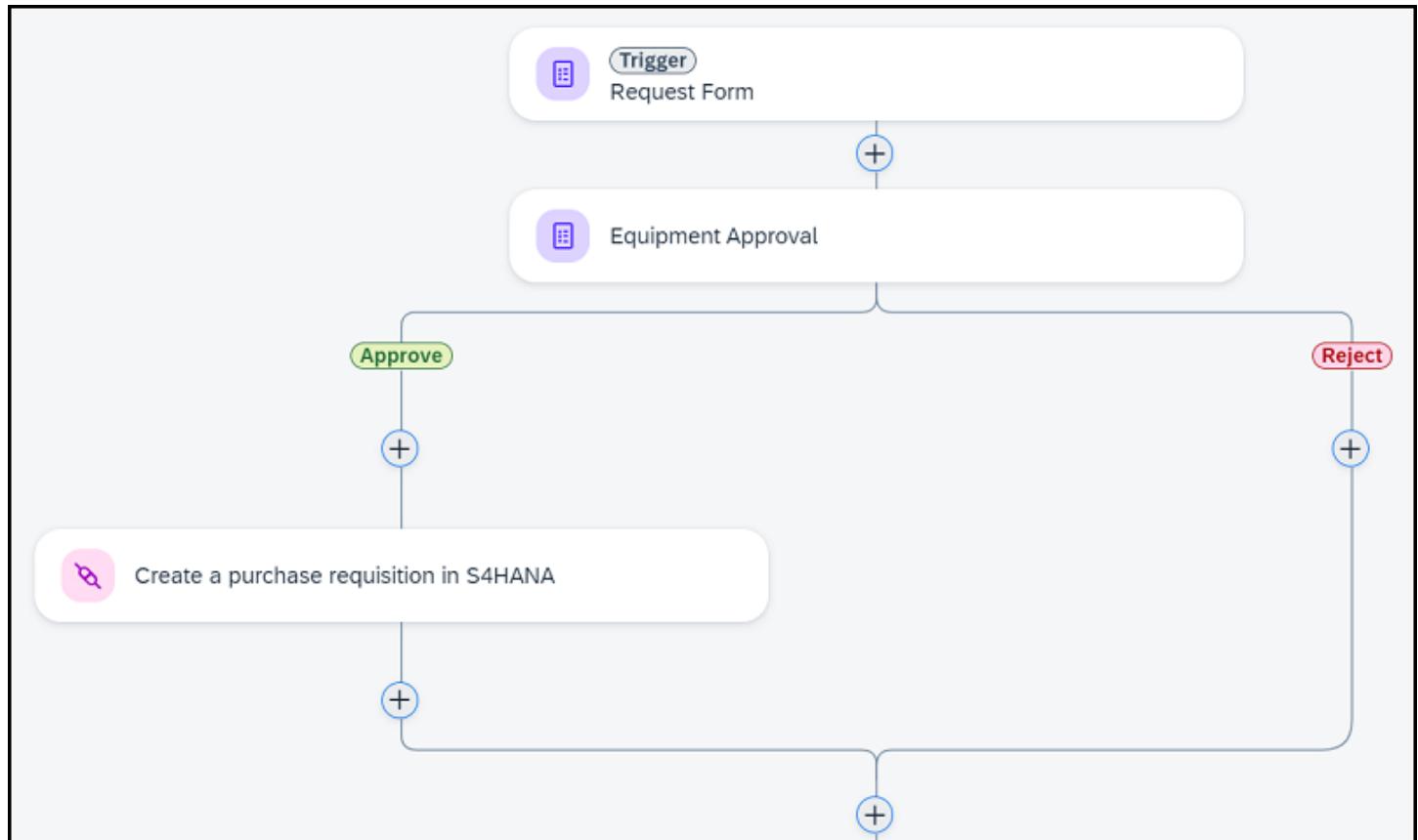
Prerequisites

Before adding an action to your business process, the corresponding action project must be available in your lobby. There are two ways to add an action project to your lobby:

- [Create an Action Project](#)
- [Use the Store](#)

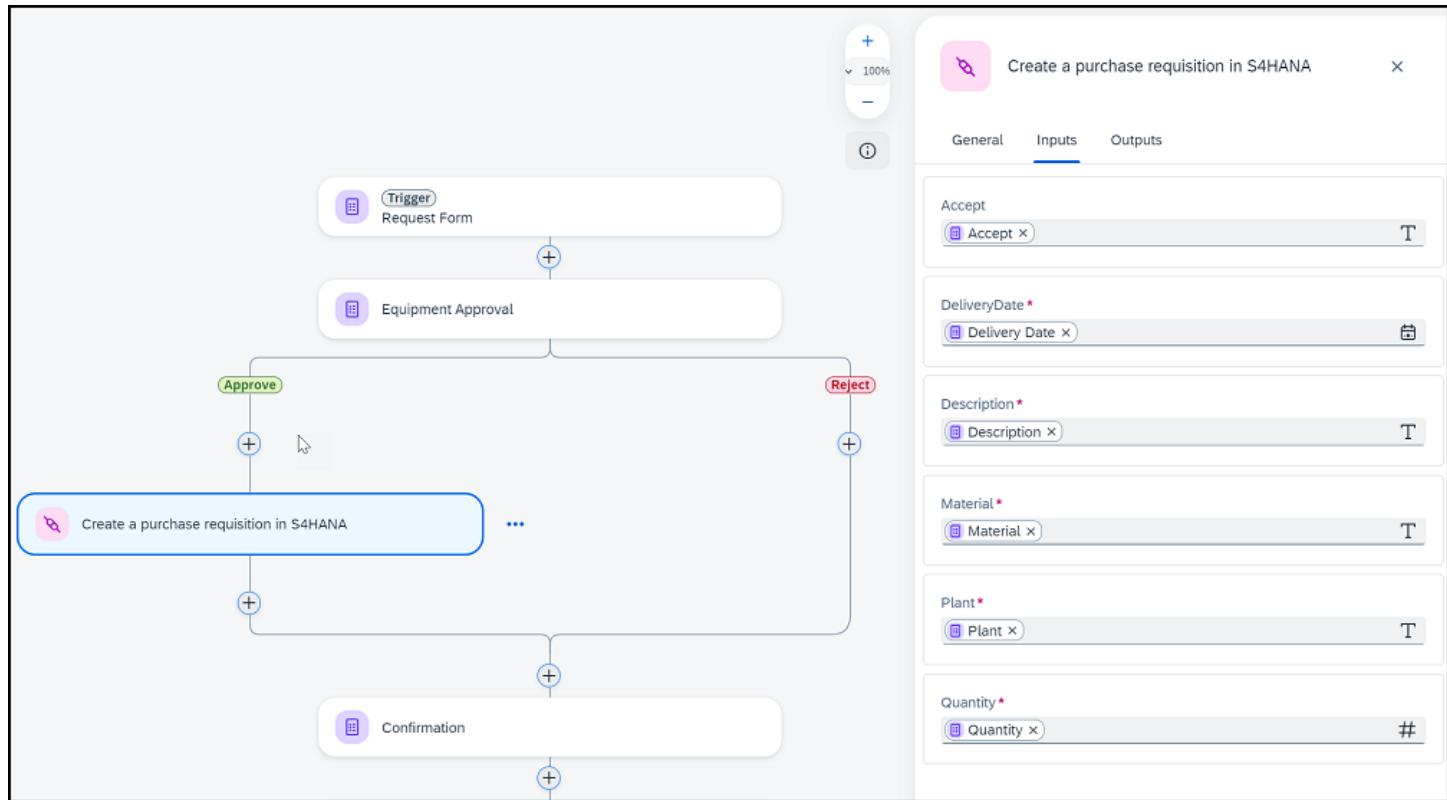
Context

In the following simple request process, a request submission form starts the process running. If the request is approved, an action is used to create a purchase requisition in an external system:



The graphic is explained in the accompanying text.

