Processes and automation

ELO Flows

Table of contents

ELO Flows - Getting started

What is ELO Flows?

ELO Flows is an ELO module for easily implementing automation and integration tasks.

ELO Flows enables you to automate processes, define simple user actions, and integrate third-party systems without the need for any scripting. This enables you to realize processes based on low-code or no-code methods.

When should I use ELO Flows?

- You want to provide input fields in the ELO clients. These forms can be configured as a tile or as a button on the ribbon.
- You want to create automation tasks across services.
- You want to connect custom developments with ELO. It is possible to develop custom components.
- You'd like to integrate ELO in an existing system or integrate other applications in your existing environment, for example ERP systems, e-mail servers, file shares, etc.

Why should I use ELO Flows?

- With ELO Flows, you can combine functions from different components.
- When configuring flows, you can perform advanced definitions. A declarative transformation language (JSONata) with a content enrichment tool is available for this.
- With a worker model, you can scale job processing.
- You can combine existing components and develop your own custom components to meet individual requirements.
- ELO Flows is part of the ELO license.

What is the difference between ELO Flows and other ELO components?

ELOas

ELO Flows can be understood as the gradual successor to ELOas in the long term. ELOas will continue to be supported and maintained, but it is no longer being developed.

ELO Flows makes it easier to create automation tasks. Ideally, no custom scripting is required at all.

ELO workflows

In conjunction with ELO workflows, ELO Flows makes it easier to map business processes.

Technical logic can be implemented with flows, while business logic is handled via ELO workflows.

Linking these two modules creates a comprehensive process tool.

ELO Business Logic Provider (BLP)

ELO Flows will exist alongside ELO BLP.

ELO BLP will remain the method of choice for complex integration scenarios.

ELO Business Solutions

The ELO Business Solutions are applications based on the ELO Suite.

ELO Flows will provide the technical platform for future ELO Business Solutions, replacing parts of the Common framework.

Quick start

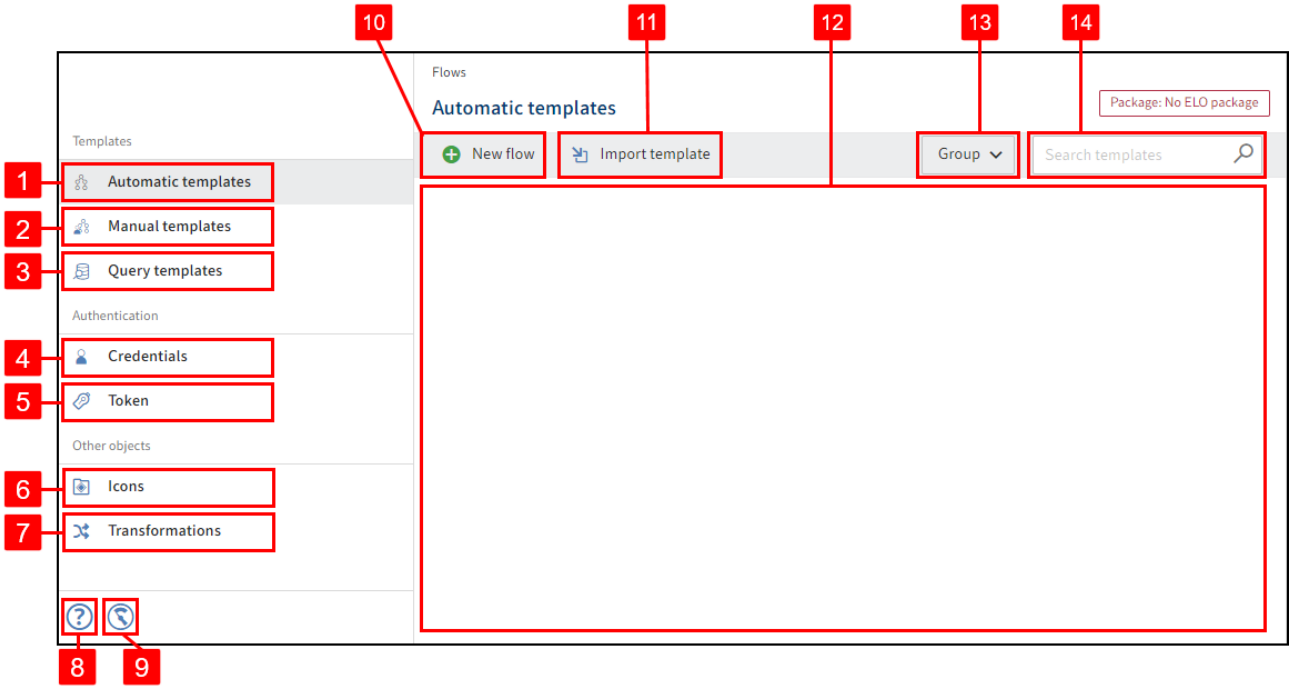
Welcome to ELO Flows. We've prepared a quick start guide for you to help you get started!

Deployment

ELO Flows is provided with the installation of the ELO Server Setup. If it isn't already installed, you can start the ELO Server Setup to do so.

Flow administration

You can access the flow administration area via the ELO Administration Console. ELO Flows is integrated in the administrative environment of ELO packages. You can define and edit the flow logic under Flows.



You have the following options:

1 Create, edit, enable, and disable automations

You can find more information under *Automations*.

2 Create, edit, enable, and disable user actions

You can find more information under *User actions*.

3 Create, edit, enable, and disable data queries

4 Configure credentials for components

You can find more information under Credentials.

5 Create token: You can safeguard trigger events that are triggered by third-party system calls by creating a token with an assigned user context.

You can find more information under Webhooks and token authentication.

6 See and edit icons

You can find more information under Icons.

7 Create transformations

You can find more information under Transformations.

8 Help: Opens the help page in a new tab.

9 Monitoring: Opens the *Status Report* page. There, you will find information about the status of the executed flow.

You can find more information under Monitoring.

10 Create new flow: Create new user actions, automations, or data queries.

11 Import: Import exported flows as a JSON file.

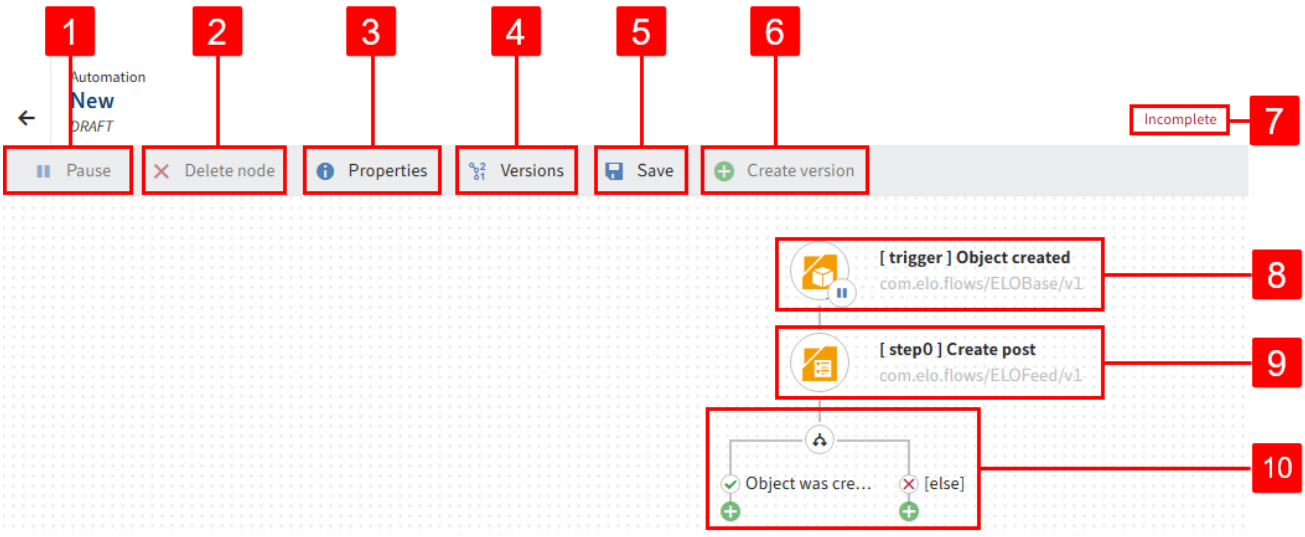
12 Viewer pane: Here you can see the flows currently available.

13 Group existing flows

14 Browse existing flows

15 Package assignment: Here, you see which package the flows are assigned to.

Flow designer



With the flow designer, you can edit existing or new flows.

The flow designer contains the following elements:

1 Pause/Continue: Set a flow to *Active* or *Deactivated*. Triggers are not called for deactivated flows. For user actions, the respective buttons are hidden in the client.

2 Delete node: In the flow designer, triggers, services, and branches are shown as nodes. When configuring a flow, you can select existing nodes and delete them from the flow using *Delete node*.

3 Properties: You can change the name of the flow, as well as export and delete the flow here.

4 Versions: You can select the versions to delete them, edit them in the flow designer, or activate or deactivate them. When you edit a flow, it will initially be labeled as a *DRAFT*. Created versions can be found under the name you assigned them.

5 Save: Save regularly so no changes are lost. If the flow is activated, saving results in hot deployment in the ELO client.

6 Create version: Create a new backup of the current version. Enter a version name. You can also add a comment to the version.

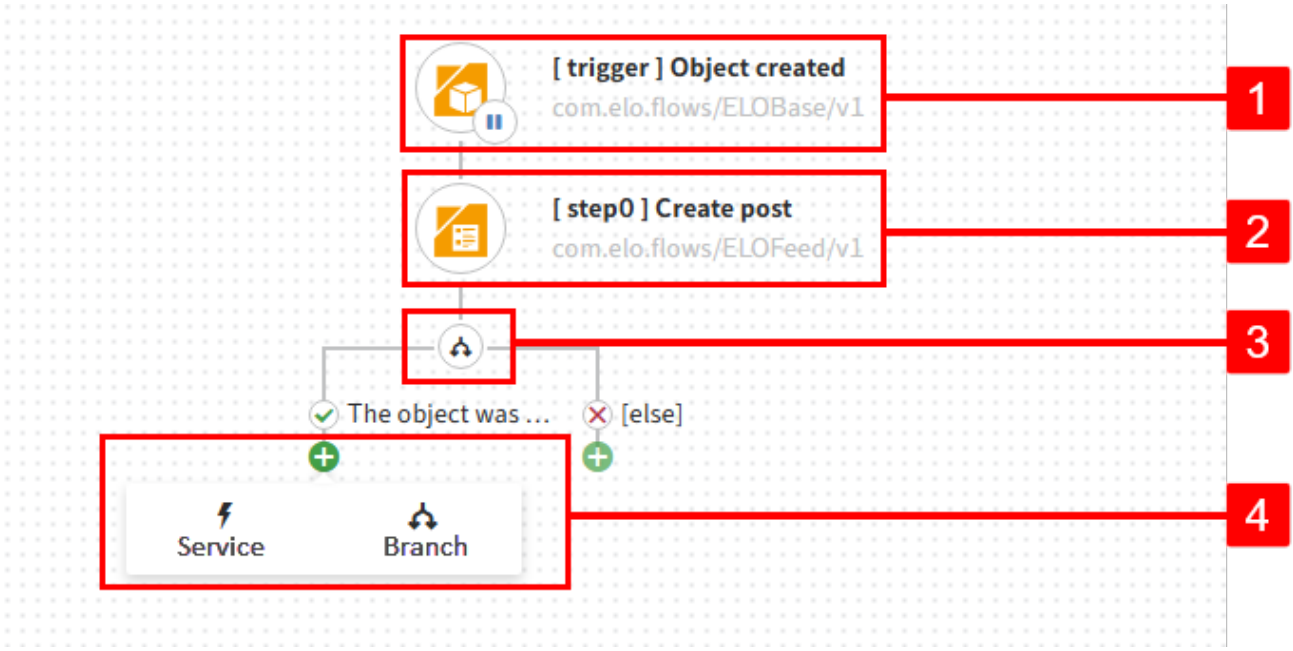
7 Flow status: *Active*, *Deactivated*, or *Incomplete*

8 Trigger node

9 Service node: The additional *[Step]* signifies the order of service nodes you have assigned to the flow.

10 Branch: You can also create a branch with a true-false condition before each step. You can nest this branch in other branches or continue with services.

Structure of a flow



A flow consists of the following elements:

1 Trigger: Starts the flow

2 A series of services: Steps in the flow

3 Branch node: True-false condition that determines how the flow proceeds

4 Add node: Adds a service or a branch

Components

Components are connecting modules or connectors to systems. You can use standard components that are provided, such as the *ELO metadata and objects* component, or create your own components that interact with the systems you want. You will find more information on the standard ELO Flows components in the *Components* section.

Triggers

In ELO Flows, flows are started by triggers. There are different types of triggers.

- Automatic triggers: Among other things, these listen to ELOix events (provided by the ELOix event connector).
- Manual triggers: These provide a button to manually start the flow in the clients. Behind this button is a definition for the activity UI, which the activities of an entry in the ELO client for ELO Flows map.
- Webhook: A webhook trigger creates a REST endpoint that can be called by external systems.

There are different types of flows:

- Automations
- User actions
- Data queries

The difference is in the initial trigger.

Automations start with ELO events, among other things. If you install a component offering triggers, use them here. In addition to the ELOix events, the action of receiving an e-mail is conceivable with an e-mail component. You could respond to this and run a flow.

User actions start with a manual trigger. These allow users to make entries within an ELO client. This data is transferred to the flow and can be processed there.

Data queries start with a webhook or with ELO events.

Services

Services are functions that are called in the flow steps.

Activities

Activities are user actions that are opened with a button in the ELO clients. The button definition is configured for a user action. This allows you to quickly map common use cases, such as a form activity or a document upload activity.

Refer to *User actions* under Add activities for more information.

Installation & operation

Overview

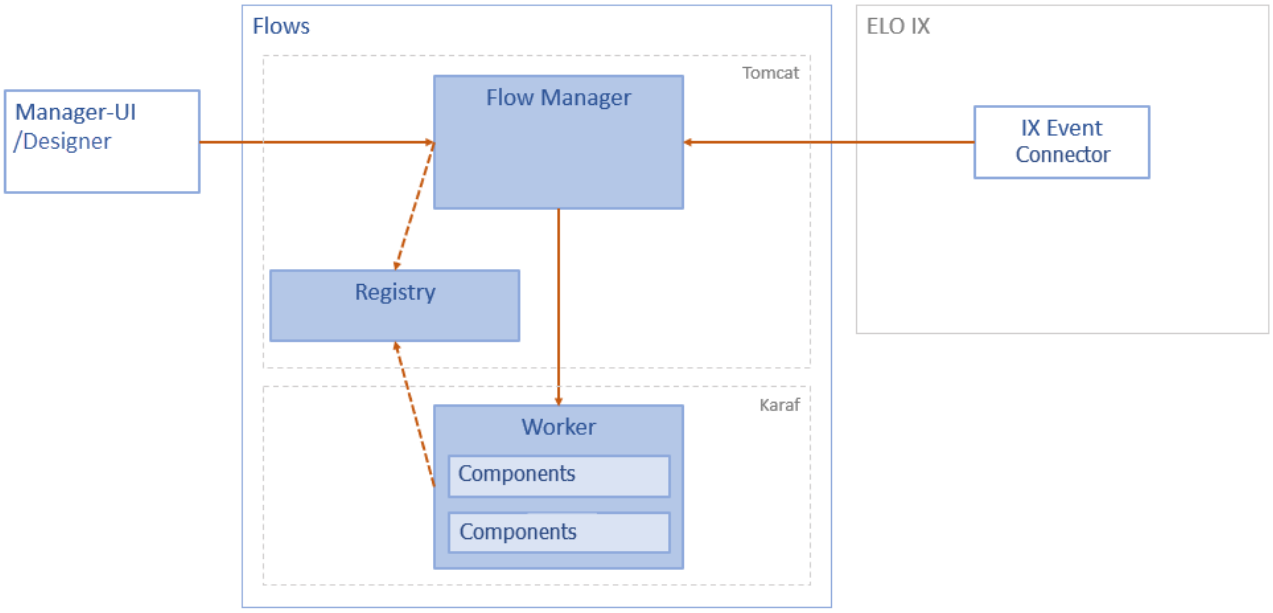
You will find more information about the following in the Installation & operation section:

- Architecture
- Installation and operation

You will also find tips for troubleshooting and collecting error data to speed up analysis in the ELO Community or by Business Support.

Architecture

The architecture of ELO Flows provides a web-based front end and a scalable worker model. ELO Flows also enables component management with hot deployment support through the integration of Apache Karaf.



Modules

ELO Flows is made up of different modules.

The ELO flow manager and the registry run on a Tomcat as a web app (as do the other ELO modules). The workers run on an Apache Karaf.

Flow manager

The flow manager is a main instance in ELO Flows. It is responsible for starting new flows and managing existing ones.

The flow manager provides the interface for flow administration. Flows can be created and managed via this interface. The interface can also access registry information.

Flow administration

The flow administration area is the interface in the ELO Administration Console. You can manage flows here. The app is provided together with the required interfaces by the flow manager.

Workers

The workers are independent containers. Components are deployed in the workers. More than one container can be used in parallel for load balancing purposes.

Registry

The registry is a main instance in ELO Flows. The workers authenticate with the registry and register their components there. The registry also checks whether the workers can be reached and is notified if the components of a worker change.

ELOix event connector (OSGi plug-in)

The ELOix plug-in is responsible for forwarding ELOix events to the flow manager.

Currently, a differentiation is made between the following events:

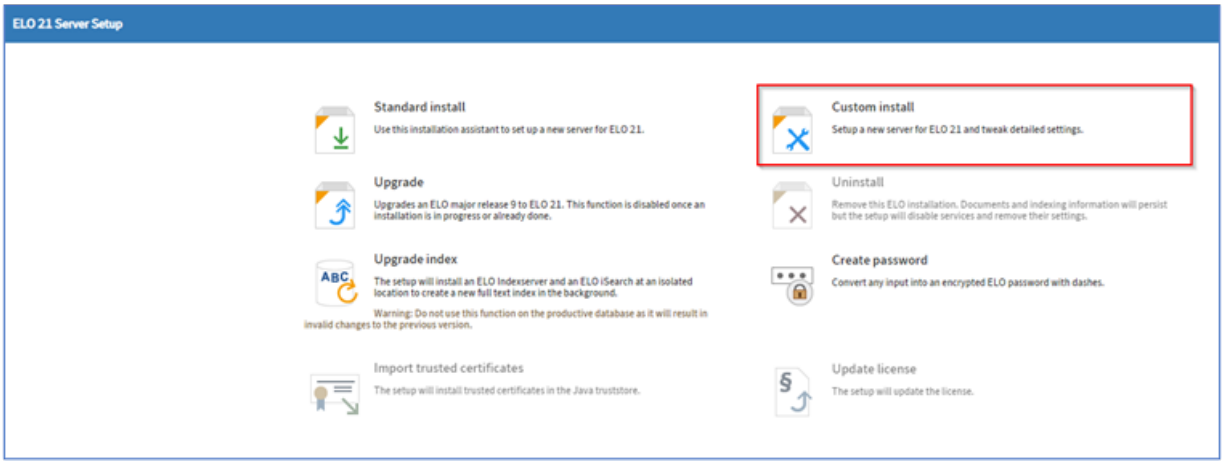
- Creation of a SORD
- Update of an existing SORD
- Call of a dynamic keyword list from a form

Components

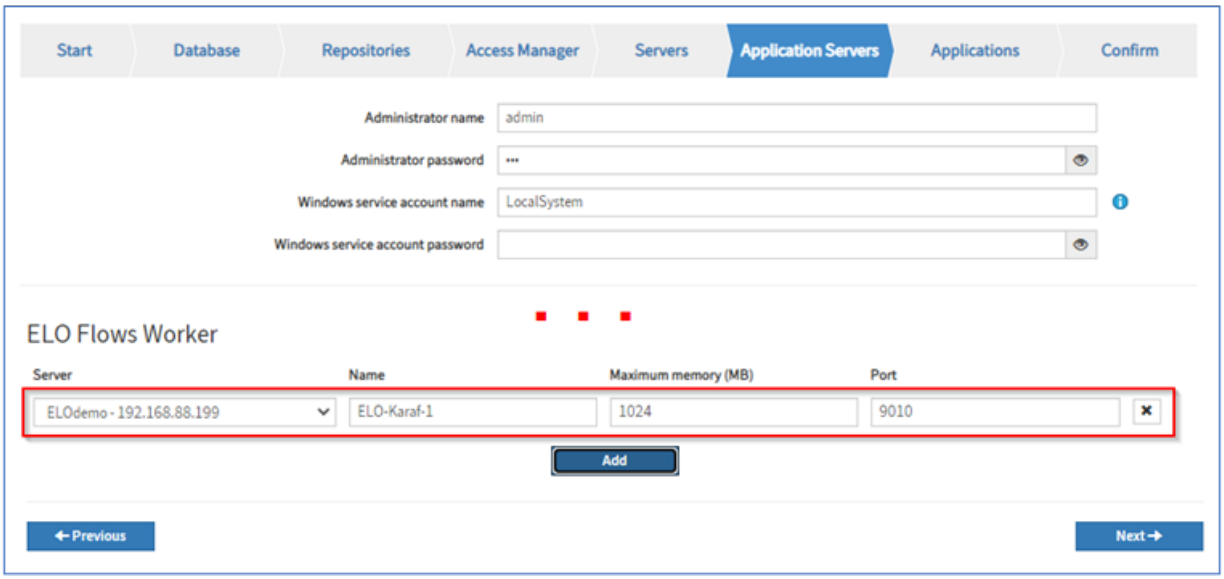
The components are developed as OSGi plug-ins. They provide triggers and services.