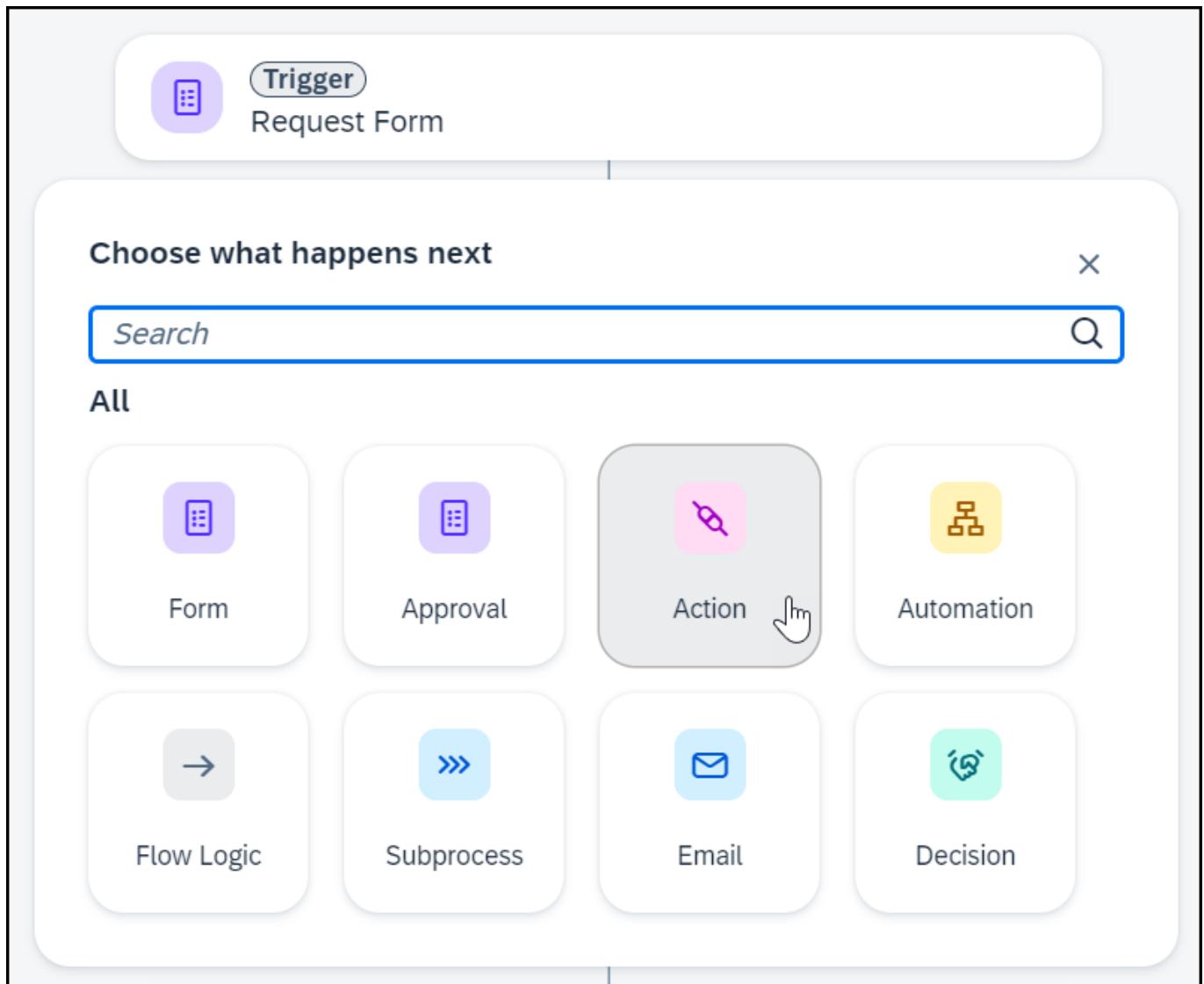
And the inputs of the purchase requisition are mapped from the fields used in the form:



Procedure

- From the process editor canvas, choose **+** and select **Actions - Browse Library**.

From the process editor, choose **►+ > Flow Logic > Parallel Branch**.



2. Search for the action you want to add to your process, and choose **Add**.

Browse library

Search Sort by Artifact Name: Ascending

Action Type	Projects	Line Of Business	Products
Action Add a new pet to the store	Action Add a new pet to the store Add a new pet to the store	Action Add new entity to AllowedStatusesSet	
Project: Julians Petstore Add	Project: PetStoreActionProject Add	Project: MM - POC - ZOMR - Status... Add	
Action Add new entity to AllowedStatusesSet	Action Add new entity to AllowedStatusesSet	Action Add new entity to BusinessUserBusinessRoleAssign...	
Project: MM - POC - ZOMR - Status... Add	Project: MM - POC - ZOMR - Status... Add	Project: faizk Add	
Action Add new entity to BusinessUserSubscriptionAssignm...	Action Add new entity to CustomerOrderItemCollection	Action Add new entity to JobRequisition	

Close

The action is added to your process editor canvas.

3. Click the **Action** on the canvas, opening the configuration side panel.

The screenshot shows a process editor canvas with a workflow step labeled "Create a purchase requisition in S4HANA". This step is highlighted with a red warning icon. To the right, a configuration side panel is open for this step. The panel has three tabs: "General", "Inputs", and "Outputs". The "General" tab shows the step name "Create a purchase requisition in S4HANA". The "Inputs" tab contains a required field "Destination variable:" which is highlighted in red with the error message "Select a Destination Variable". The "Outputs" tab is empty.

4. Configure your action as needed. Outstanding configuration items are highlighted in red and displayed in the design console.

In this example, the action requires a destination to be configured and for inputs to be mapped.

i Note

When configuring your action, you may need to add and configure other process or automation artifacts before your project can be released and deployed.

5. Save your changes.

Your action has been added to your process. It can now be consumed when the process is running.

Run Step on Behalf Of

With this feature, you can allow business users who participate in the business process to perform an action on external systems or to execute a subprocess or workflow in the same system.