Installation

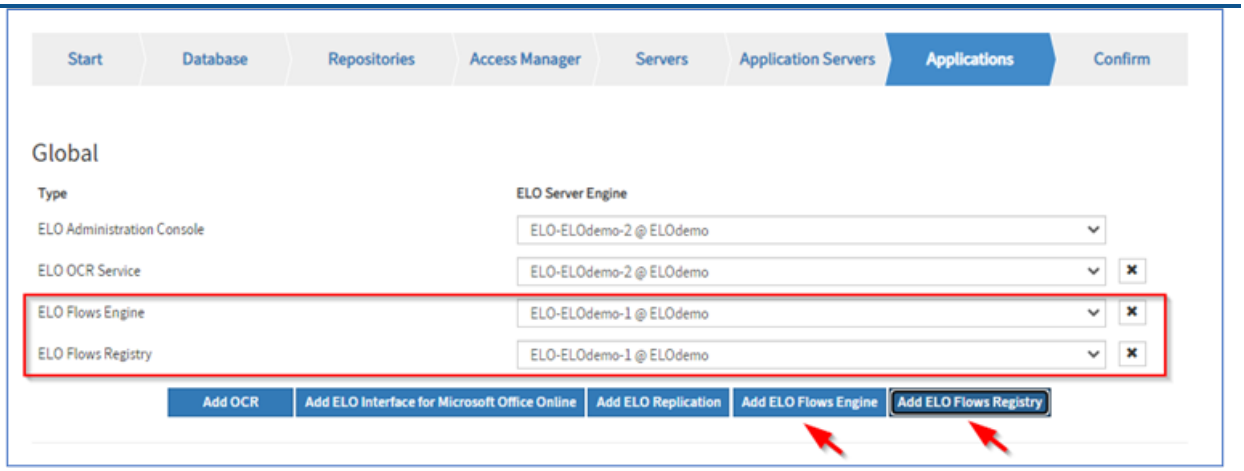
1. Start the ELO Server Setup.



2. Select the option *Custom install*



3. On the *Application Servers* page, add an entry for the ELO Flows worker with *Add*.



4. On the *Applications* page, add the two ELO Flows server applications *ELO Flows Engine* and *ELO Flows Registry*.

5. Finish the ELO Server Setup.

Manager authentication

After installation, the flow administration area can be opened in the ELO Administration Console.

Alternatively, you can open it directly via a link structured as follows:

`http://<HOSTNAME>:<PORT>/ix-<REPOSITORYNAME>/plugin/de.elo.ix.plugin.proxy/flows/`

Troubleshooting

Troubleshooting

Troubleshooting can be performed in multiple steps:

1. Check whether all modules are running properly. Multiple status pages are available here.

   The Monitoring section explains how to reach them.

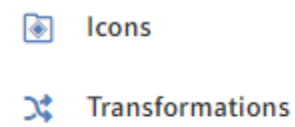
2. Check the log files.

   You will find the log files worker.log and manager.log in the ELO Flows working directory. Errors begin with the prefix ERROR.

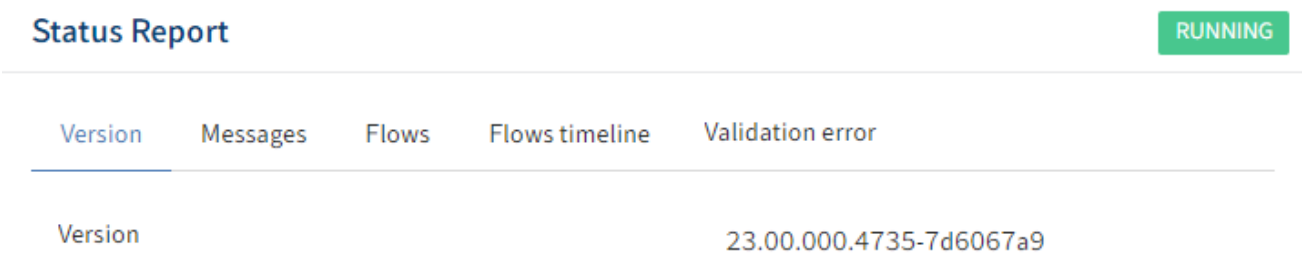
With the complete error stack trace, you can check the cause for the error or share it with support, development, or the community.

Monitoring

ELO Flows provides a monitoring tool. In this tool, you can monitor the processing of flows and analyze errors.



1. Select *Monitoring* in the flow administration area.



The *Status Report* page opens. The page consists of the following tabs:

-

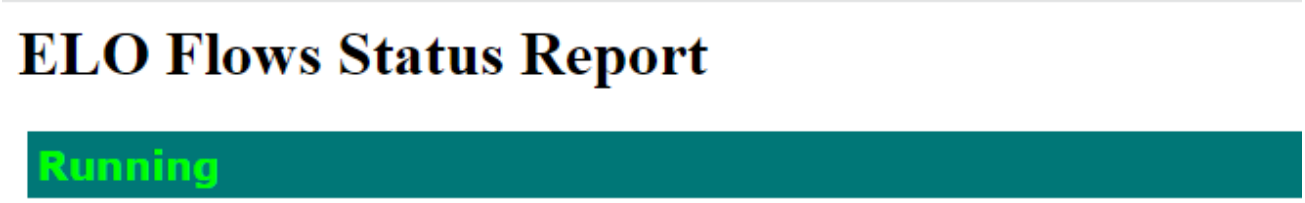
Version: Shows the version of ELO Flows currently being used
- Messages: Shows current status messages
- Flows: Shows the available flows with the corresponding statistics
- Flows timeline: Shows the timeline of executed flows and, if applicable, errors in the list. The list can be filtered.
- Validation errors: Shows any errors and error sources in flows and transformations

Alternatively, you can perform manual troubleshooting, supported by the following status pages.

Status pages

Multiple status pages are available, which you can check. In addition to the currently installed version number, they also indicate the online status of the modules:

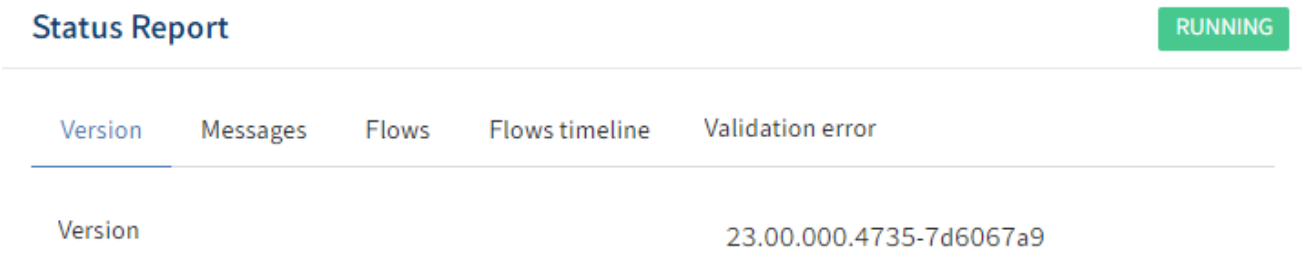
Simple status page of the ELO Flows Manager



The manager offers a very simple status page that only shows the status and can be used without authenticating:

```
http://<hostname>:<port>/flows/status
```

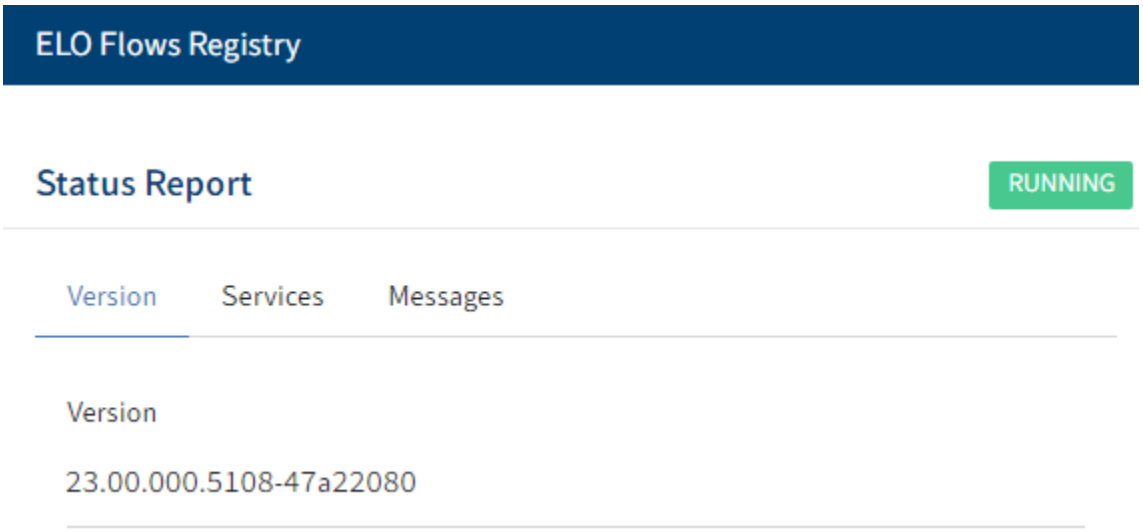
Flow administration status page



You can reach the flow administration status page at the following link:

```
http://<hostname>:<port>/ix-<repository>/plugin/de.elo.ix.plugin.proxy/flows/#/status
```

ELO Flows registry status page



You can reach the ELO Flows registry status page at the following link:

```
http://<hostname>:<port>/registry
```

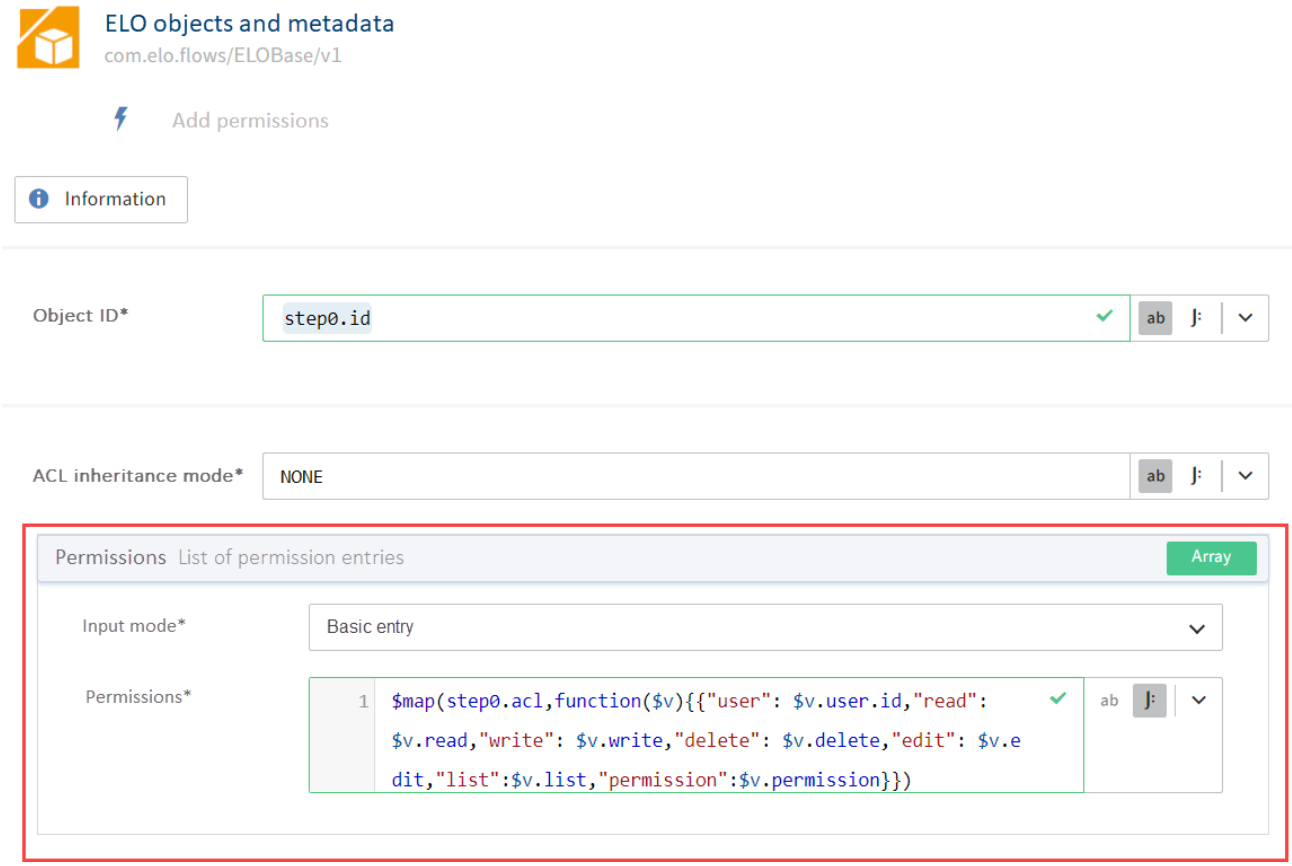
> **Information**
>
> Change the `host name` and `port` based on the respective installation.

Known issues

Transfer ELO permissions

Output ACL objects are not currently compatible with input ACL objects. To transfer permissions, they have to be mapped. This requires a transformation.

Workaround



1. Create the service *Add permission*.
2. Navigate to the *Settings* tab.
3. Enter the *ObjektID*.
4. Select an *ACL inheritance mode*.
5. Open the list settings menu for permissions by selecting *Array*.
6. Select *Basic entry* as the *input mode*.
7. Transfer the following JSONata call in the JSONata editor under *Permissions*:

```
$map(stepX.acl, function($aclItem) {
  {
  "user": $aclItem.user.id,
  "read": $aclItem.read,
  "write": $aclItem.write,
  "delete": $aclItem.delete,
  "edit": $aclItem.edit,
  "list": $aclItem.list,
```

```
  "permission": $aclItem.permission
  }
}
```

8. Adapt the context to the source object of the permissions.

Handling of date values with the ELO Flows JSONata editor

Internally, ELO uses the ISO format without a delimiter (e.g. 20130701).

The `$toMillis()` function in ELO Flows does not currently support conversion without a delimiter.

Workaround

Delimiters must therefore first be inserted in the ISO date.

`$milliRes` is the date converted into milliseconds which you can then use to continue working.

```
(
$iDate := activities.calendar.date;
$cDate := $substring($iDate,0,4) & "-" & $substring($iDate,4,2) & "-" & $substring($iDate,6,2);
$milliRes := $toMillis($cDate);
)
```

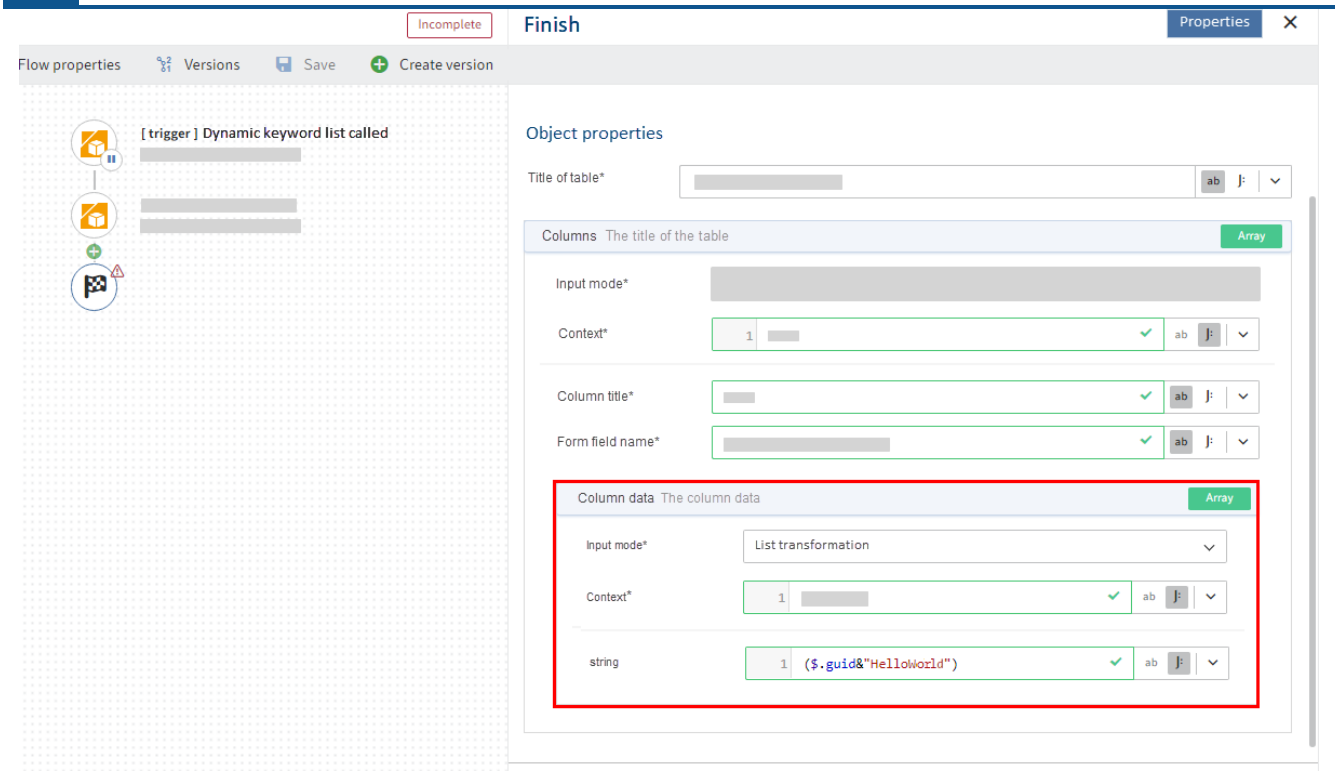
Concatenation in transformation fields

ELO Flows enables concatenation within transformation fields, for example in the *List transformation* field.

When making an entry in the JSONata editor, place your expression in brackets. Without the brackets, the concatenation is applied to the entire expression instead of to the individual elements of the array.

> **Please note**
>
> Concatenation in transformation fields is only possible in expert mode in the JSONata editor. Select the mode with the *J:* icon.

Example concatenation of GUID and string




```
/*Input in JSONata editor*/

($.guid&"HelloWorld")

/*Result of the concatenation*/

[
  "guid1HelloWorld",
  "guid2HelloWorld",
  "guid3HelloWorld"
]
```

Migrate packages from old data

The ELO Flows migration is carried out once. A flag is then set in the database indicating that the migration of old data has already been performed. Migration is not started over for this reason. Loading non-migrated data permanently destroys it.

There isn't currently an event for package import able to correctly handle the imported old data. Use either your own import from ELO Flows (old flows are correctly migrated here) or the workaround described below.

Workaround

1. Stop the ELO Flows Manager service.
2.

Delete the *migration flag* in the database.

Example:

```
delete from configurations where cfgcomponent = 'flows'
       and cfgkey = '23.00.000'
       and cfggroup = 'TEMPLATE_MIGRATION'
```

3. Start the ELO Flows Manager service.

Result: Old data is migrated again the next time ELO Flows starts.

Problems in the ELO Java Client when running manual flows without activities

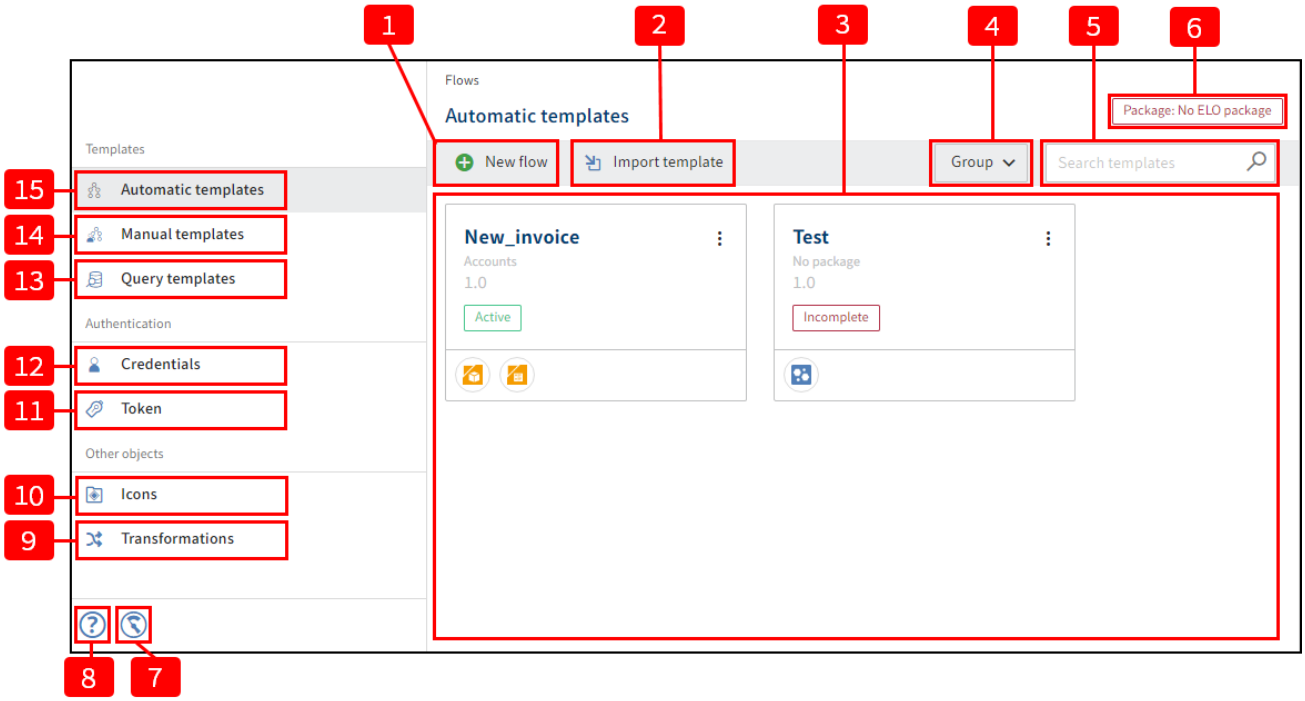
Due to the cookie policy, it isn't possible to execute manual flows without activities in the ELO Java Client if they are running on the same machine as the ELO Flows Manager. The reason for this is that the ELO Flows Manager is unable to verify the request if the client doesn't transfer a JSESSION-ID. Calls of manual flows with activities, however, always work, as they are executed in the client's web view. The ELO Web Client is not affected by the problem.

Use

Flow administration

The flow administration area is the central application for managing ELO Flows. You can reach it in the ELO packages context.

Home screen



1 Automations: On start-up, the *Automations* menu item is selected. You can create flows with automatic triggers via this menu item.

You can find more information under *Automations*.

2 User actions: Selecting the *User actions* menu item shows you the current flows with manual triggers.

You will find more information under *User actions*.

3 Data query: With the *Data query* menu item, you can create special flows that use triggers in the form of queries.

4 Credentials: Under *Credentials*, you can configure credentials for each component.

You will find more information under Credentials.

5 Token: With the *Token* menu item, you create tokens. You can safeguard trigger events that are triggered by third-party system calls by creating a token with an assigned user context.

You will find more information under Webhooks and token authentication.

6 Icons: Select the *Icons* menu item to view existing icons and create new ones.

You will find more information under Icons.

7 Transformations: You can create transformations via the *Transformations* menu item.

Refer to Transformations for more information.

8 Help: Opens the help page in a new tab.

9 Monitoring: Select *Monitoring* to open the *Status Report* page. There, you will find information about the status of the module and executed flows.

10 New flow: Create a new user action, automation, or data query.

11 Import: Import exported flows as a JSON file.

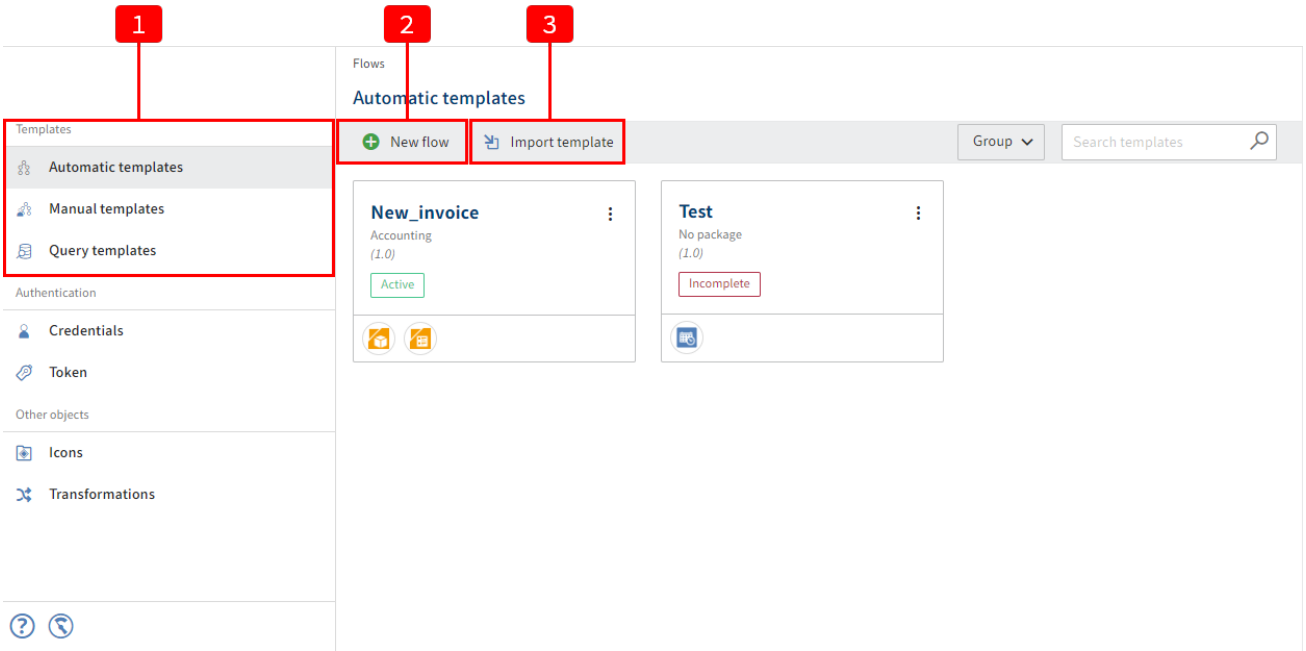
12 Viewer pane: Here you can see the flows currently available.

13 Group: You can sort the available templates via the *Group* drop-down menu.

14 Search templates

15 Package: Here, you see whether the current flows are part of a package in ELO.

Create new flow



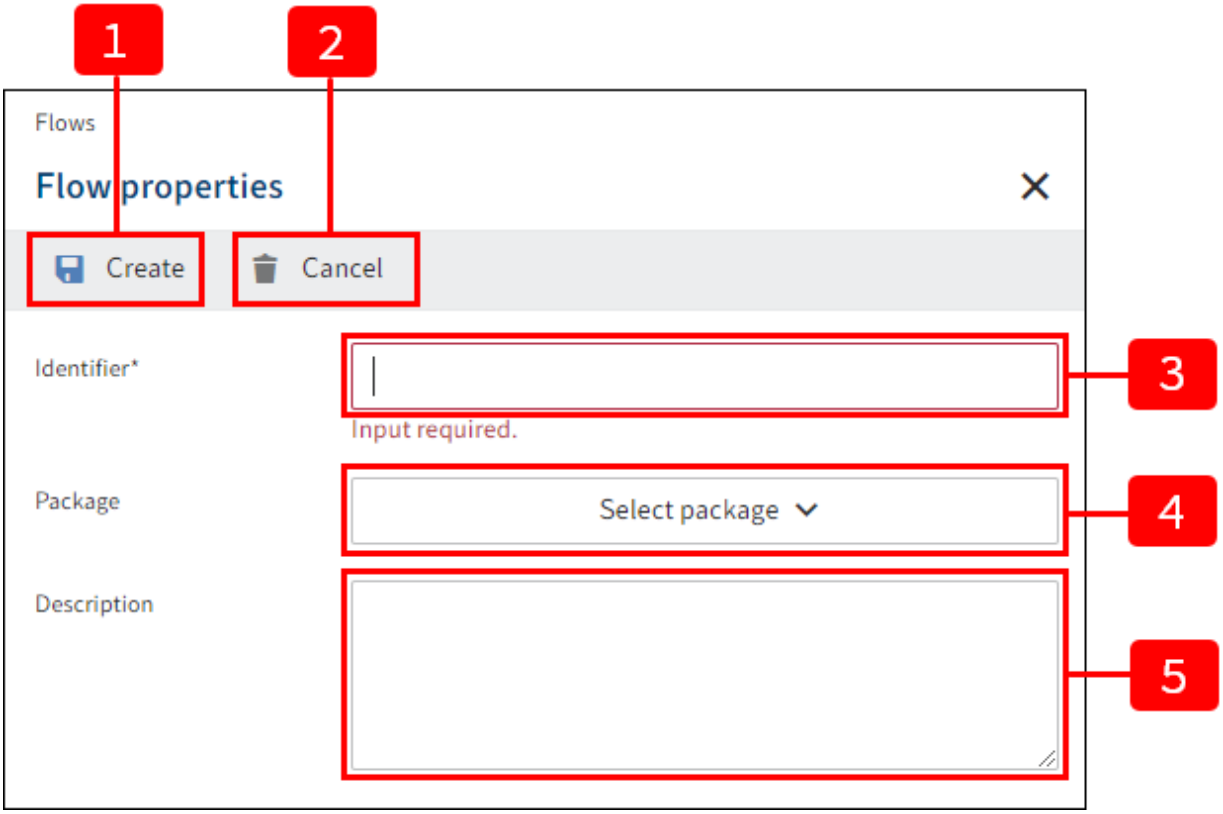
1. On the ribbon, select a type for your flow: *Automations, User actions*, or *Data queries* (1) .

2.

Depending on the flow type, select *New automation, New user action*, or *New data query* (2).

Alternative: By selecting *Import* (3), you can import exported flows as a JSON file.

3. Enter the corresponding metadata for the flow to be created:



1 Create: Saves the entries. The new flow is created.

2 Discard: Cancels flow creation

3 Name: Unique, technical name of the flow

> **Information**
>
> Enter a name for the flow that briefly describes what the flow does.

> **Information**
>
> Do not use special characters.
>
> The following characters are allowed:
>
> - a-z
> - A-Z
> - 0-9
> - äöü
> -

ÄÖÜ
- ß
- _

4 Package: Shows which package context the flow is created in

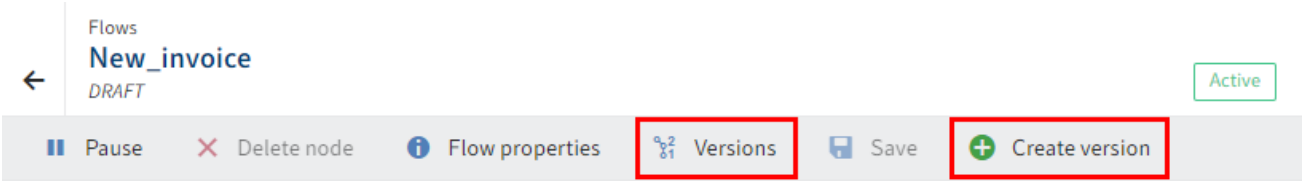
5 Description: Detailed description of the function of the flow. Use this field if your flow is complex.

4. Select *Create*.

   You will see the new flow in the overview. Initially, it will show *Incomplete*, as no triggers or services have been defined yet.

Flow versions

In the flow designer, you can save different versions of your flows.



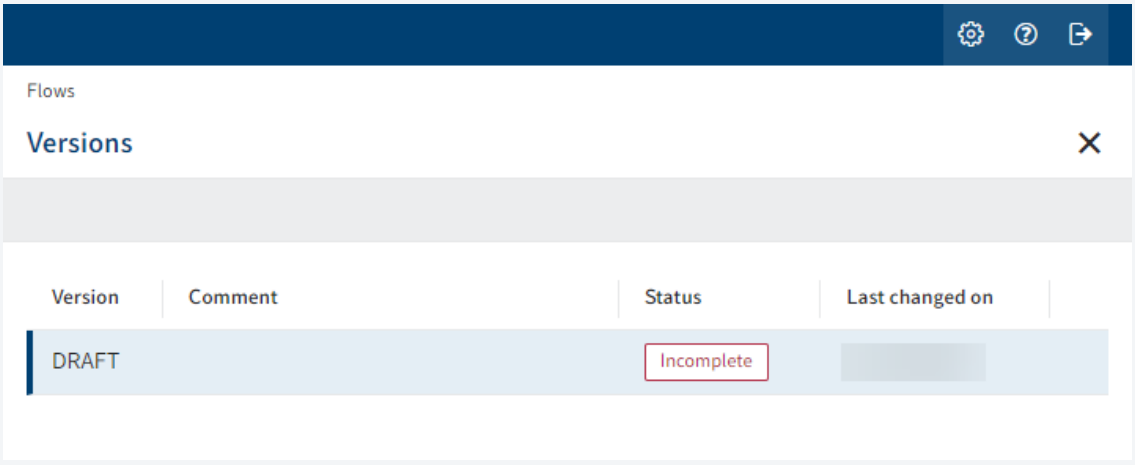
Create version

Create a backup of the current version of your flow with the *Create version* function.

Method

1. Edit a flow in the flow designer.
2. Save your version by selecting *Save*.

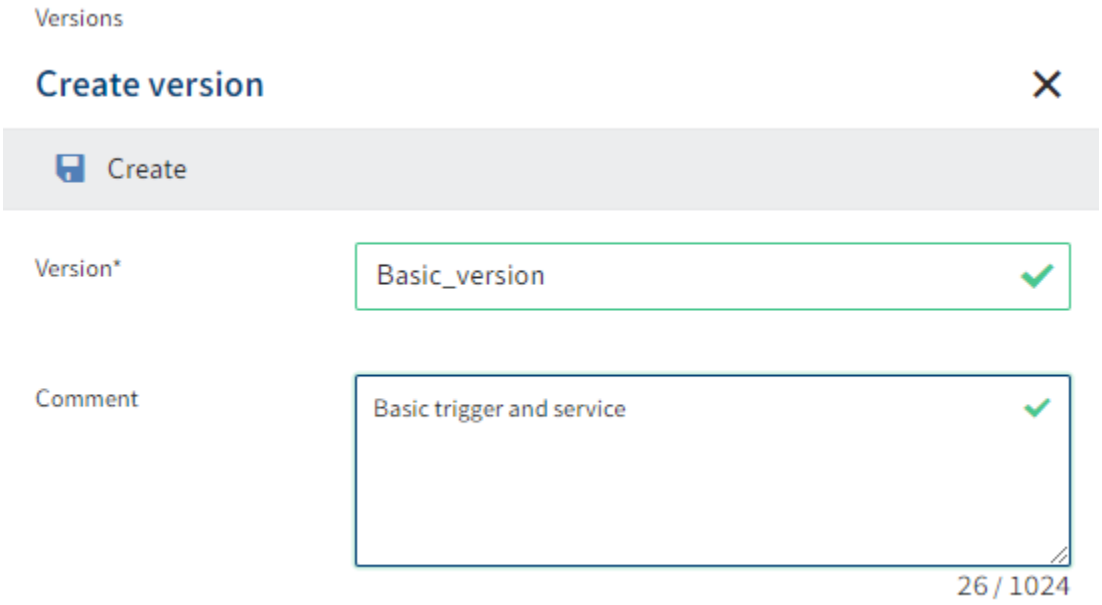
Information



When you select *Save*, ELO Flows automatically generates the *DRAFT* version. You can find this draft version in the *Versions* overview.

Every time you save, the *DRAFT* version is updated and overwritten.

3. To back up the current version, select *Create version*.



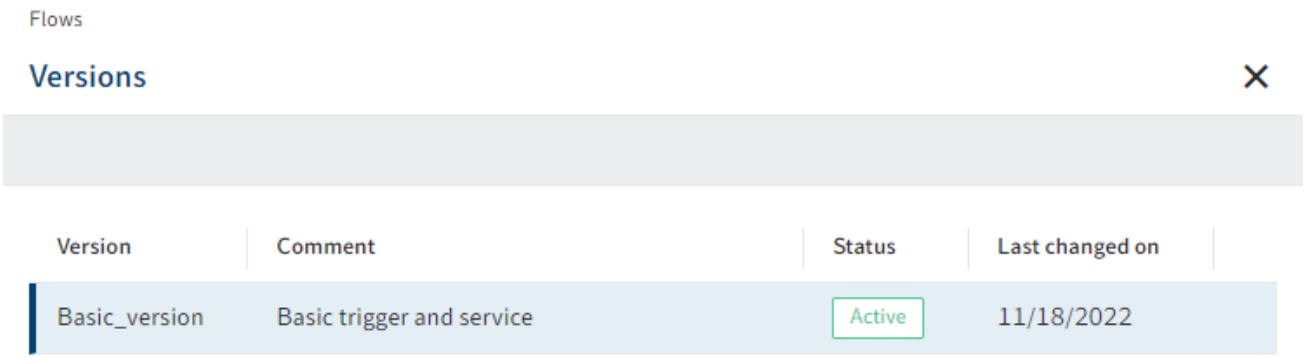
The *Create version* window opens.

4. Enter a name for your version of the flow.

   Optional: Comment on the version.

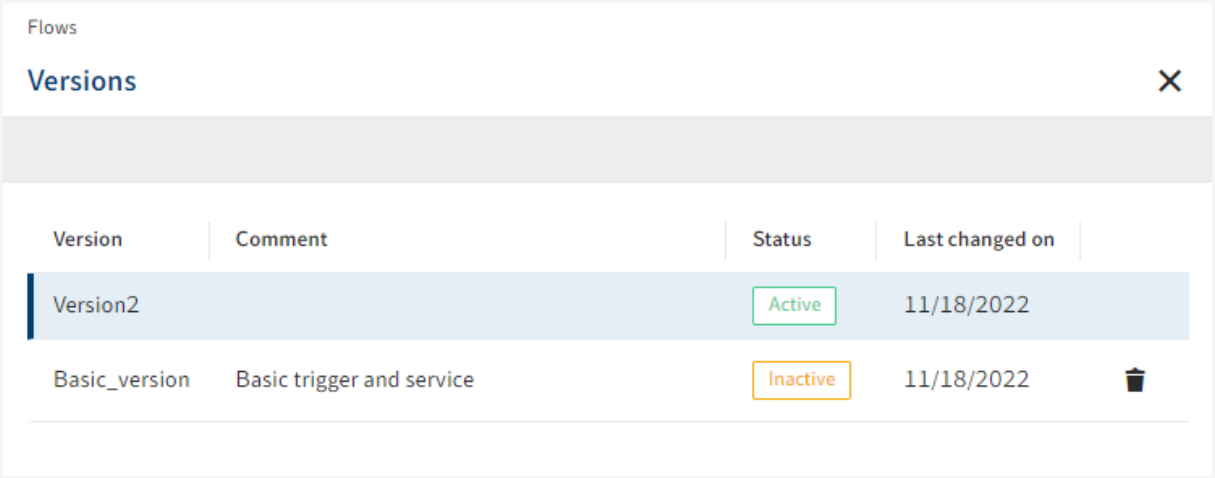
5. Select *Create* next to the floppy disk icon.

Result



The newly created version is now available in the *Versions* overview. There, you can select the individual versions to continue processing them and activate/deactivate them in the flow designer, for example.

Information



The new version assumes the status of the previous DRAFT version. If the DRAFT is active, the new version is automatically activated. If the DRAFT is deactivated, the new version is automatically deactivated.

Select version and continue processing

You can select created versions individually to continue processing the flow based on them.

Method

1. Select *Versions*.

   The *Versions* window opens.

2. Select the version you would like to process.

   If the desired version is already selected, you can close the window by clicking the X icon.

Result

The *Versions* window closes. You are back in the flow designer.

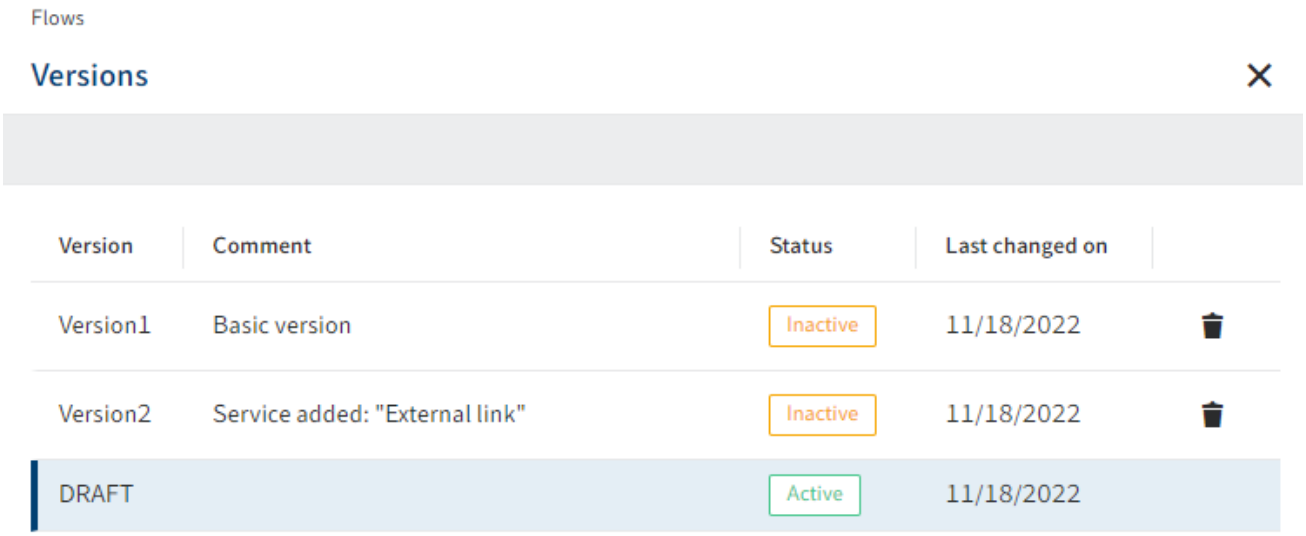
Edit version

Using the *Edit* function, you can modify a flow based on the current status of a version.

Method

1. Select the version in the *Versions* window.
2. Select *Edit*.
3. Edit the flow.
4. Save your draft version.

Result



The version you used to edit the flow remains unchanged. You can find the current status as a DRAFT version in the *Versions* window.