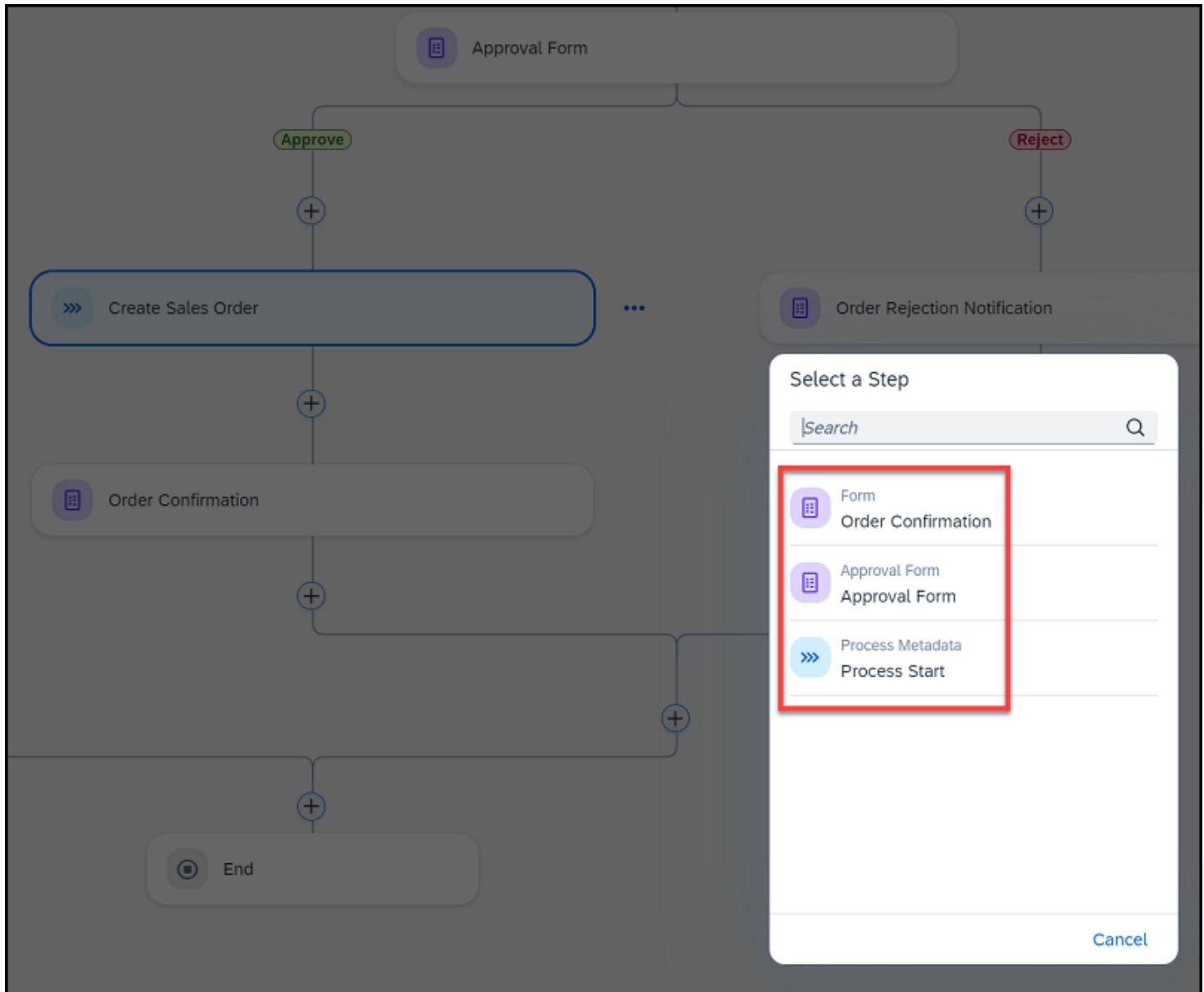
This feature also provides clear information on who triggered the step. This information can help your IT department and is valuable during the audit process.

Prerequisites

- To enable principal propagation, the identity of cloud users gets forwarded to a remote system or service in the Cloud Foundry environment. Principal propagation is possible between two cloud systems or between a cloud and an on-premise system. Set up the connections to the required system using the SAP Destination service. See [Configure SAP Build Process Automation Destinations](#).
- It's important that the step that you are mapping includes the details of a human user who has access to the required system. If the user doesn't have access or if the user is a technical user, the process might fail at runtime.
- The steps eligible to be run on behalf of another user are actions or subprocesses.

Context

You can select the process participant, on whose behalf the step runs, from the process steps. That is, you either map the user who started the process or a user who executed any previous step of the process. That is, you select a step and doing so, indirectly reuse its user in the subprocess. Then, the action or subprocess is executed on behalf of that process participant.



In a procurement process, for example, an employee requests a laptop and gets the manager's approval. As a result, a purchase requisition is created on behalf of the manager who approved the request. Therefore, the purchase requisition is run on behalf of the manager.

Sometimes determining the user of this previous step is tricky:

- We don't recommend using the user of a step inside a condition branch. This is because the outcome of the condition might come from another branch that doesn't include the step you are referring to, in which case, the user that you set isn't available and the process fails.
- By default, selecting a user in **Run step on behalf of** is optional. However, in a sequence of steps like a purchase requisition, it becomes mandatory because the subprocess has an action with **Run step on behalf of** configured to process start.

Example

You create a process A that includes an action for which you set **Run step on behalf of** to the user who started the process.

You create a second process B in which you include process A as a subprocess. Now, this subprocess requires an entry in **Run step on behalf of**.

- It's also possible, that you set up your process well, but then you delete the step that contained the user that you referred to in the action or subprocess. An error message prompts you to enter a user for the action or subprocess.

again.

Related Information

[User Propagation from the Cloud Foundry Environment to SAP S/4HANA Cloud](#)

[User Propagation from the Cloud Foundry Environment to SAP SuccessFactors](#)

[User Propagation between Cloud Foundry Applications](#)

Add a Subprocess to a Process

You can add subprocesses to your main process in the process editor, allowing you to modularize your business process and reuse subprocesses..

Prerequisites

The subprocess must be in the same or a dependent project of your main process in the process editor.

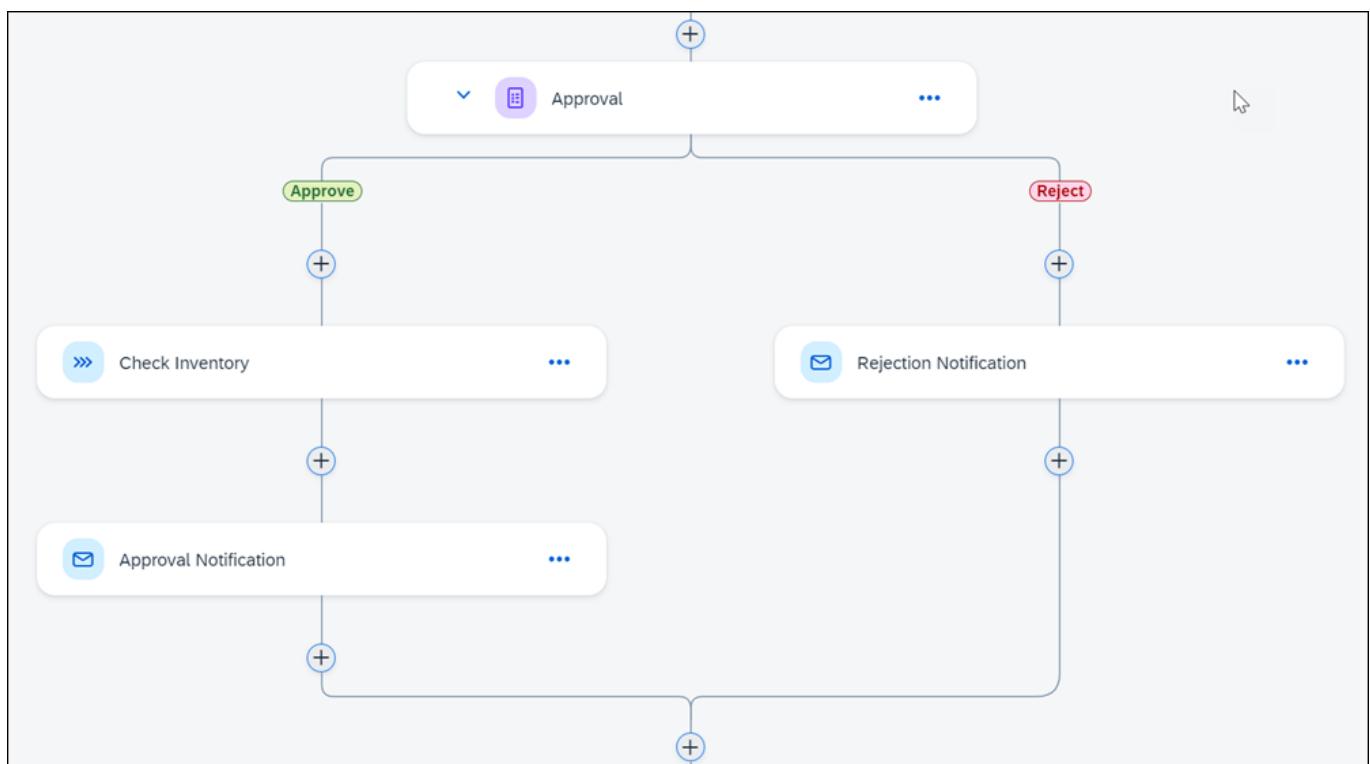
Context

Subprocesses can help you organize your processes. Technically they are simply processes but you can use them to create reuse sequences. Then, you can add them to multiple processes and increase your efficiency and consistency. There are the following types of subprocesses:

- A process that is included, and later run, as part of a main business process.