Outlook

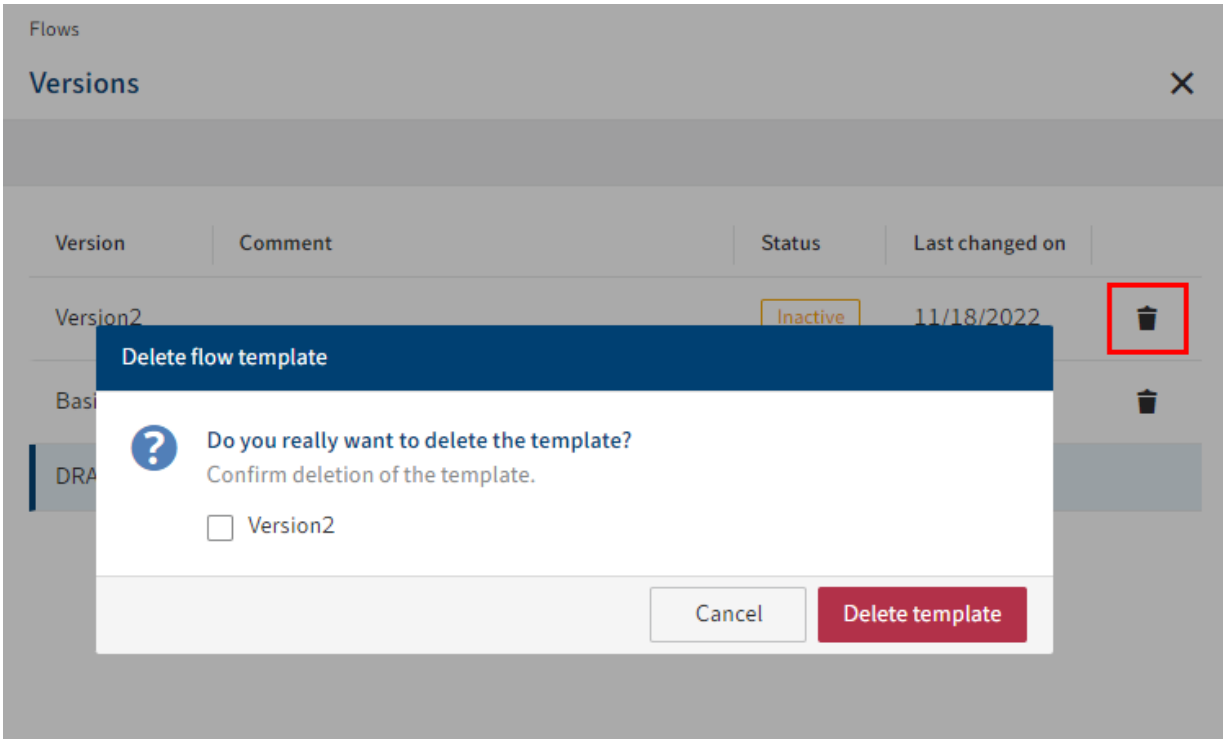
Create a new version to save the current status as a version.

Delete version

To delete a version, it must first be deactivated.

1. Select the version to delete in the *Versions* window.
2. Select *Pause* in the flow designer.
3. Select *Versions*.
4.

Select the recycling bin icon beside the version to delete it.



5. Confirm deletion in the *Delete flow* dialog box.

Result

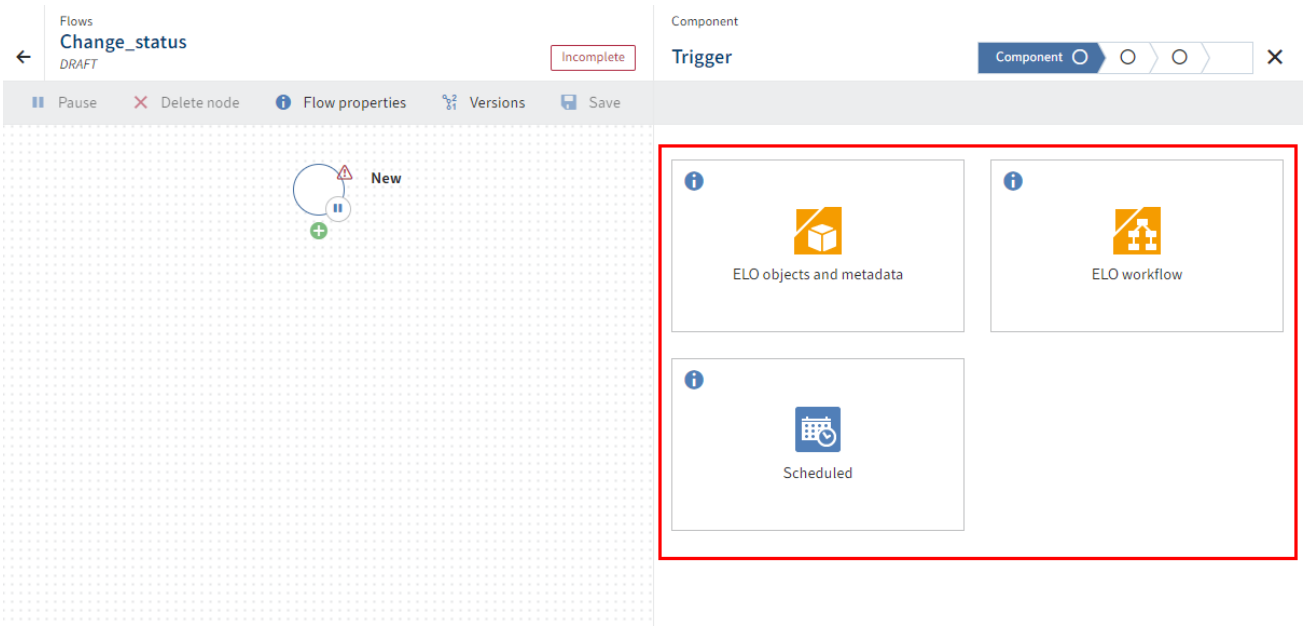
The version you selected is deleted and is no longer shown in the *Versions* window.

Automations

Flows in the Automations area consist of an (automatic) trigger and of at least one or more services. Optionally, you can make flows more complex by using branches.

Select trigger

First, select a trigger. Proceed as follows:

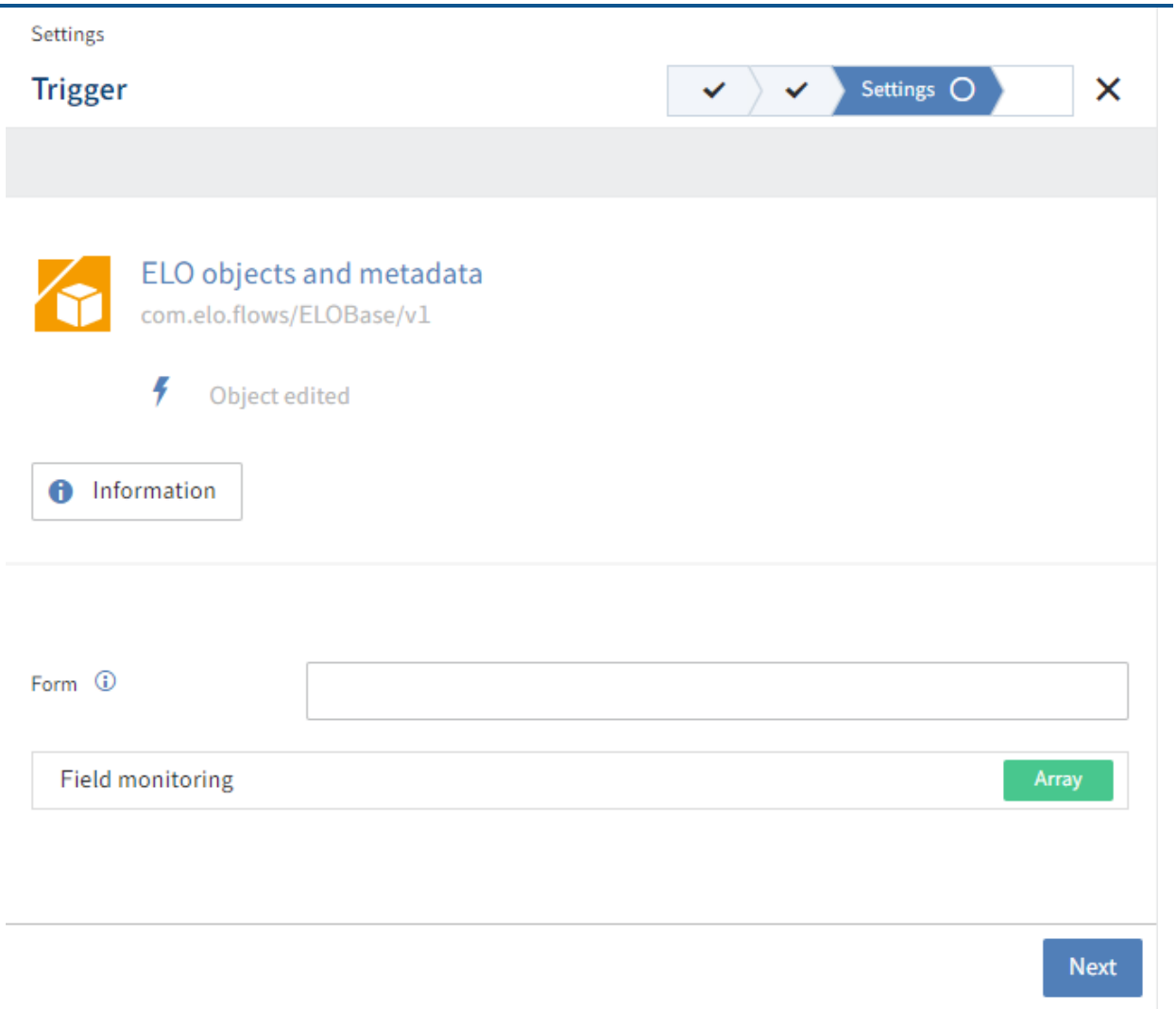


1. Select a *component* whose trigger you want to use.

   The flow designer automatically jumps to the following *Trigger* tab.

2. Select the *trigger* you want to start this flow.

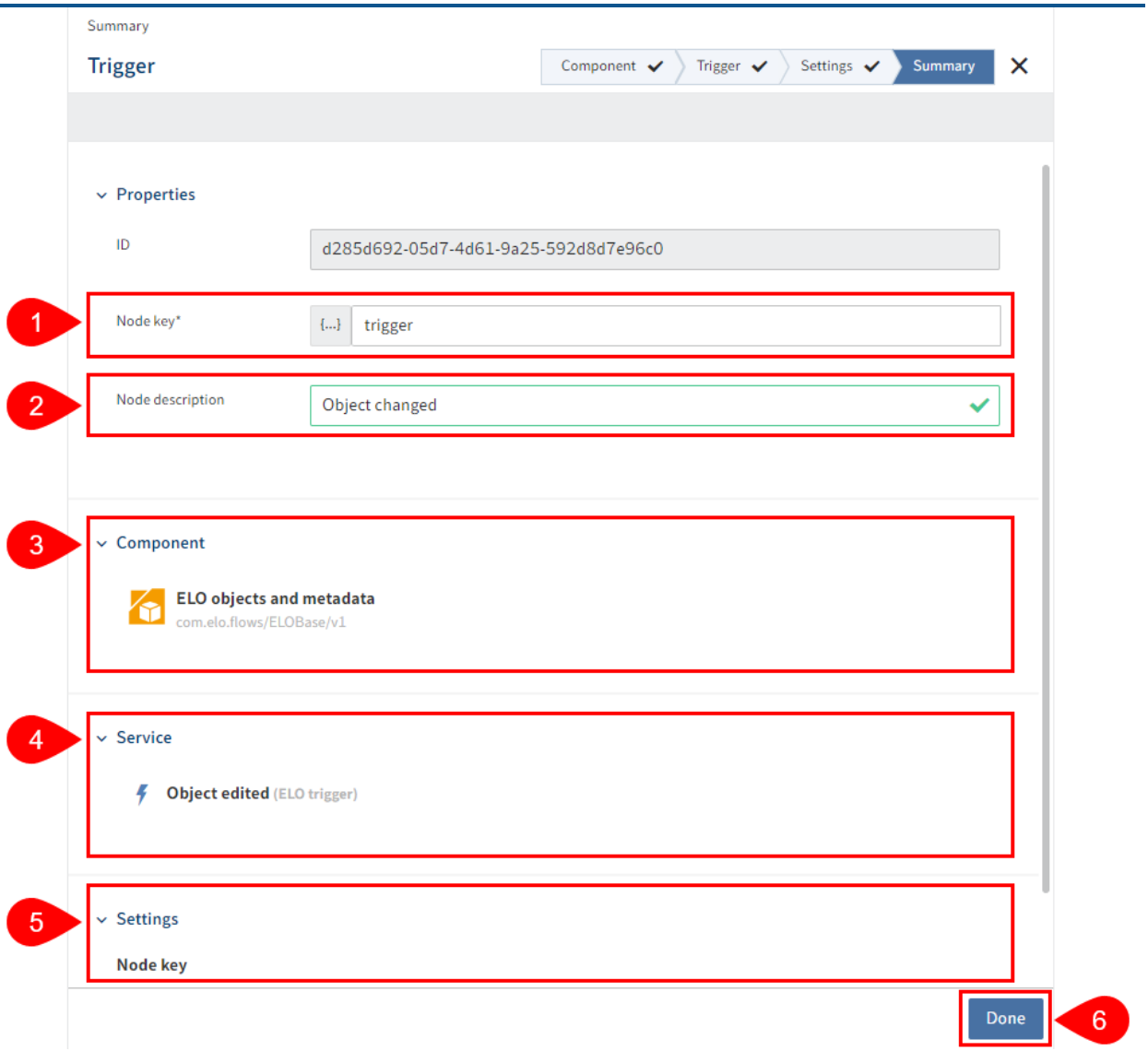
   The flow designer automatically jumps to the following *Settings* tab. The setting options for the trigger are shown here:



In this example, you can restrict the *Object changed* trigger to specific metadata forms by entering the metadata form name. Other triggers have other settings here.

> **Information**
>
> Under Components > General information, you will find more information on completing fields with gen. 1 and gen. 2 metadata in the flow designer.

3. Select *Next* to open the *Summary* tab.



This tab shows a summary of all settings:

1 Node key: Name of the node shown in square brackets in the flow designer.

2 Node description: More detailed description of the node.

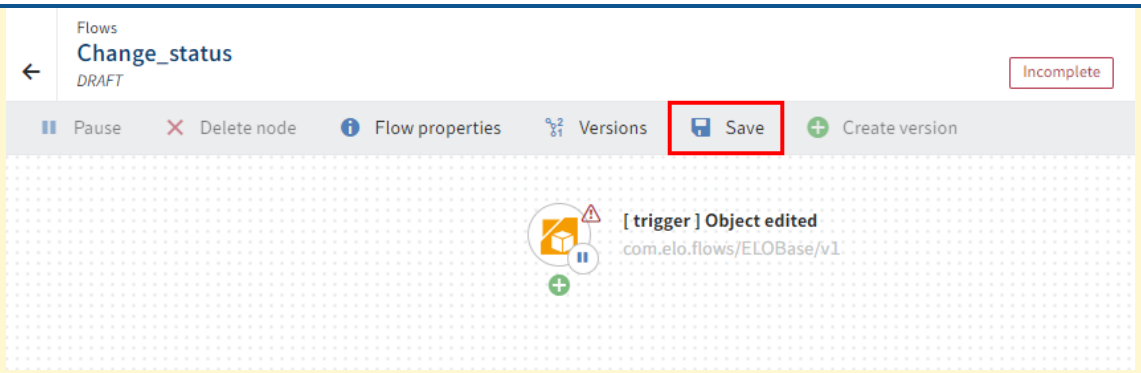
3 Component: Displays the selected component

4 Service: Displays the selected trigger with a copyable path to the endpoint

5 Settings: Displays the node key

4. Select *Done* to finish configuring the trigger.

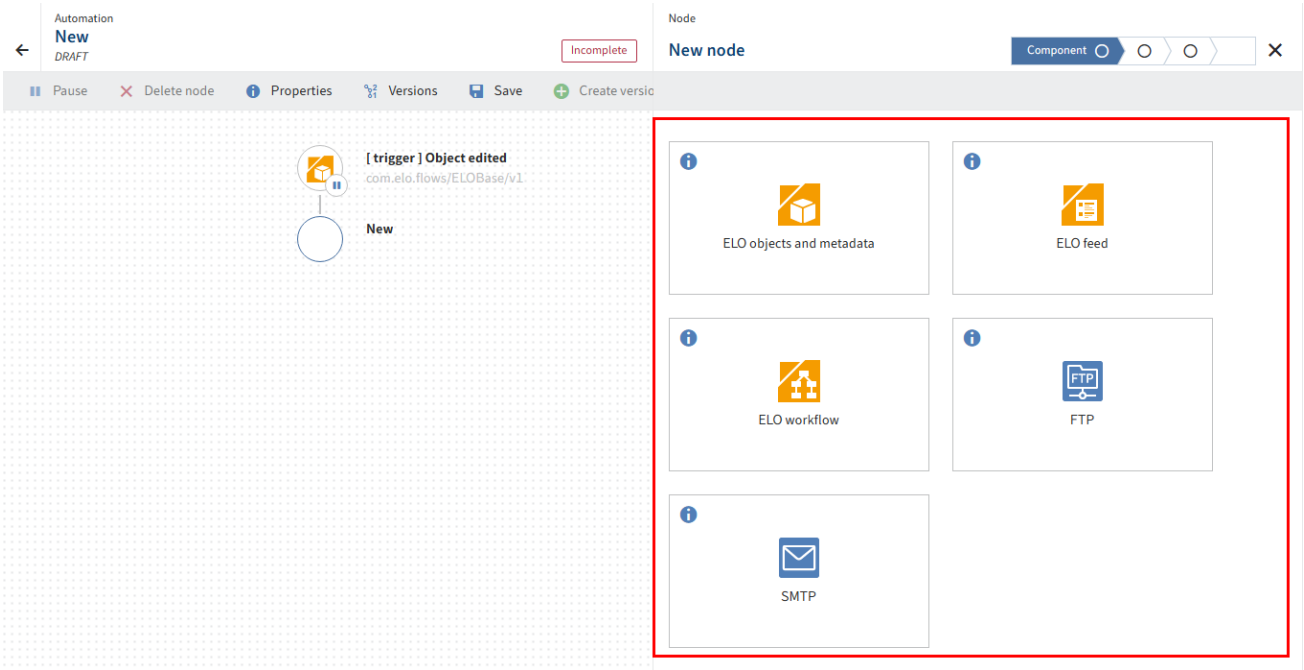
> **Please note**



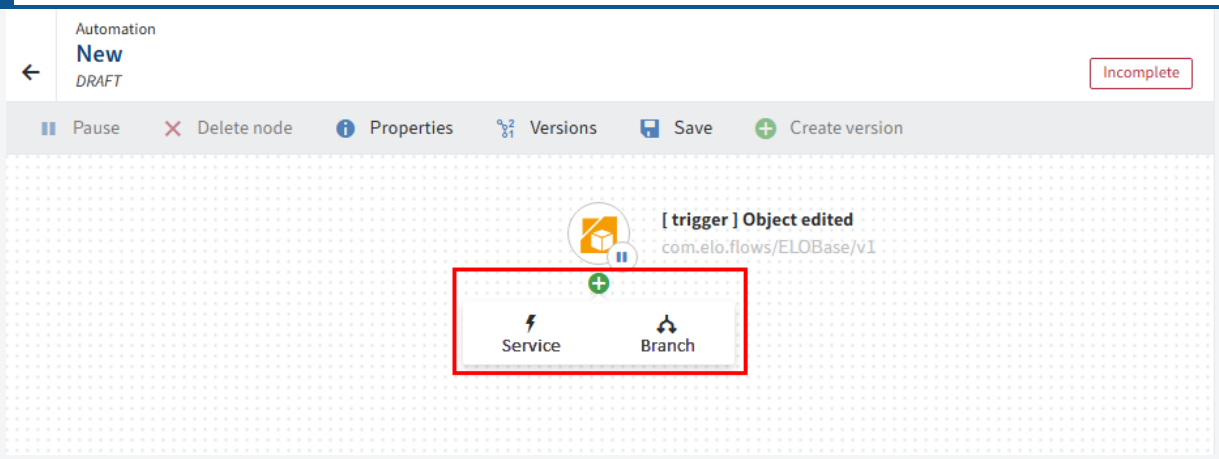
With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

Add service



Information



Before each step, you can decide whether to create a branch with a true-false condition or to add a service. You can nest branches in other branches or continue with services.

You will find more information on branches in ELO Flows in this documentation under Advanced flow.

1. To add a service, select the round plus icon next to the trigger node and select *Service*.

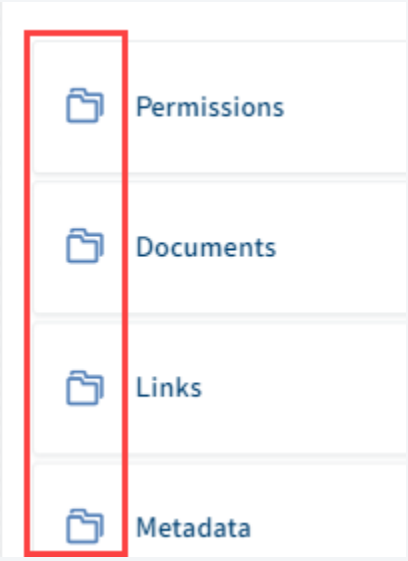
   The configuration area with the available components opens for the new node.