Eligible subprocesses need to be available in the same business process project. All processes contained within the same project can be a main process or used as a subprocess, depending on how you use them.

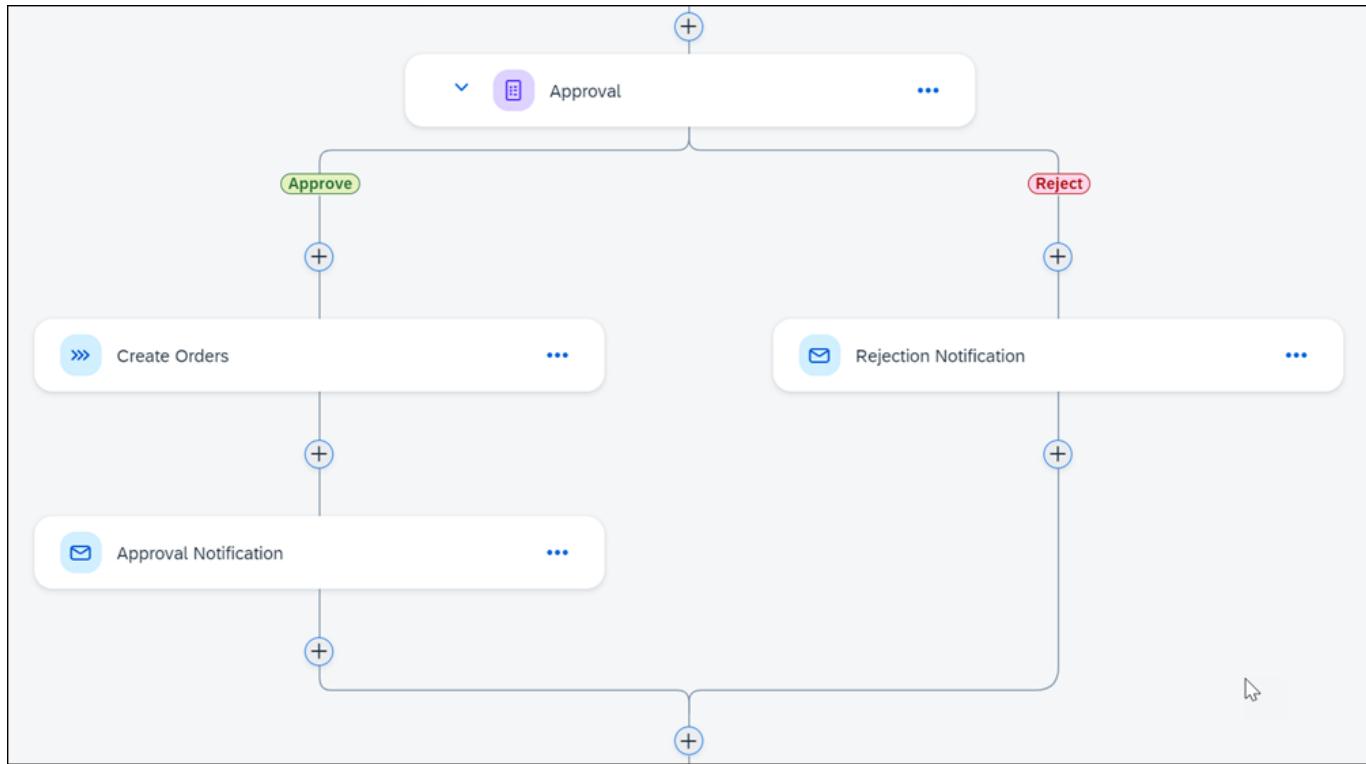
As an example, a subprocess that checks for inventory levels is included as part of the following purchase order request process:



The subprocess must run successfully before the approval mail notification is sent to the requestor.

- A workflow that was developed in SAP Business Application Studio or was added as part of live process projects from the store.

As an example of a workflow started from a process, this request process includes a workflow to create orders for approved requests:



In this case, the workflow runs only for approved requests. This workflow must run successfully before the overall process progresses, which means the approval mail is only sent once the order has been created.

Procedure

- From the process editor canvas, choose **+ (plus) > Subprocess**
 - Select one of the following options:
 - Use a listed existing process. If a listed process is grayed out, it doesn't fulfill all criteria.
 - Create a new subprocess by choosing **Blank Subprocess**. This is simply another process that you create from scratch.
 - Use an existing workflow by choosing **Call a Workflow**.
 - Configure any necessary process or workflow **Inputs** and **Outputs**.
 - Save your changes.
- The process or workflow is added as a subprocess and must run successfully before the main process continues.
- If you inserted a blank process, a new workflow, or a workflow from the library, choose the respective link in the image below and design that subprocess.

Related Information

[Create a Business Process](#)

[Call a Workflow](#)

[Modeling a Workflow](#)

[Create a Workflow Module](#)

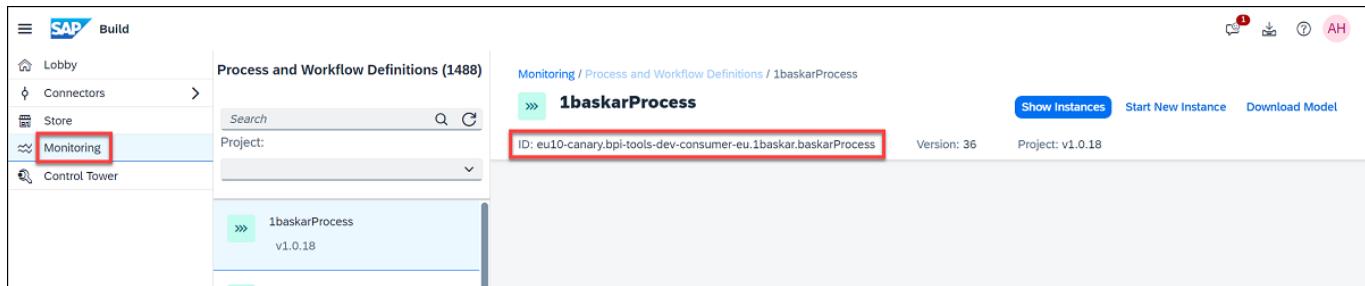
[Introducing Subprocesses as Referenced Subflows in SAP Build Process Automation](#)

Determine the Workflow Definition ID

Find the workflow definition ID of live process projects.

Procedure

1. In SAP Build, choose **Monitoring**.
2. Under **Manage**, choose **Processes and Workflows**.
3. Select the relevant workflow.



Call a Workflow

You want to add a prebuilt workflow as a subprocess.

Prerequisites

From the process editor canvas, you choose **Call a Workflow** to insert an existing workflow as a subprocess.