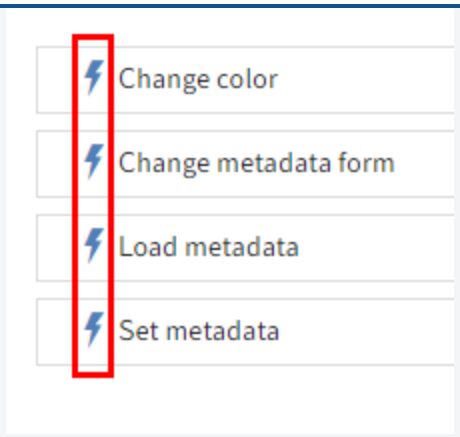
2. Select a component.

   By selecting the component, you are automatically taken to the next tab with the services available for the component.

3. Select the desired service.

Information

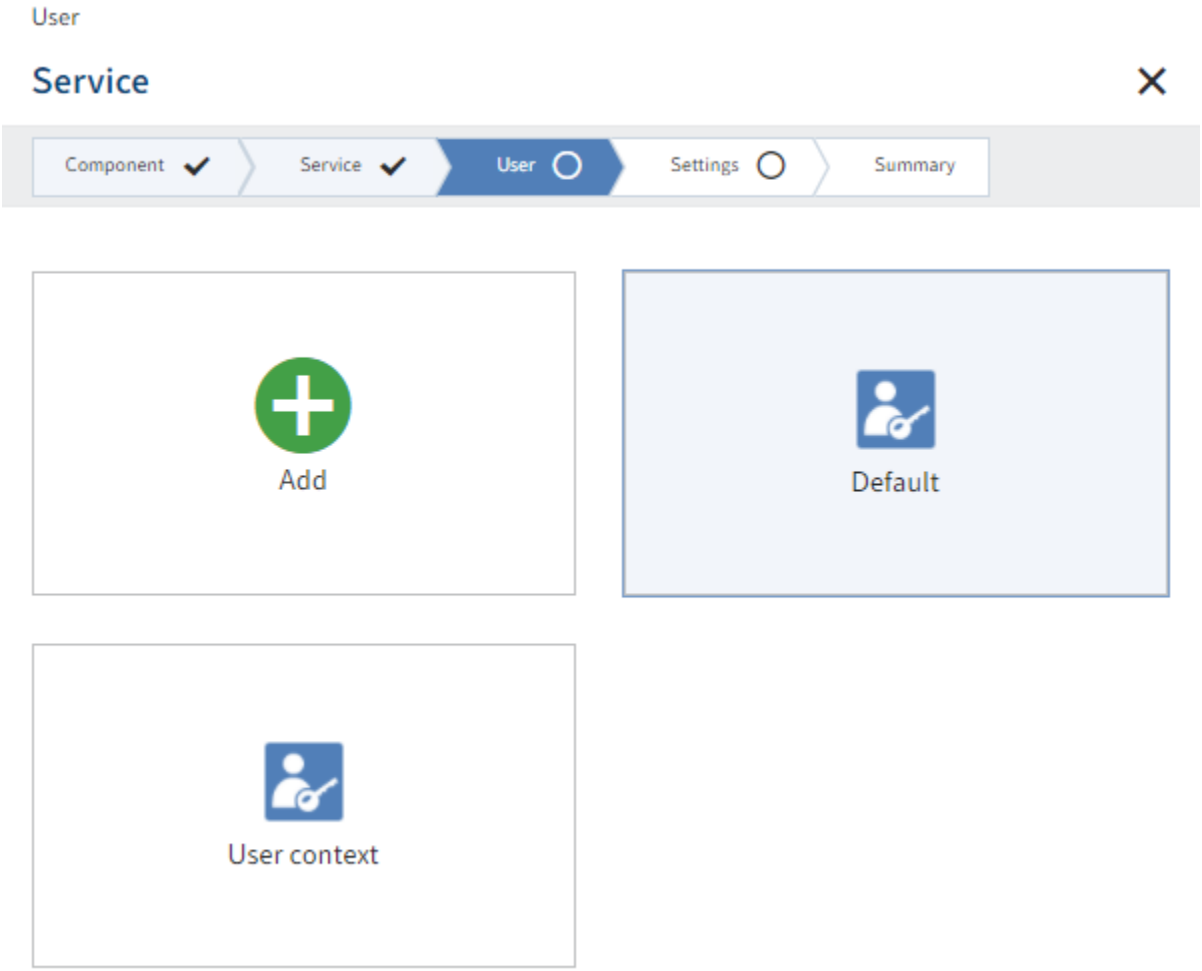




Services can be listed individually (lightning icon) or in groups (folder icon).

Selecting the service automatically takes you to the next tab with the user settings.

4. Select a connection.

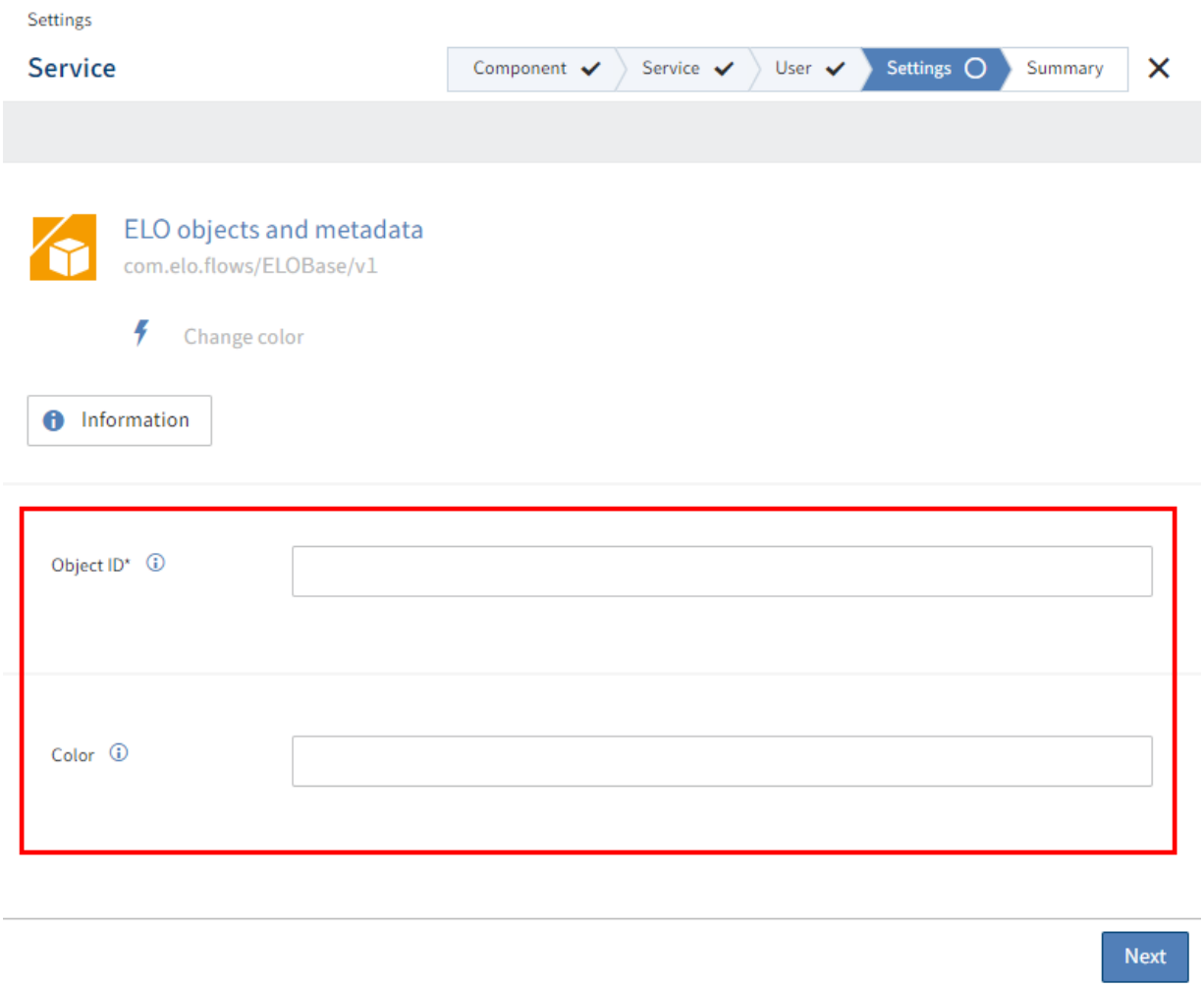


Information

If you've already configured a user under Credentials, you can select the user here. Otherwise, you can select the default connection or add a new one.

5. Confirm your selection with *Next*.

   The *Settings* tab appears.



6. Configure custom data for the service.

Information

You can complete the fields with *static* or *dynamic values*. Dynamic evaluation is performed based on *JSONata*. With this transformation language, you can access event objects of the previous nodes and triggers in addition to using transformations or simple functions.

You will find more about the use of JSONata in ELO Flows under JSONata editor.

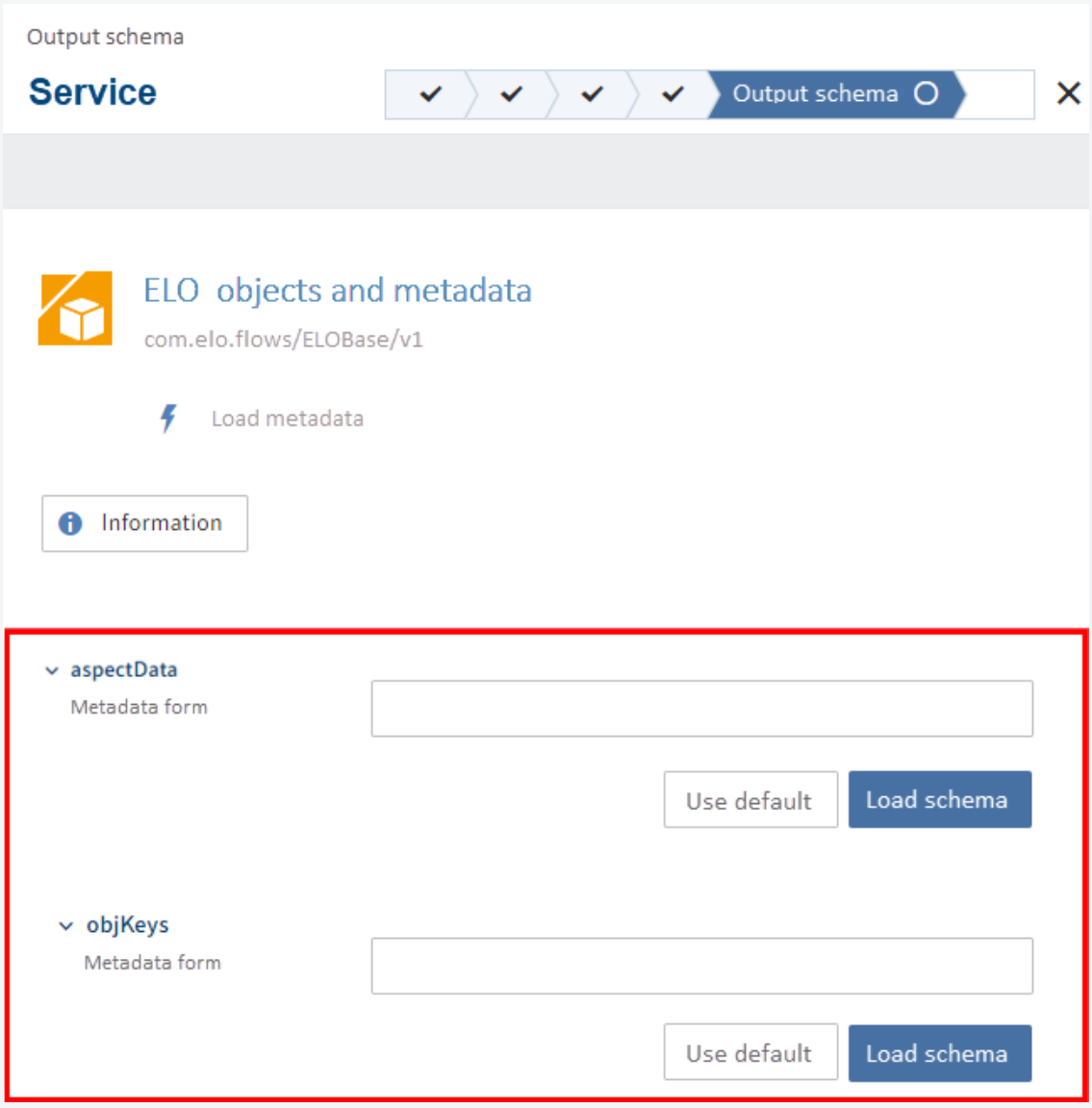
7.

Confirm your selection with *Next*.

Information

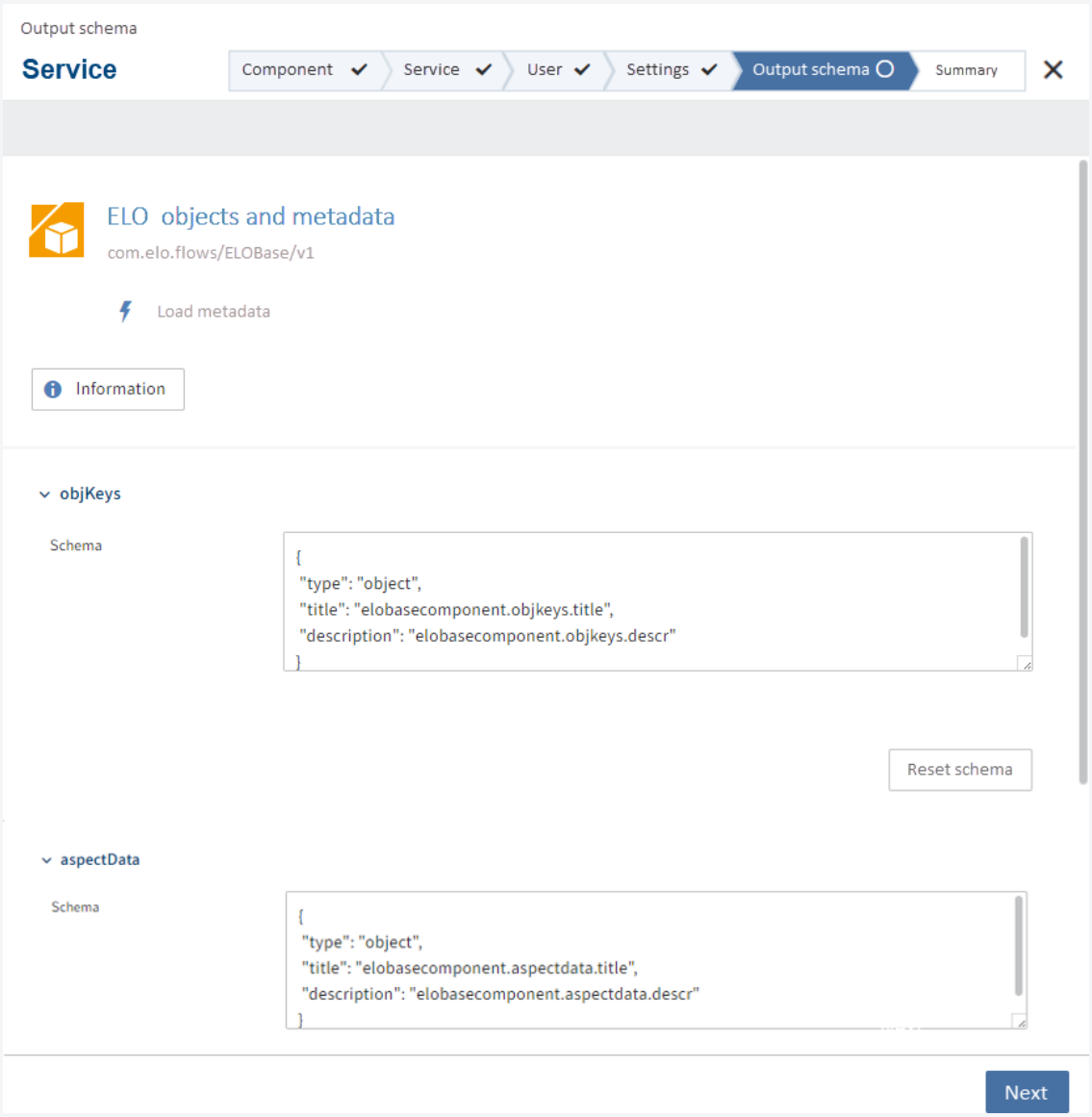
Some services offer you the output schema function. That means that this service offers a dynamic output object. This can be the case with metadata objects, for example.

Example: You file a document with the *E-mail* metadata form and want to access a field of this document in a subsequent node. As ELO Flows initially does not know which fields are relevant for this document, the service only offers the parent object (e.g. objKeys and aspectData) in the subsequent object without access to the individual metadata fields.



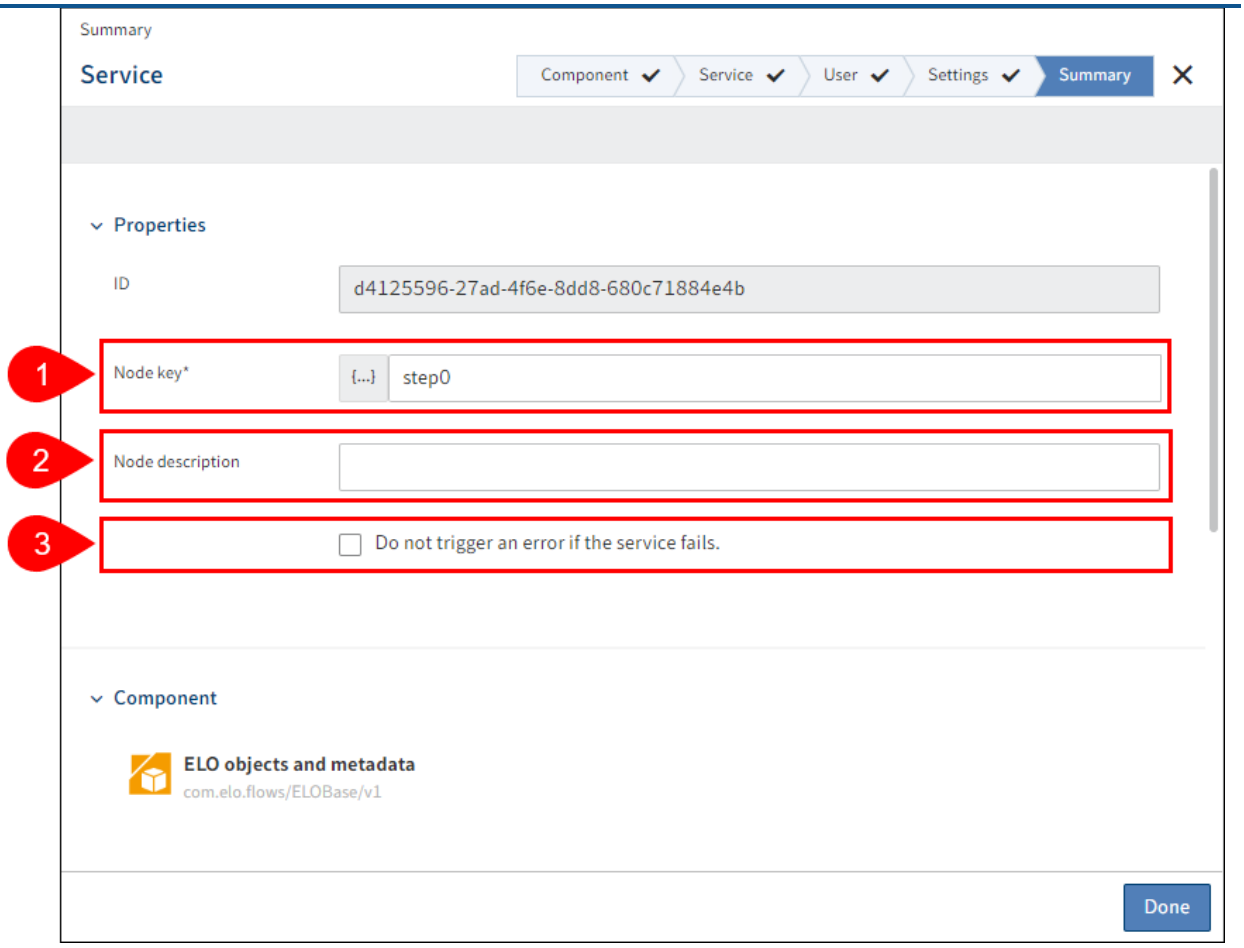
So that ELO Flows knows which metadata fields the output object has, you can either use a *standard metadata form* or load a special one. In the metadata form field, enter

E-mail and select *Load schema*. The schema is loaded with the appropriate metadata form properties.



Alternative: You can also select *Use default*. The object is then output with the default properties, and the properties of the object have to be accessed manually via JSONata.

If applicable, confirm the settings on the *Output schema* page by selecting *Next*.



The *Summary* tab opens. All the settings are shown once again.

1 Node key: Name of the node shown in square brackets in the flow designer

2 Node description: More detailed description of the node

3 Do not trigger an error if the service fails.: The entire flow runs, even if this service fails. This function is for test purposes. The result data for this service may be faulty or unavailable.

8. Select *Done* to apply the service settings in the configuration area.

> **Please note**
>
> With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.
>
> In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

9. Save the entire flow.

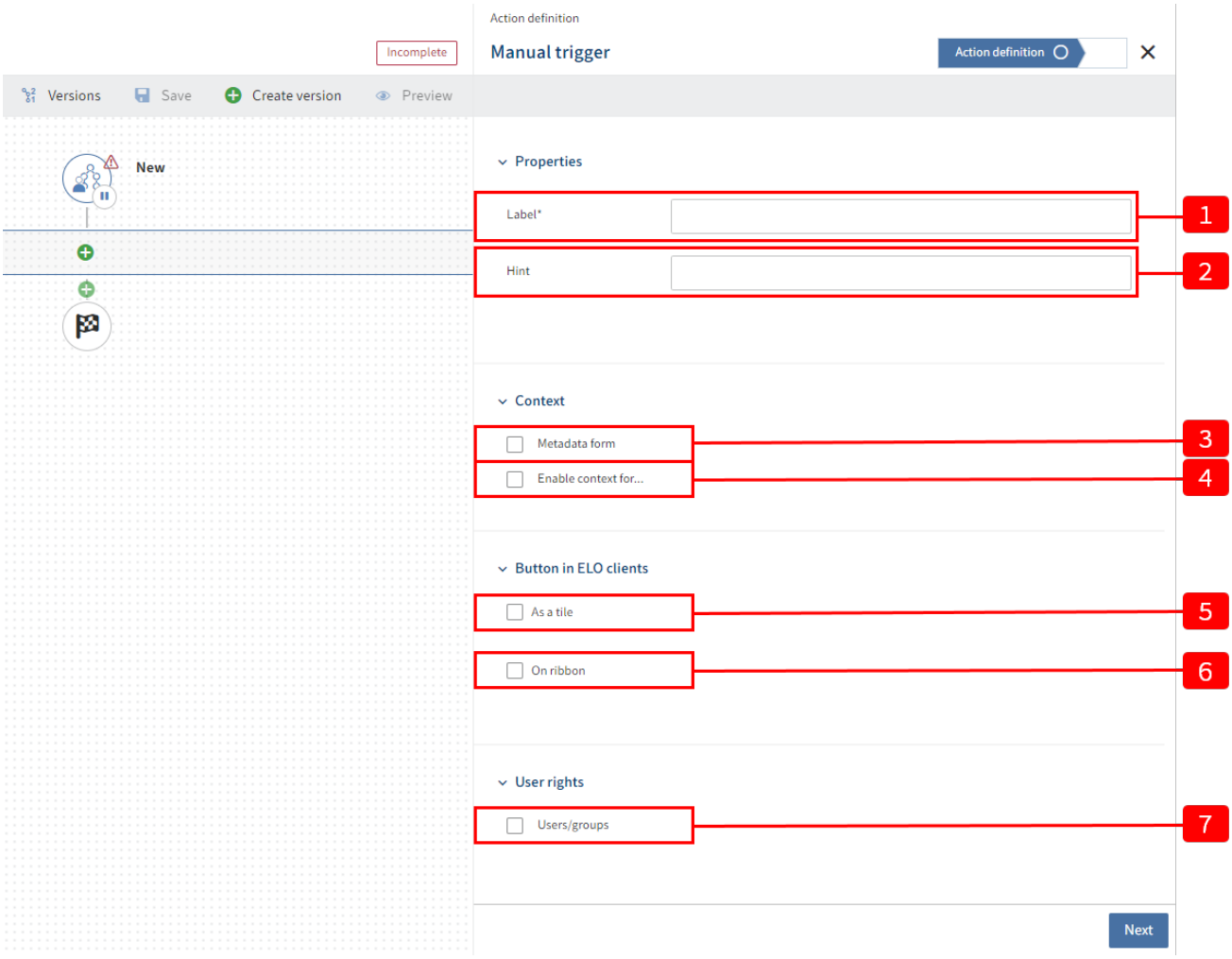
   The status display now changes to *Active*.

User actions

Flows in the user actions area consist of a manual trigger and of at least one or more services. Optionally, you can make flows more complex by using branches.

Select trigger

For user actions, you don't select a trigger for a component, but instead define a button for ELO clients. This button starts the flow.



1. Open the trigger in the node editor by selecting the trigger node.

   The *Action definition* tab opens.

   You have the following configuration options:

   1 Select label: Button title shown in the ELO clients

   2 Add tip: Tooltip text for the button you've created

3 Configure metadata form: Configure the relevant metadata form for this button. The button is only enabled if the selected element has the defined metadata form.

> **Information**
>
> Under Components > General information, you will find more information on completing fields with gen. 1 and gen. 2 metadata in the flow designer.

4 Enable context for…: This button is only enabled if the selected element is a folder and/or a document. After activation, the object ID of the object selected in ELO Flows is provided.

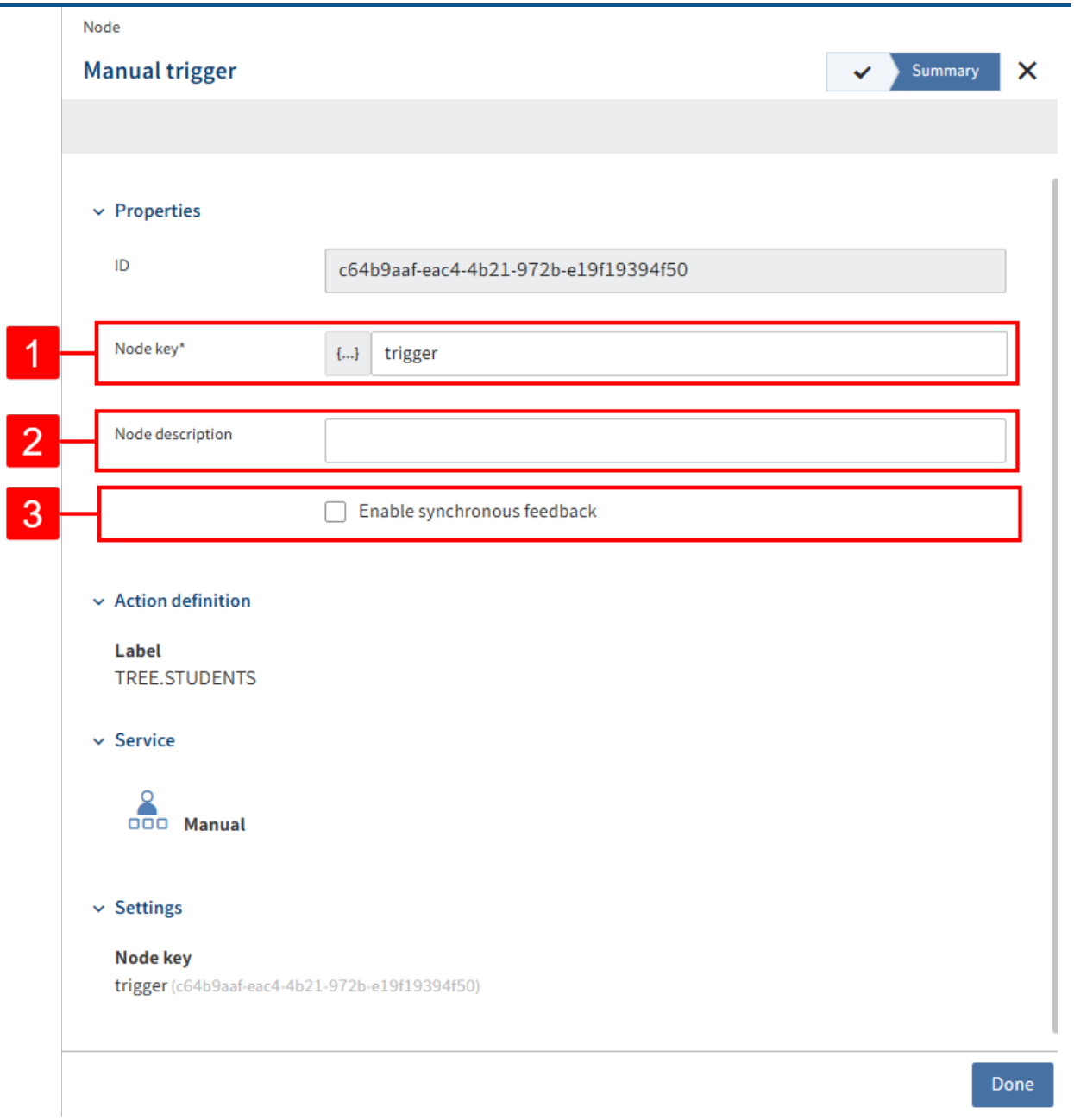
5 Show as a tile: The user action is shown as a tile in the *My ELO* work area. You can configure an icon for it.

6 Show on ribbon: Defines the position of the button on the ribbon in the ELO client. Select the *tab*, the *group*, and the *icon*.

7 Only show in certain ELO clients: Controls which clients the button is shown in. If nothing is selected here, the button is shown in all clients.

8 Users/groups: Defines which users and groups the button is shown to. If this option is not enabled, the button is shown to all users.

2. Complete at least the mandatory fields.
3. Select *Next* to confirm your entries.



The *Summary* tab opens. All the settings are shown once again.

1 Node key: Name of the node shown in square brackets in the flow designer

2 Node description: More detailed description of the node

3 Enable synchronous feedback: Executes the flow as a synchronous flow.

> **Information**
>
> You can find more information on this topic in the Synchronous flows section.

4. Select *Done* to finish configuring the trigger.

Please note

With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

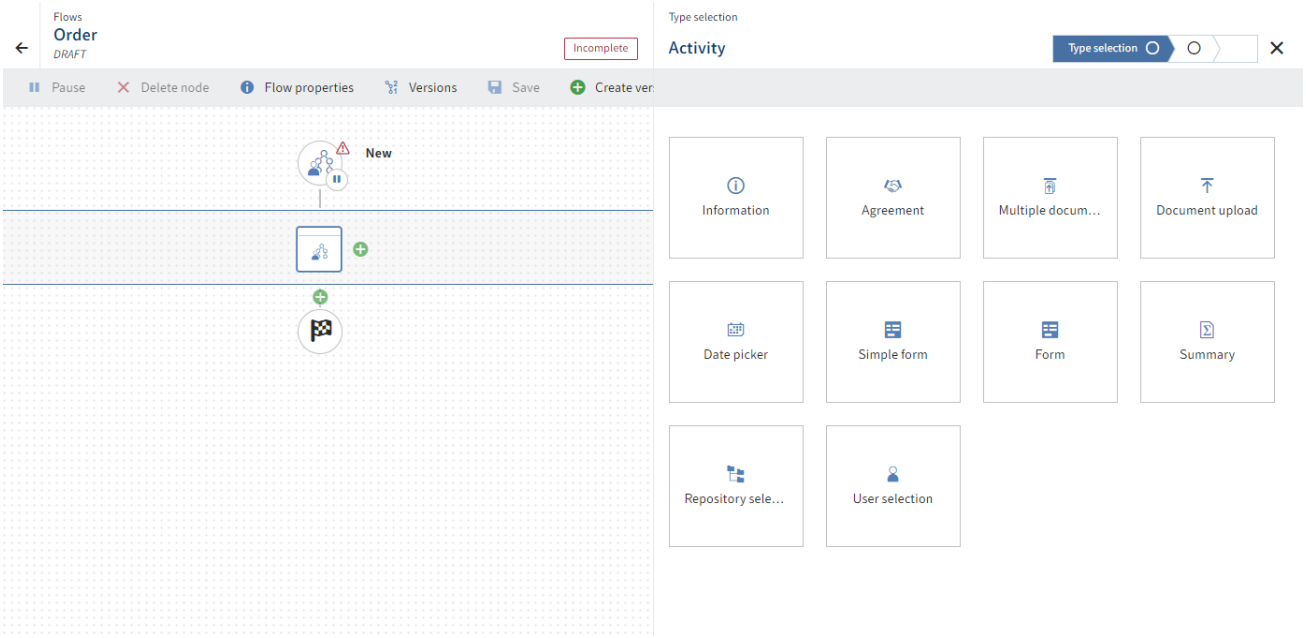
In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

Add activities

Via *Add activity*, define what to show when the button is selected in the client.

Information

Activities are optional.



1. To add an activity, select the plus icon next to the trigger node.

   All activities currently available are shown on the *Activity* pages.

   - Information: Shows a purely informational text. This activity does not provide any data.
   - Agreement: Here, you can define check boxes to reach an agreement with the users. For example, a data privacy agreement, license agreement, etc.
   - Multiple document upload: With this activity, you can prompt the user to upload multiple documents. You can save it in ELO with the ELO objects and metadata, for example.
   - Document upload: With this activity, you can prompt users to upload a document. You can save it in ELO with the ELO objects and metadata, for example.
   -

Date picker: Shows a calendar. Users can select a specific date or date interval here.

- Simple form: Shows a form. For example, you can implement a contact form with this activity.
- Summary: The summary shows all data from previous activities. You can place this at the end to give users a final overview.
- Repository selection: With the *Repository selection* activity, you can prompt users to select an element from the ELO repository. In the settings, set the *root element* to define the level at which users are offered a selection.
- User selection: With this activity, you can prompt users to select an ELO user and/or ELO group from the ELO system.

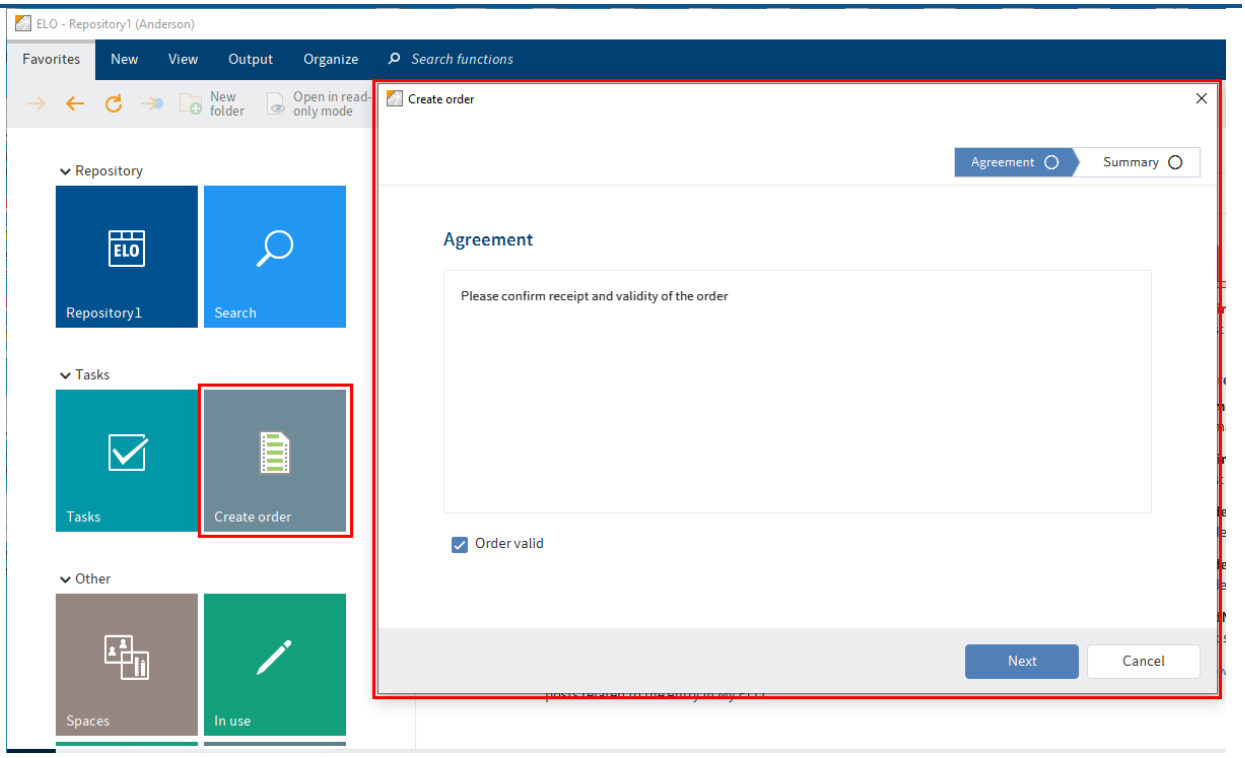
> **Information**
>
> You can select localization keys for activities with text or titles visible to the user in the client.
>
> If you configure flows outside of ELO packages, you can only use keys available in the system.
>
> The keys can come from the following sources:
>
> - Keys already configured via ELO packages
> - Keys filed in the repository as properties files
>
> For flows you edit within ELO packages, you can also create new localization keys and configure existing ones. The localization is configured for the active language of the ELO Administration Console.
>
> 1. Enter a localization key.
>
>   Alternative: Select an existing localization key from the drop-down menu.
>
> 2. Enter a name in the *Localization* field.
> 3. Select the floppy disk icon to assign the name to the localization and save it.

2. Select an activity.

   Option 1: To map a series of activities, these can be chained. Add another activity.

   Option 2: Delete the existing activity by selecting *Delete node* on the ribbon.

   The button and activities in the ELO client can be shown as follows:

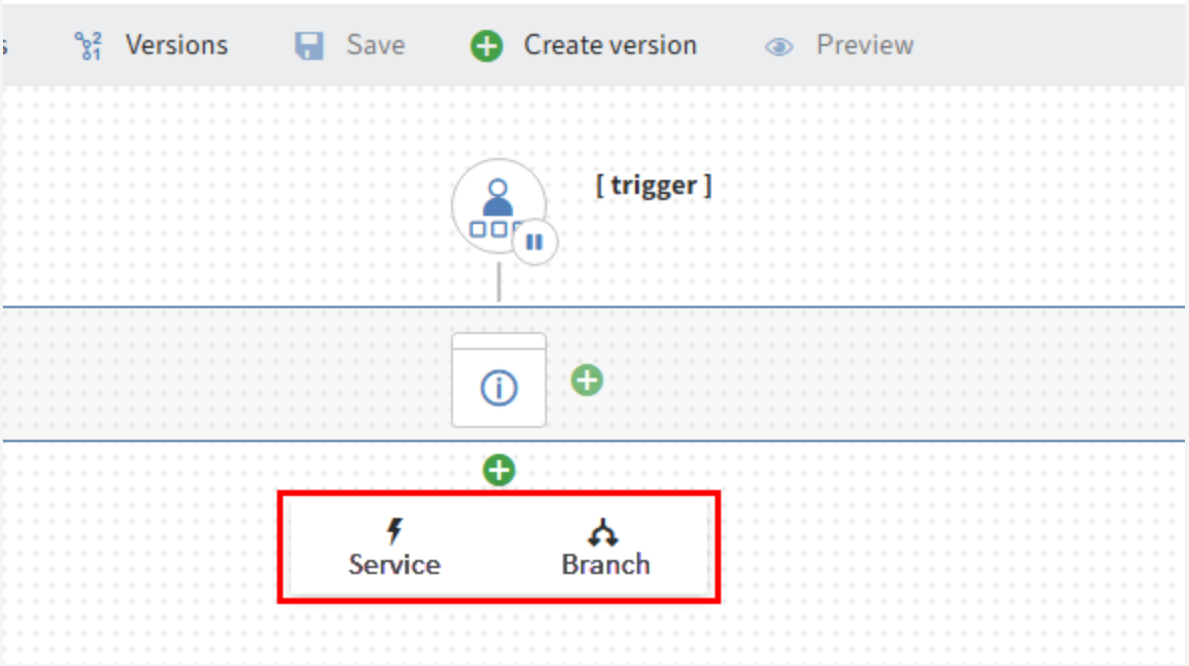


Please note

The button is only shown if the status of the flow is *Active*. You have to define at least one service.

Once you have defined activities, you can create your first branch or service.

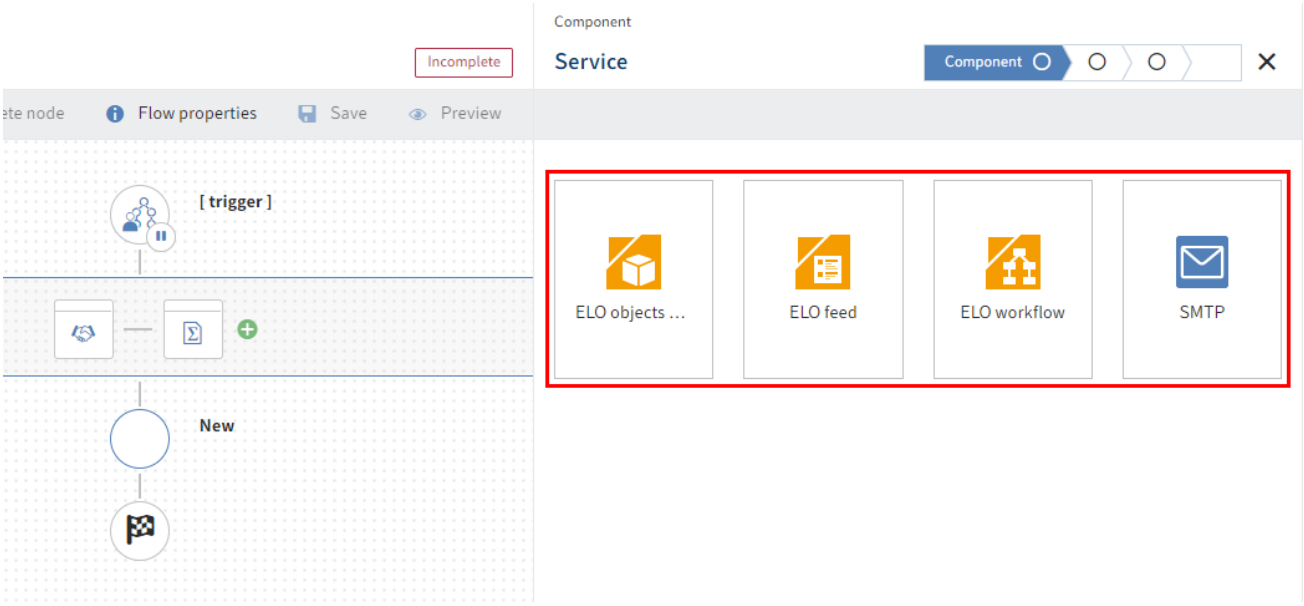
Information



Before each step, you can decide whether to create a branch with a true-false condition or to add a service. You can nest branches in other branches or continue with services.