Procedure

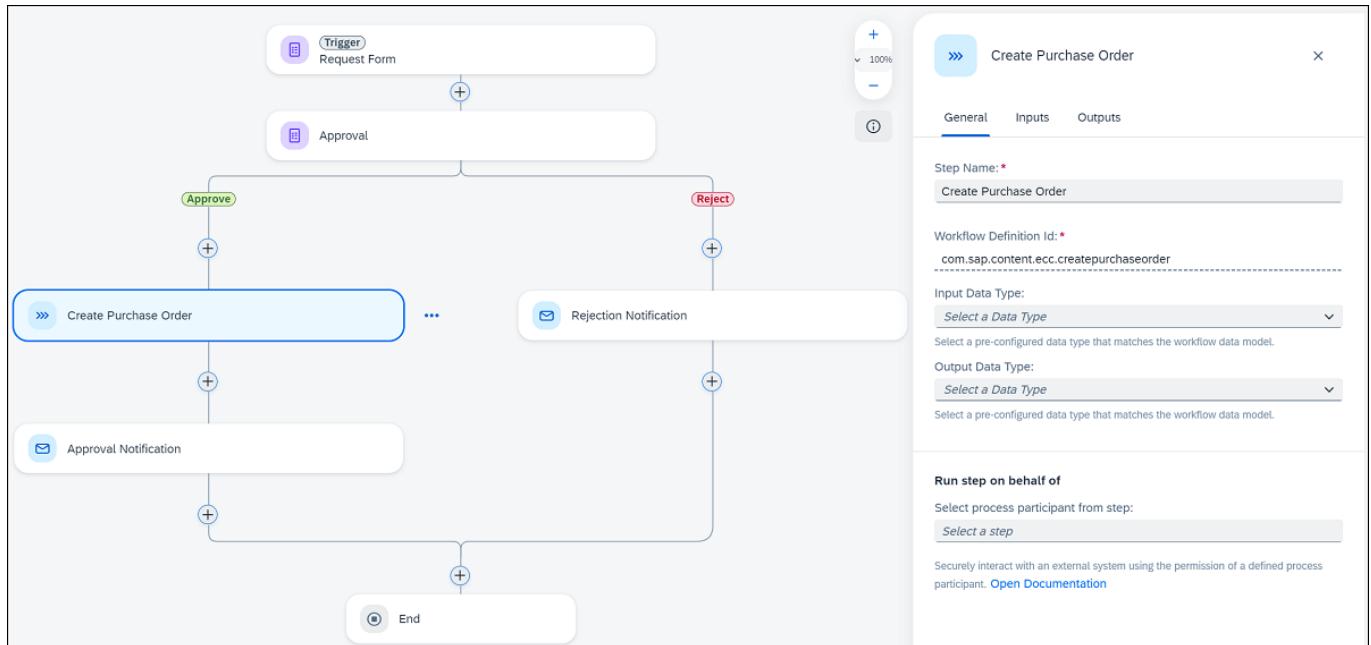
1. On the **Call Workflow** screen, select a workflow and choose **Add**.

On the **General** tab, the workflow definition ID is filled in automatically and is read only. The workflow name is also prefilled but can be changed.

2. Select an existing **Input Data Type** or click **+ New Data Type** to create one.

If creating a new data type, you must then configure this via your **Project Overview** area. See [Create a Data Type](#).

3. Select an existing **Output Data Type** or click **+ New Data Type** to create one.



4. Save your changes.

Add Mail Notifications to a Process

You can add mail notifications to your business process using the process editor, allowing you to send preconfigured emails to recipients while a process is running.

Prerequisites

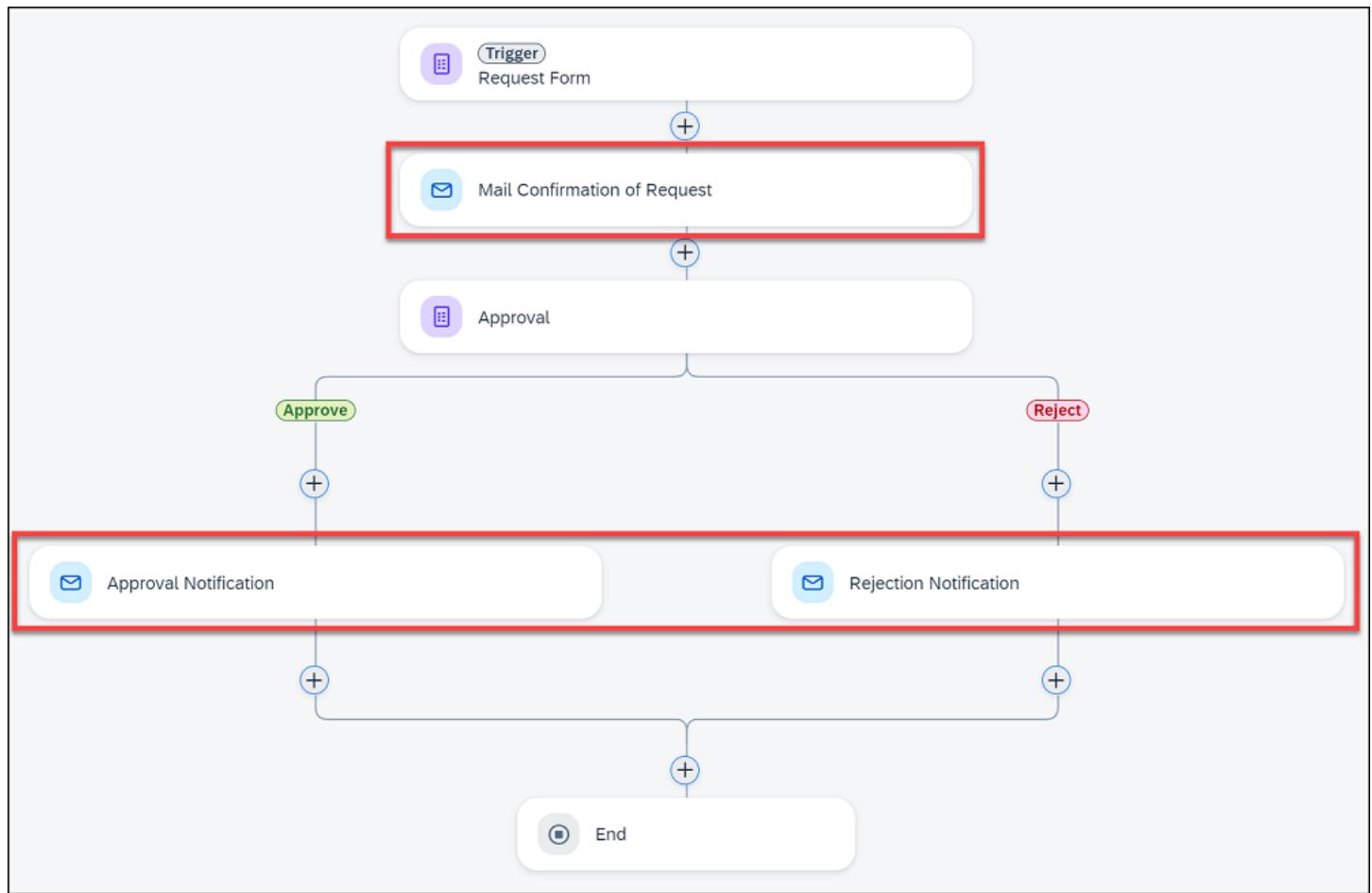
Before adding mail notifications to a process, you must first configure your SMTP mail destinations for SAP Build Process Automation. See [Configure an SMTP Mail Destination](#).