You will find more information on branches in ELO Flows in this documentation under Advanced flow.

Add service



1. To add a service, select the plus icon next to the trigger node and select *Service*.

   The configuration area with the available components opens for the new node.

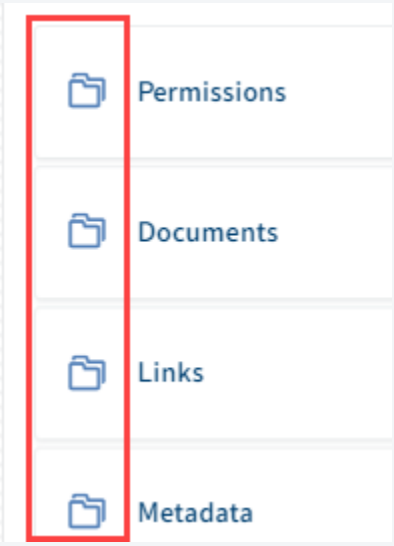
2.

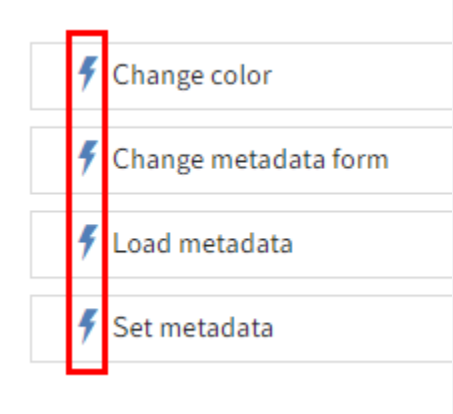
Select a component.

By selecting the component, you are automatically taken to the next tab with the services available for the component.

3. Select the desired service.

Information

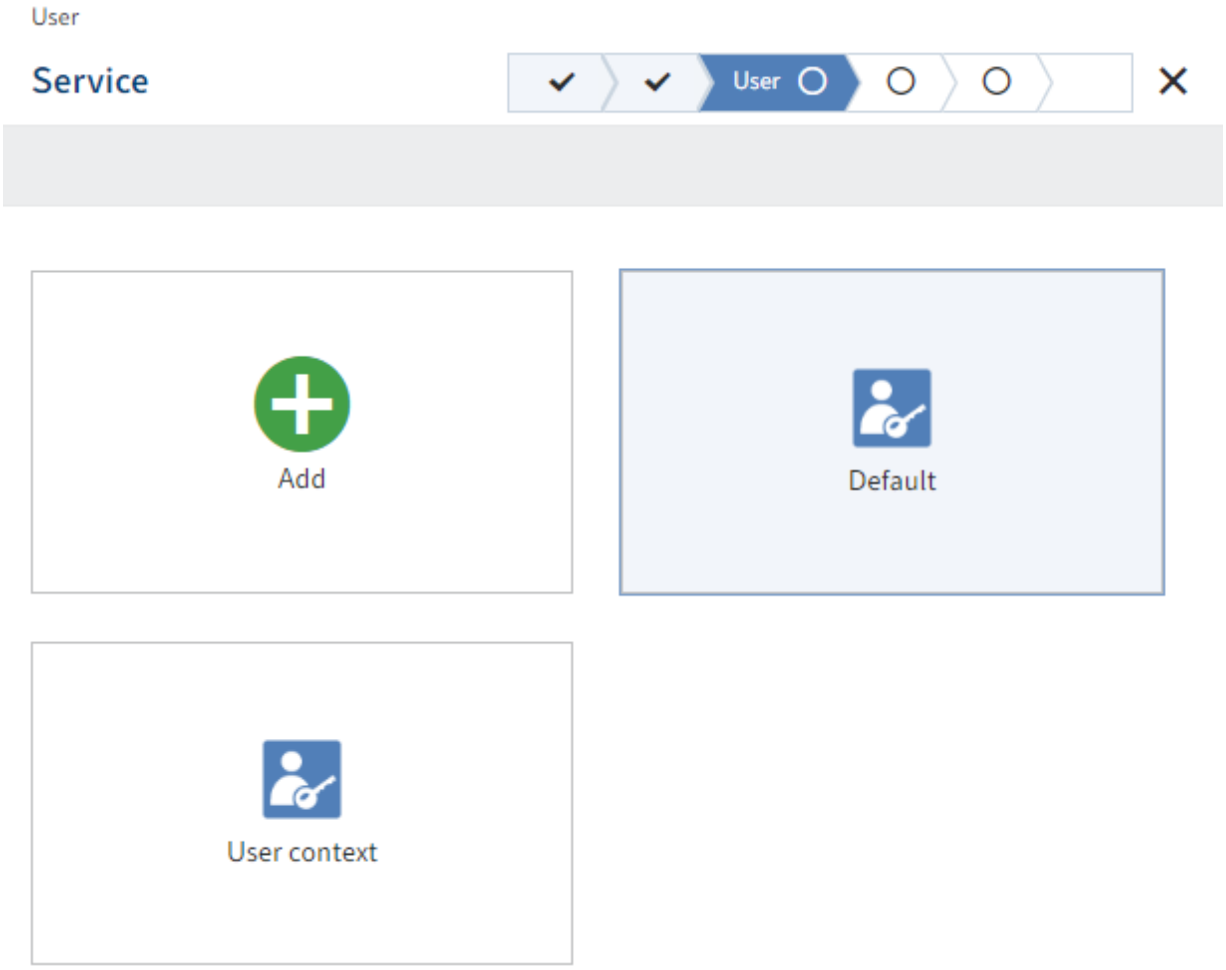




Services can be listed individually (lightning icon) or in groups (folder icon).

Selecting the service automatically takes you to the next tab with the user settings.

4. Select a user connection.

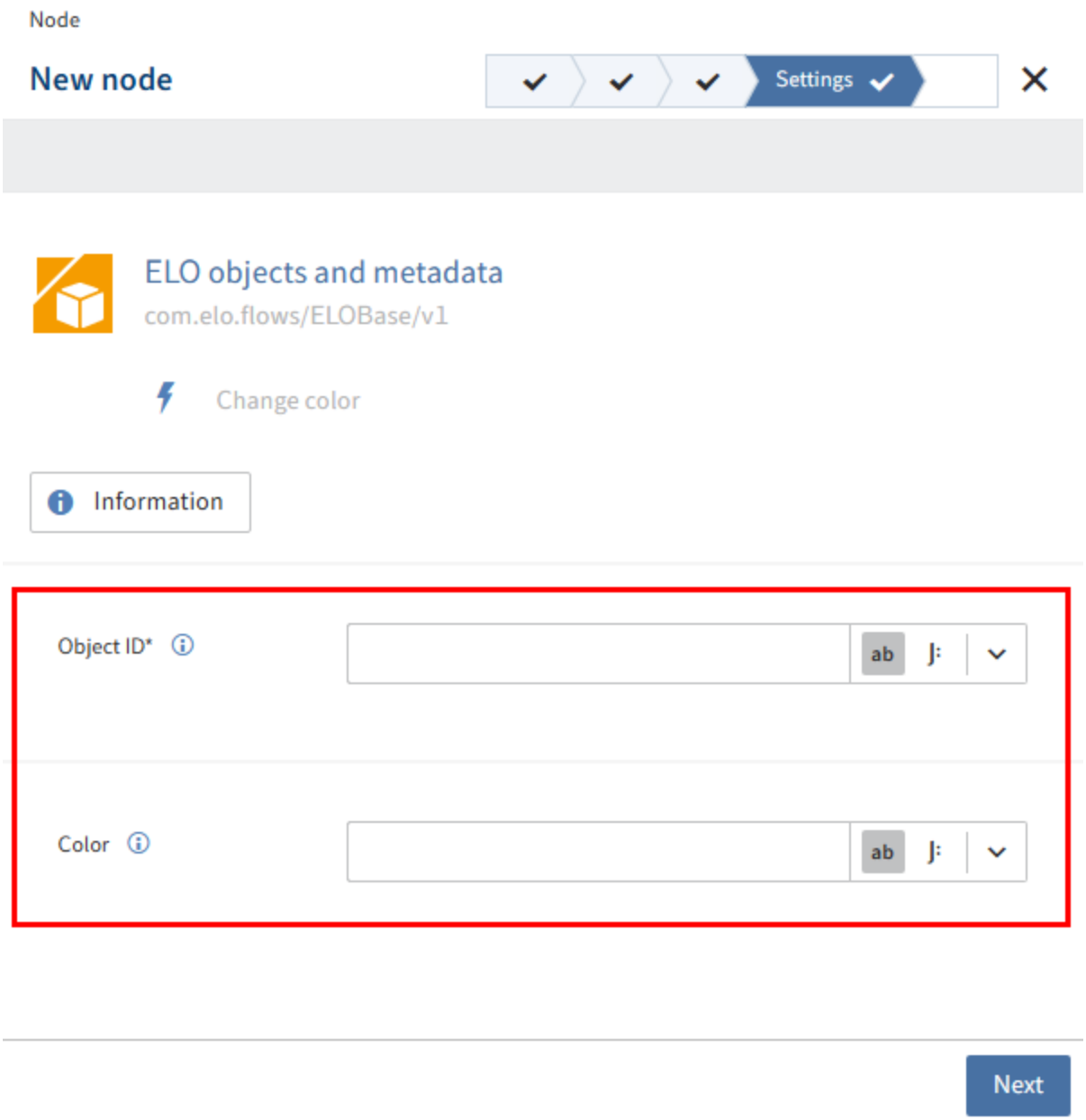


Information

If you've already configured a user under Credentials, you can select the user here. Otherwise, you can select the default connection or add a new one.

5. Confirm your selection with *Next*.

   The *Settings* tab appears.



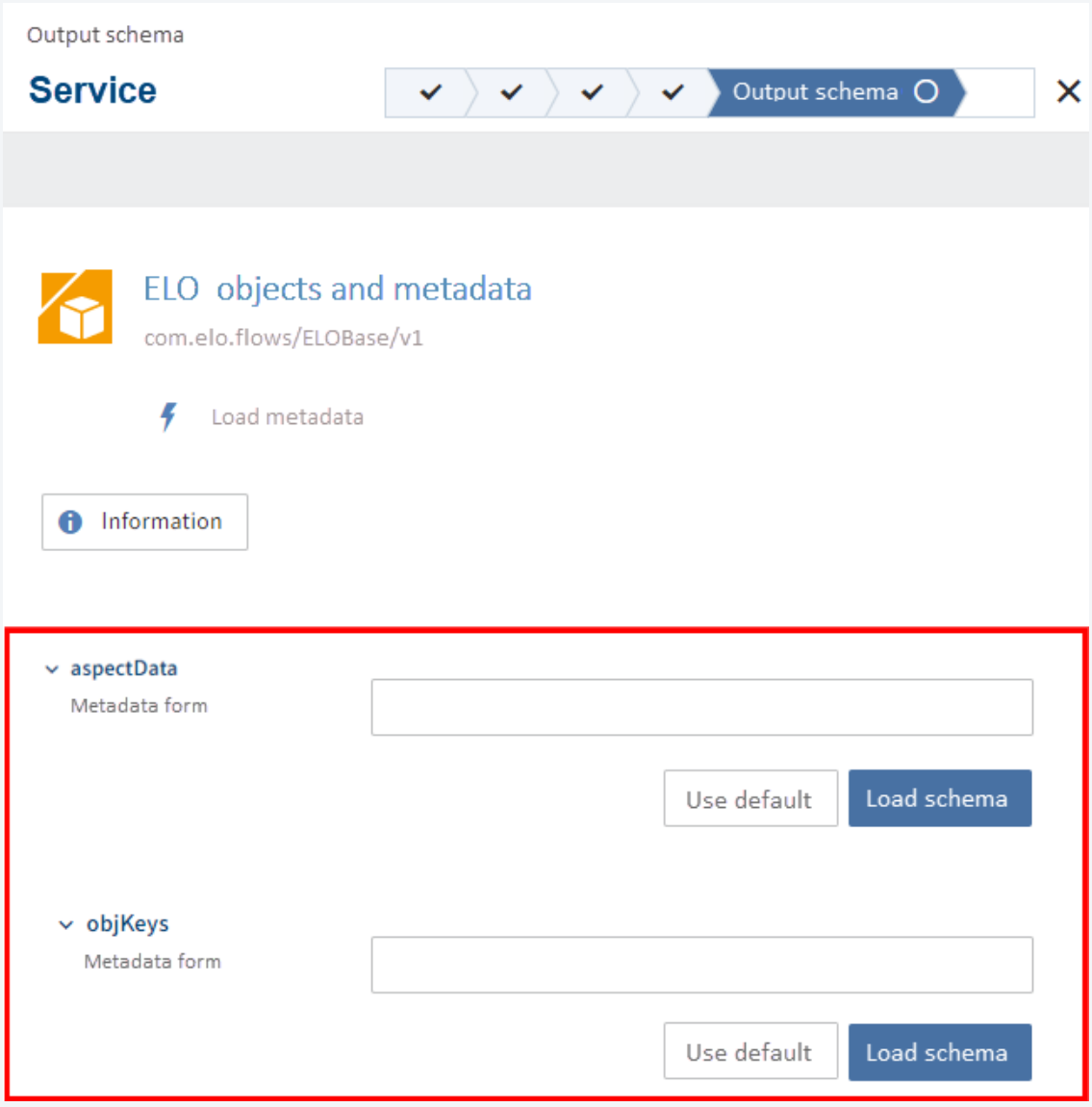
6. Configure custom data for the service.

> **Information**
>
> You can complete the fields with *static* or *dynamic values*. Dynamic evaluation is performed based on *JSONata*. With this transformation language, you can access event objects of the previous nodes and triggers in addition to using transformations or simple functions.
>
> You will find more about the use of JSONata in ELO Flows under JSONata editor.

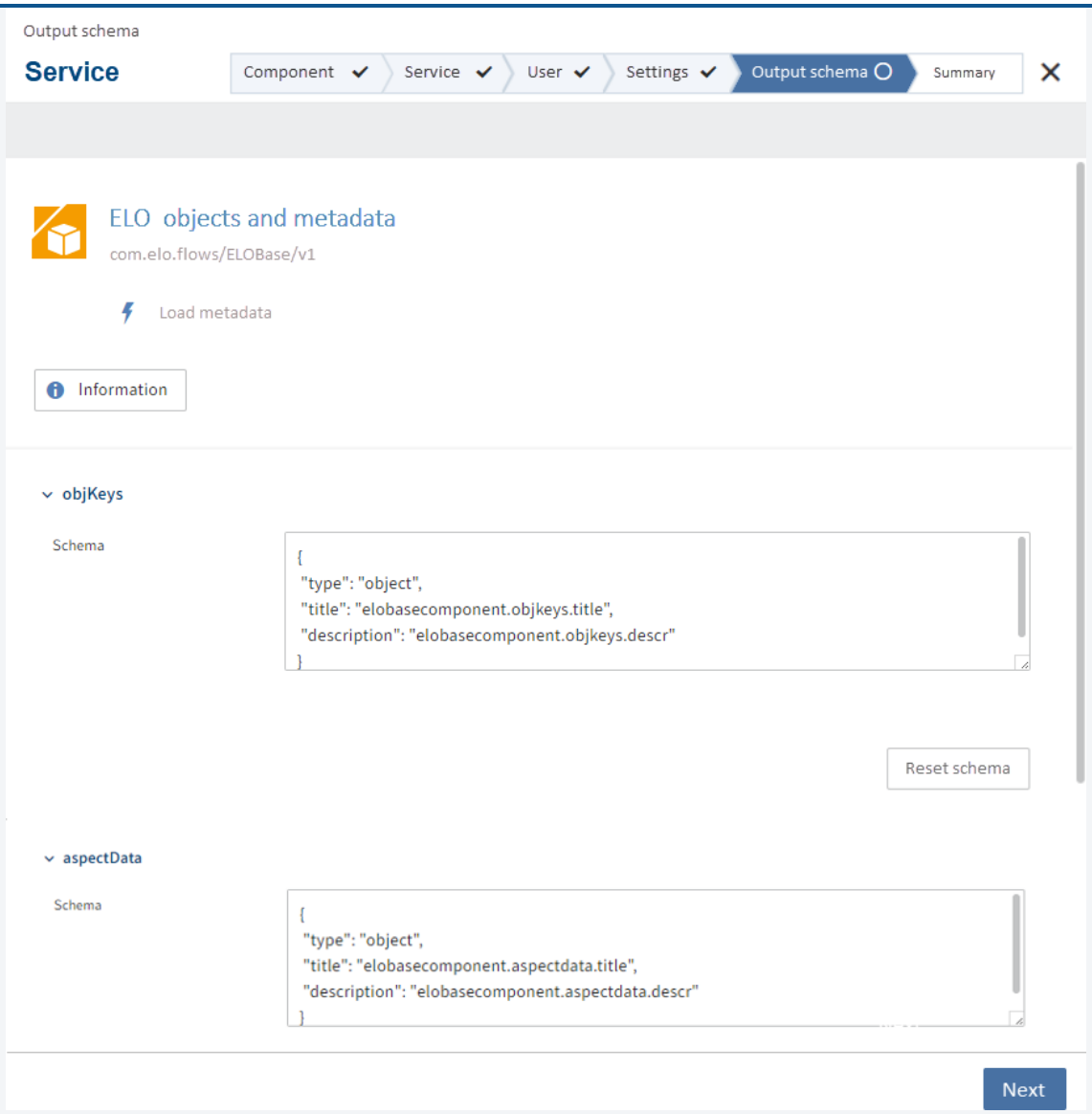
> **Information**

Some services offer you the output schema function. That means that this service offers a dynamic output object. This can be the case with metadata objects, for example.

Example: You file a document with the *E-mail* metadata form and want to access a field of this document in a subsequent node. As ELO Flows initially does not know which fields are relevant for this document, the service only offers the parent object (e.g. objKeys and aspectData) in the subsequent object without access to the individual metadata fields.

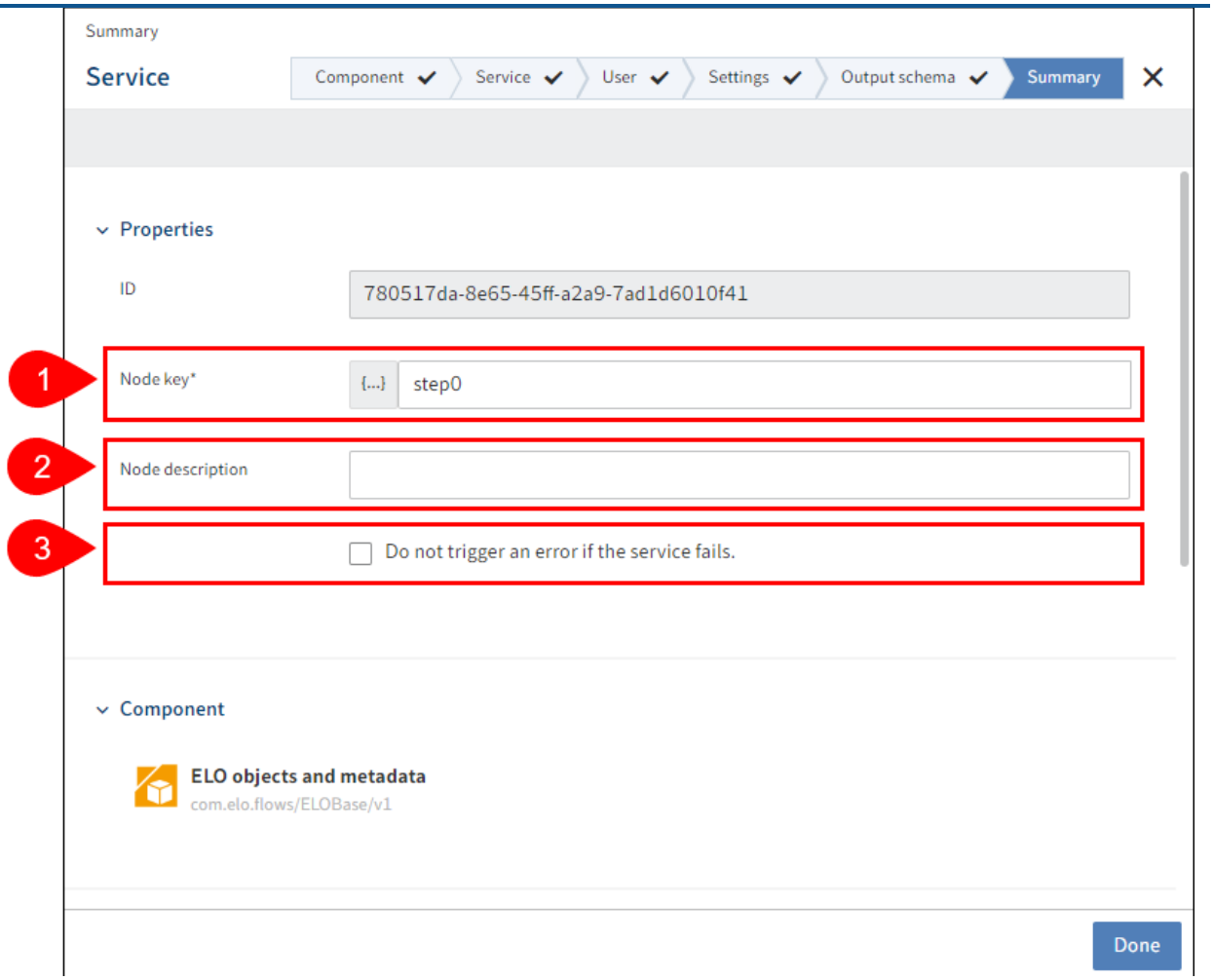


So that ELO Flows knows which metadata fields the output object has, you can either use a *standard metadata form* or load a special one. In the metadata form field, enter *E-mail* and select *Load schema*. The schema is loaded with the appropriate metadata form properties.