This configuration includes assigning the 'From' address for sent mail, for example: user@example.com. This inbox must be able to receive replies.

Context

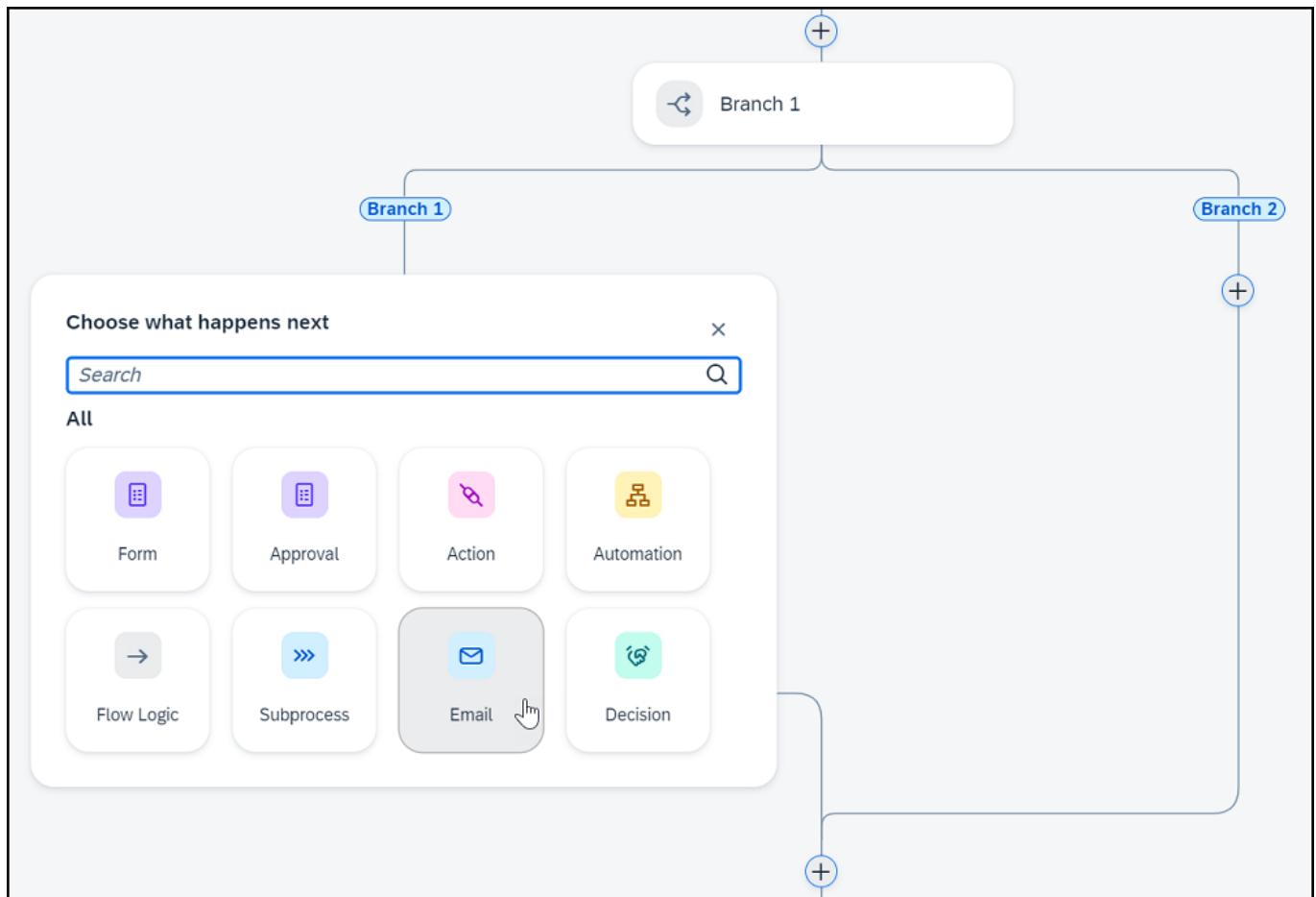
As an example, there are three mail notifications in this simple request process:

- Confirmation that the request was received
- Confirmation that the request was approved
- Confirmation that the request was rejected

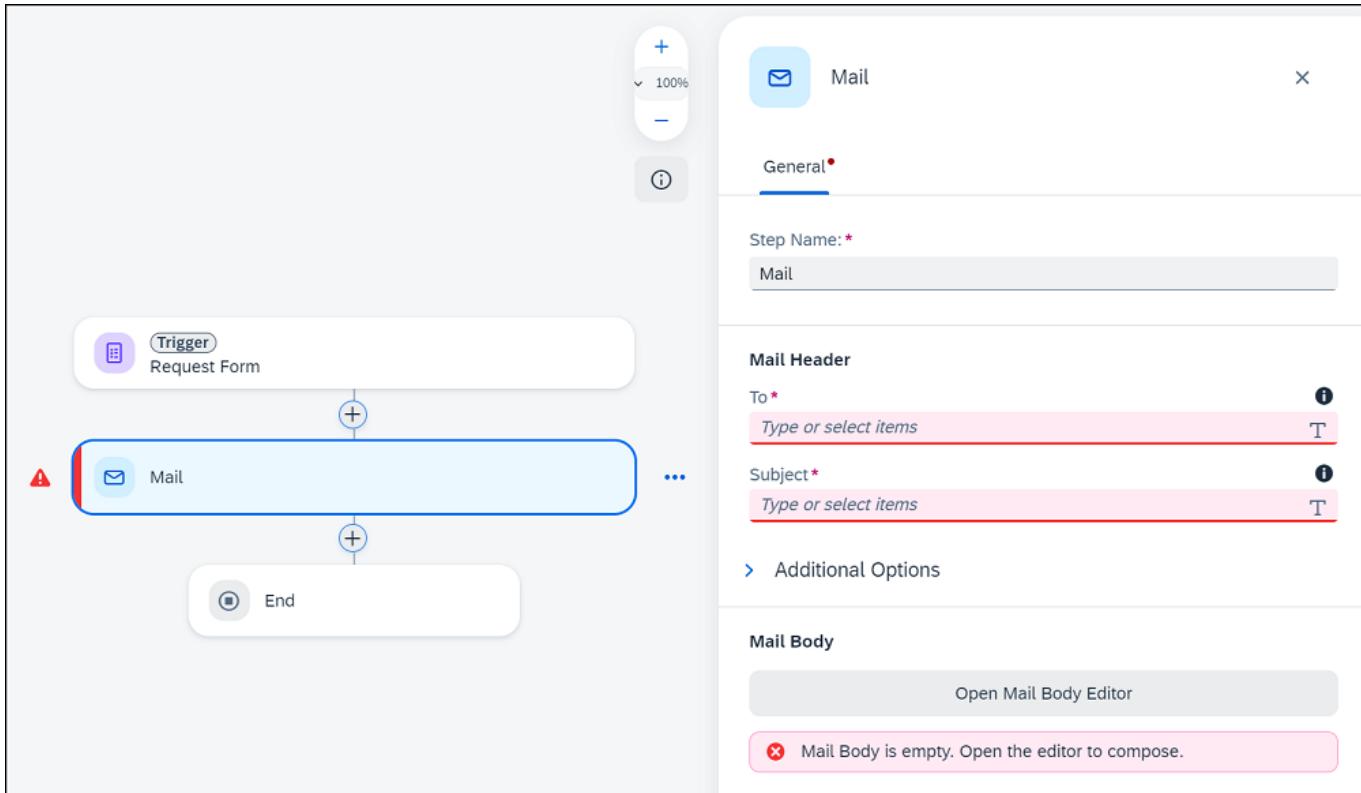


Procedure

1. Either choose **+** or right-click on the process editor canvas and select **Mail**.



The mail notification is added to the process and the settings are displayed in the side panel:



2. Configure the **Mail Header** fields.

Field	Description
To	The recipient of the mail notifications. Either add specific mail addresses or use information taken from process metadata or context. For more detailed guidelines when adding recipients, see Guidelines for Specifying Recipient Users
Subject	The subject of the mail itself. Either add specific text here or use information taken from process metadata or context.
CC	The copied recipient of the mail notifications. Either add specific mail addresses or use information taken from process metadata or context. For more detailed guidelines when adding recipients, see Guidelines for Specifying Recipient Users
BCC	The blind copied recipient of the mail notifications. Either add specific mail addresses or use information taken from process metadata or context. For more detailed guidelines when adding recipients, see Guidelines for Specifying Recipient Users

3. Choose **Open Mail Body Editor** and configure the mail body. The mail body can include the following:

- Text
- Process context information (such as the Form fields in the example)
- Process metadata (such as the 'Process Started By' information)

Edit Mail Body

Value Binding

Search

Form (Trigger)

- T Office Location
- T Name
- T Item Required
- # Value
- Deadline

Process Metadata

- T Process Started By
- T Process Instance Id

Hello

Your request has been received:

Your request reference number is:

The deadline for a decision is:

Thanks

4. Choose **Apply**.

5. Review and fix any missing mandatory fields that are still marked in red.

Results

The mail notification is added to the process, with mails sent to recipients when a process is running.

Add Wait for Duration to a Process

You can add a wait to a process, controlling how long the live process waits (or pauses) before continuing.

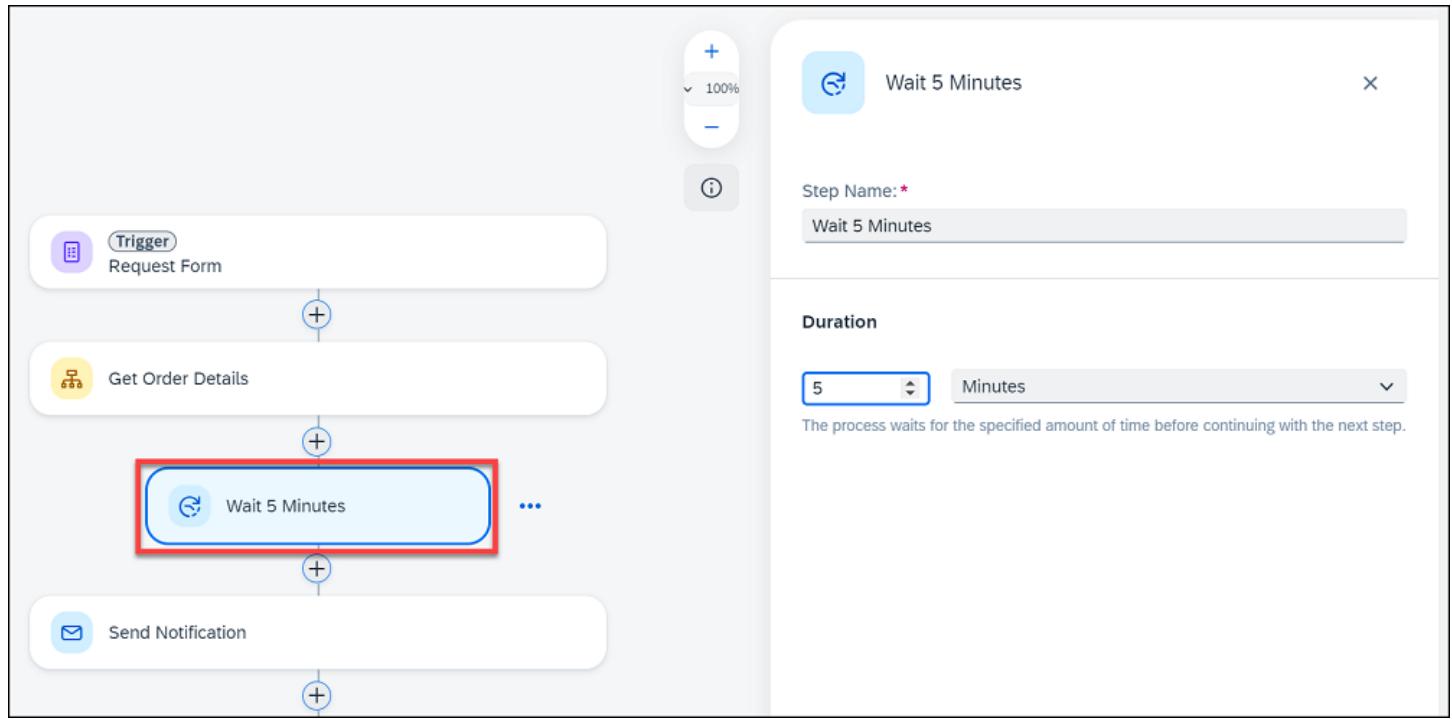
Context

Waits can be added and configured for the following intervals:

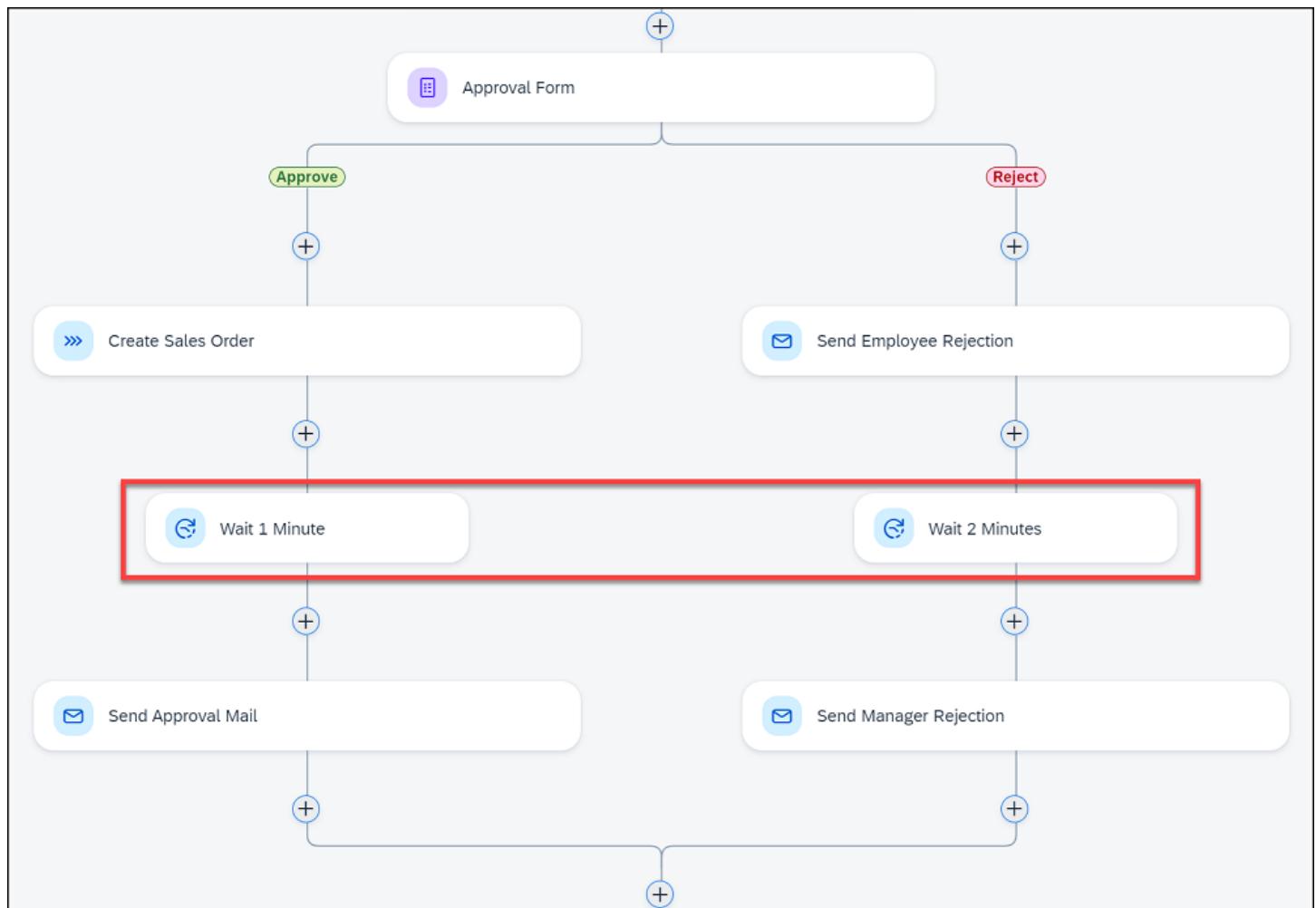
- Minutes
- Hours
- Days
- Months

These intervals then occur from the time the live process first reaches that point. As an example, if the process reaches a 1 hour wait at 09.37, the process continues at 10.37.

In the following example, the process is configured to wait for 5 minutes after an automation has run before the approval is sent:

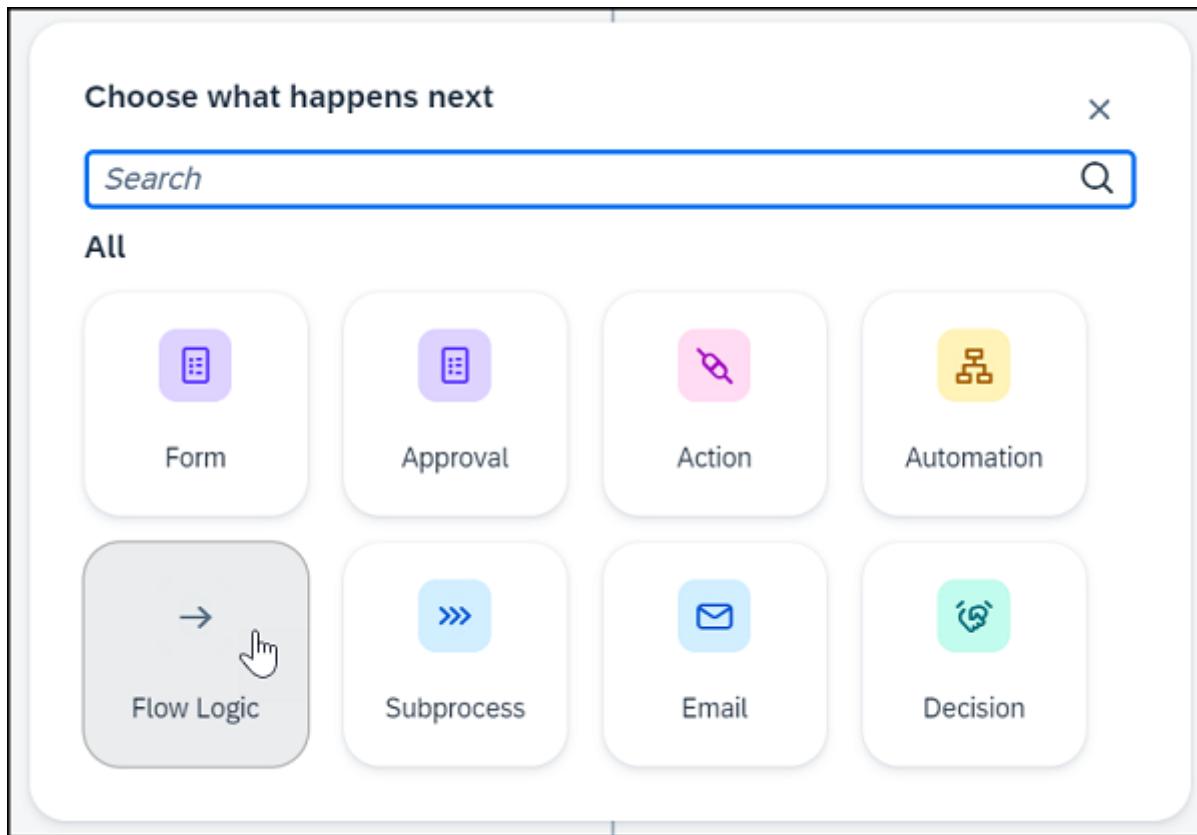


Multiple wait events can be added to the same process, too. In the following example, the process waits for 1 minute after a request has been approved, and 2 minutes after a request has been rejected:



Procedure

1. In the process editor, choose **+ (plus) > Flow Logic > Wait for Duration**.



2. Enter a **Step Name**.
3. To define the duration, select a **Unit** of wait (minutes, hours, days, and months) and enter a **Value**.
4. Save your changes.

Create and Design Automations

Automations are composed of a succession of steps you build in SAP Build Process Automation. An automation can orchestrate multiple activities on different applications and screens available on a specific computer.

Automation Tools

To build an automation, you have access to a list of tools from the side panel.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. At the top, there's a navigation bar with 'SAP' logo, 'Intelligent Robotic Process Automation Cloud Studio', 'Automation Project', and various tool icons. Below the navigation bar is a toolbar with icons for Overview, My First Automation (selected), Share, Generate Package, and Save. The main workspace is currently empty, displaying a placeholder message: 'Start building your automation by dragging activities from the panel to the flow.' A green 'Start' node and a red 'End' node are positioned at the top and bottom of this message respectively. To the right of the workspace is a sidebar titled 'Automation information' with tabs for Tools, Input/Output, and Info. The Tools tab is selected. Under Tools, there's a search bar and a tree view with categories: Automations, Activities, Data, and Controls. The 'Design Console (0)', 'Test Console (0)', and 'Variables (1)' tabs are visible at the bottom left of the workspace.

- *Automations*

You can use a previously created automation from this cloud project, as a tool to include in the flow of another automation.

- *Screens*

An Application Screen is a capture of any application. For more information, see the section about [Capture and Declare Applications](#).

- *Activities*

Activities are used to build the workflow of your automation. They come from the SDK packages imported in the Cloud Studio the first time you create an automation. For more information, see the section about [Automation Activities Provided by SDK Packages](#)

- *Data Types*

A data type is a complex data used to describe a data structure. It can be defined as an input or an output and used throughout your automation. For more information, see [Data Types](#).

- *Controls*

Controls allow you to add tools such as conditions, loops, and scripts to your automation. For more information, see [Add a Control to an Automation](#).

	Condition	Inserts multiple situations defined by an expression, that determines the following step.
	End	Stops the automation.
	For Each	Inserts a sequence of actions to perform on a list of objects such as users.
	Forever	Repeats the step in a loop until it meets the required conditions.
	Repeat	Repeats the step for a defined number of times.

	Loop End	Stops a step set in a loop.
	Screen switch	Inserts multiple situations defined by the screen the user is working on, that determine the following step.
	Custom script	Inserts a step defined by custom Javascript mode.
	Try	Checks if errors occur in a sequence of actions and defines a behavior depending on the error type.
	Stop automation in error	Stops the automation in an error state.