Alternative: You can also select *Use default*. The object is then output with the default properties, and the properties of the object have to be accessed manually via JSONata.

If applicable, confirm the settings on the *Output schema* page by selecting *Next*.



The *Summary* tab opens. All the settings are shown once again.

1 Node key: Name of the node shown in square brackets in the flow designer

2 Node description: More detailed description of the node

3 Do not trigger an error if the service fails.: The entire flow runs, even if this service fails. This function is for test purposes. The result data for this service may be faulty or unavailable.

7. Select *Done* to apply the service settings in the configuration area.

> **Please note**
>
> With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.
>
> In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

8. Save the entire flow.

The status display now changes to *Active*.

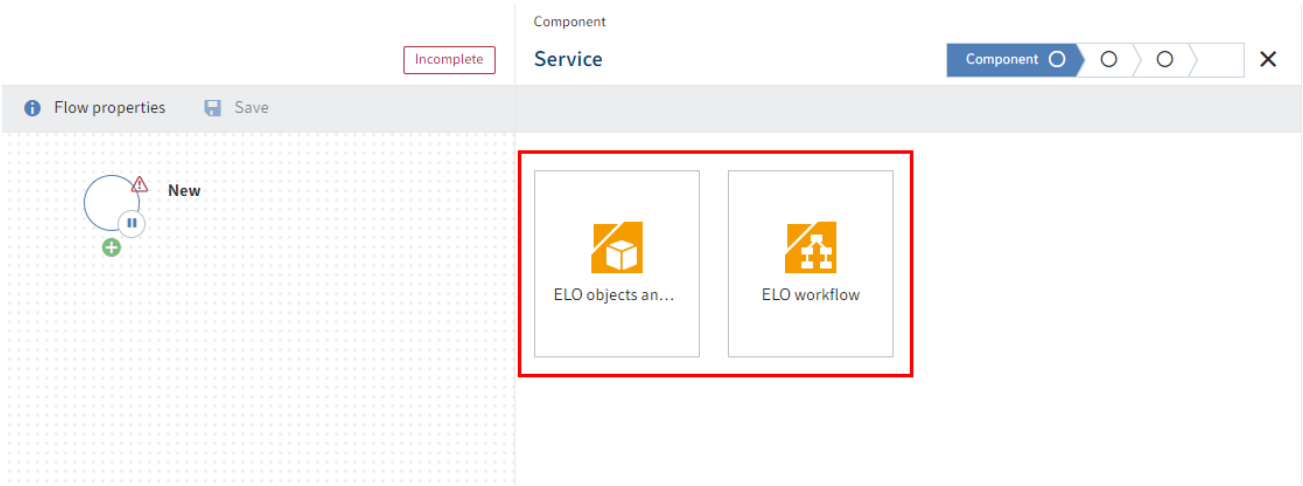
Data queries

Data queries are special flows for providing data. They run synchronously and enable you to provide data queries with the corresponding components. One application is dynamic keyword lists, for example.

If necessary, you can implement components yourself and thus provide data queries. Check the 'ELO Flows component development' documentation.

Select trigger

First, select a trigger. Proceed as follows:

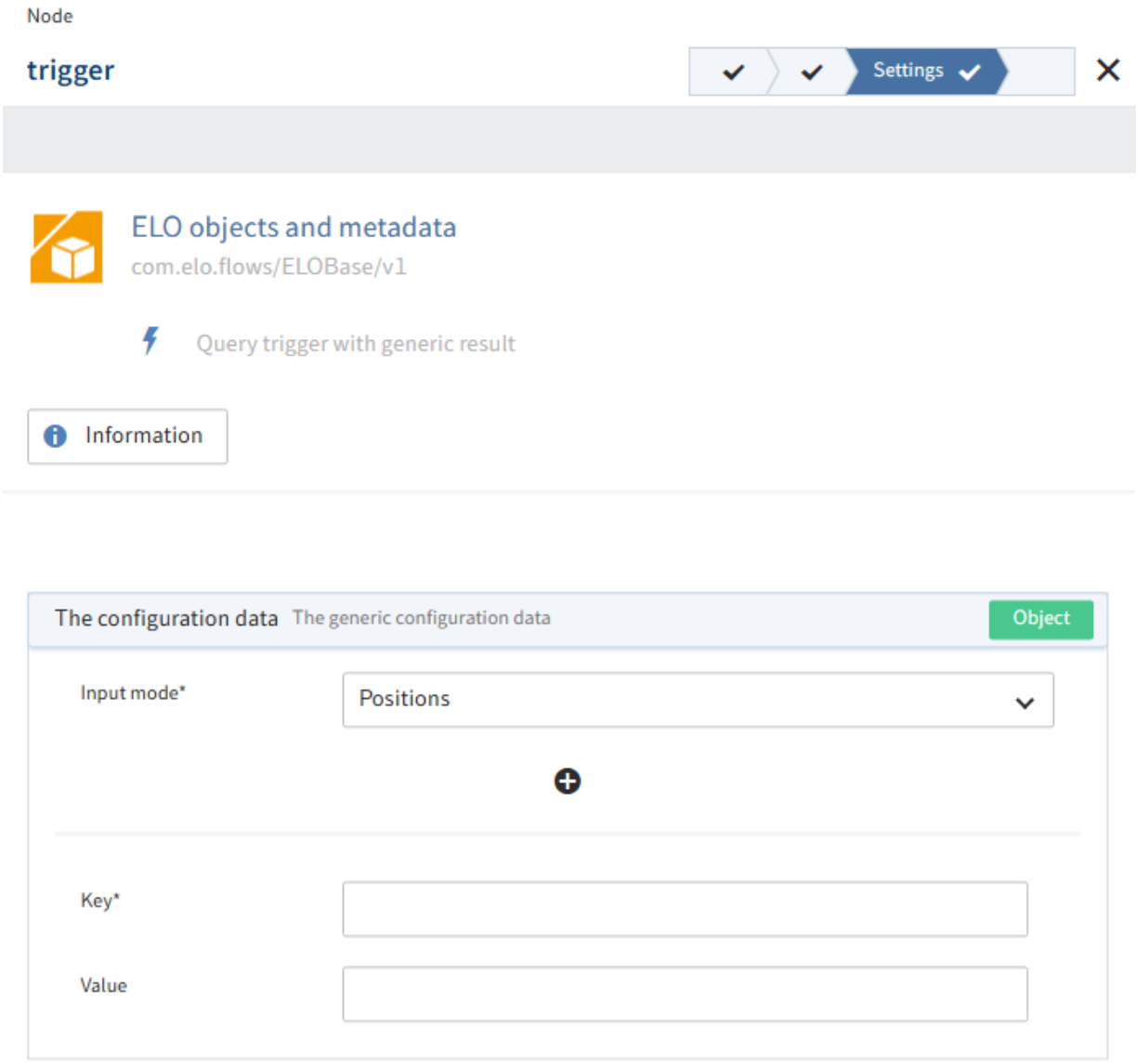


1. Select a component whose trigger you want to use.

   The flow designer automatically jumps to the following *Trigger* tab.

2. Select the trigger you want to start this flow.

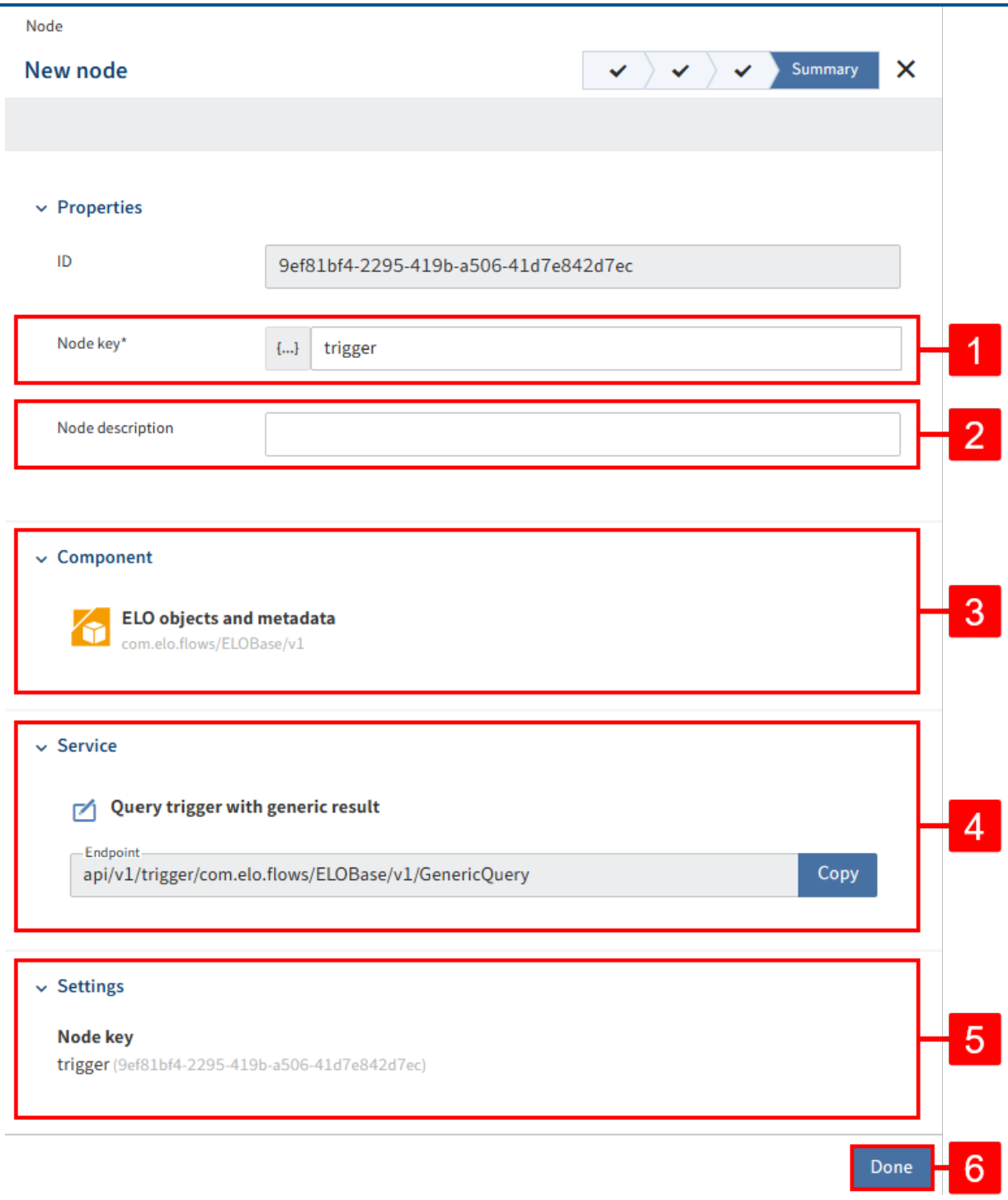
   The flow designer automatically jumps to the following *Settings* tab. The setting options for the trigger are shown here:



In this example, you can change the configuration data for the *Query trigger with generic result* trigger.

> **Information**
>
> The section Components > ELO metadata and objects > Triggers goes into more detail on the trigger settings for the *ELO metadata and objects* component.

3. Select *Next* to open the *Summary* tab.



This tab shows a summary of all settings:

1 Node key: Name of the node shown in square brackets in the flow designer.

2 Node description: More detailed description of the node.

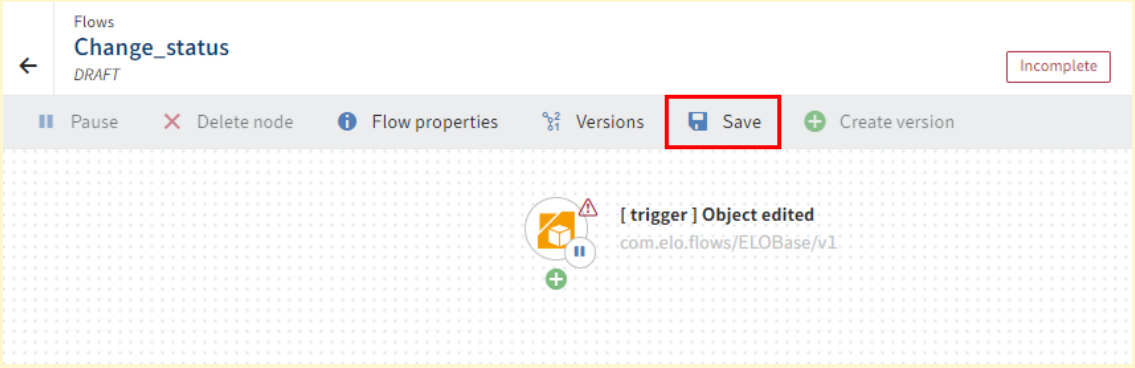
3 Component: Displays the selected component

4 Service: Displays the selected trigger with a copyable path to the endpoint

5 Settings: Displays the node key

4. Select *Done* to finish configuring the trigger.

Please note



With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

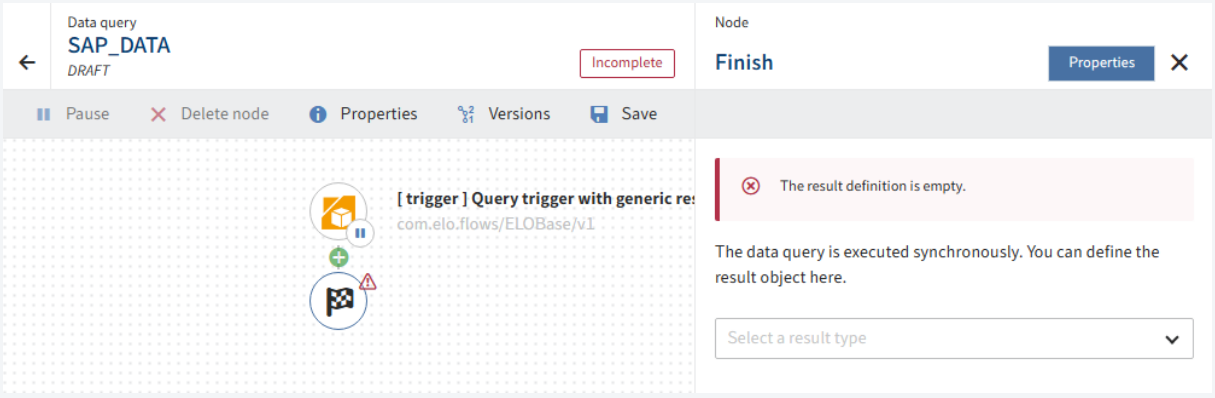
In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

5. Save the entire flow.

Information

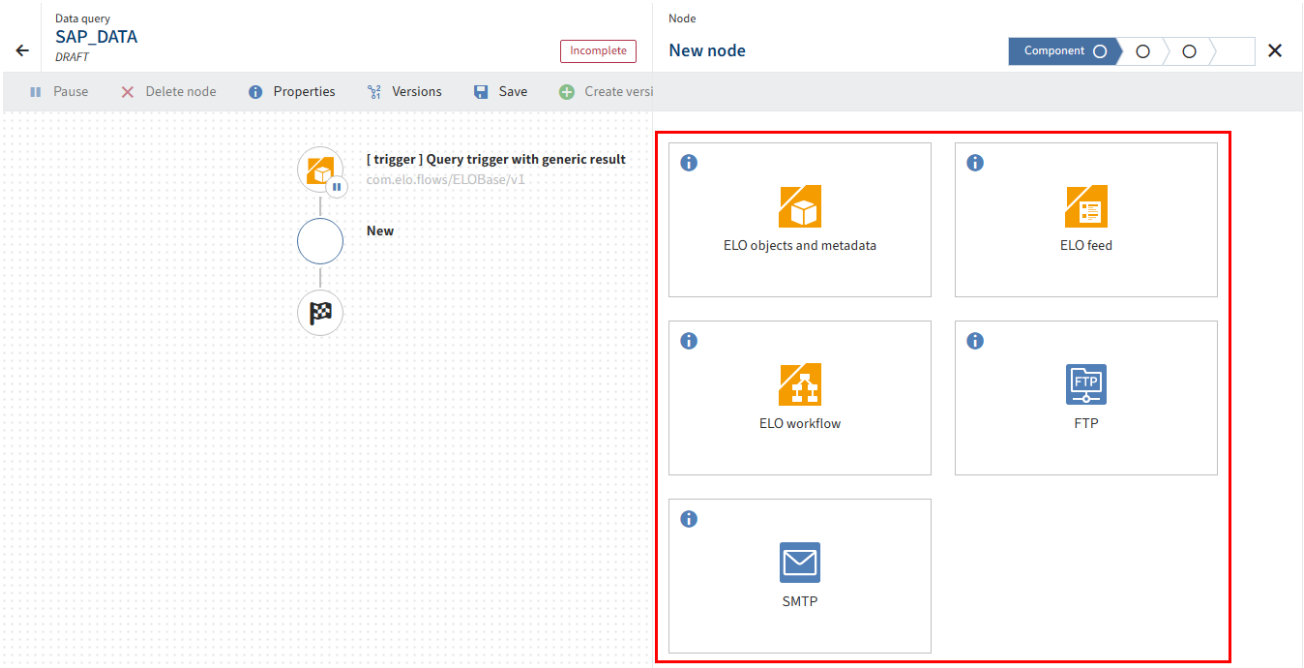
With synchronous flows, saving changes the status of the flow to *Active*.

Synchronous flows require a result definition in the end node to switch from *Incomplete* to *Active*.

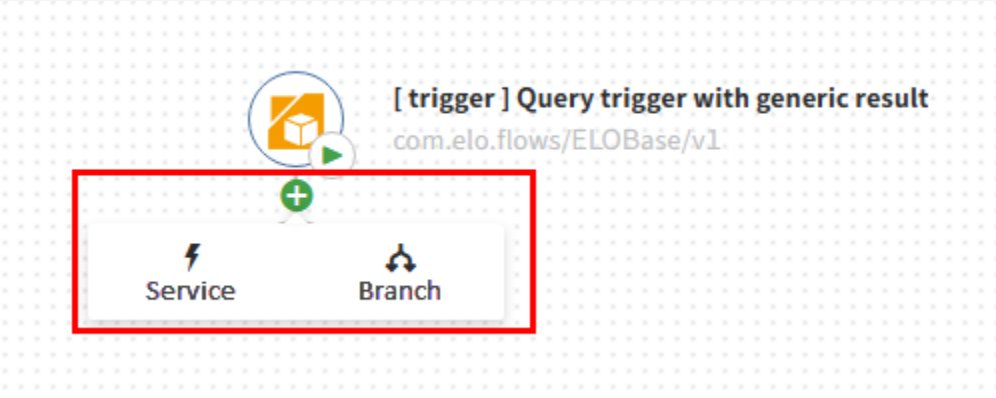


You will find more information about synchronous flows under Synchronous flows.

Add service



Information



Before each step, you can decide whether to create a branch with a true-false condition or to add a service. You can nest branches in other branches or continue with services.

You will find more information on branches in ELO Flows in this documentation under Advanced flow.

1. To add a service, select the plus icon next to the trigger node and select *Service*.

   The configuration area with the available components opens for the new node.

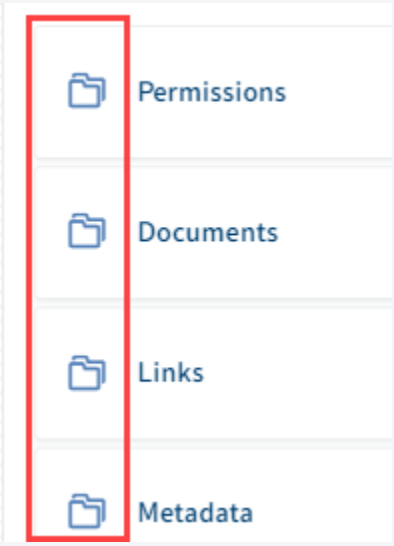
2. Select a component.

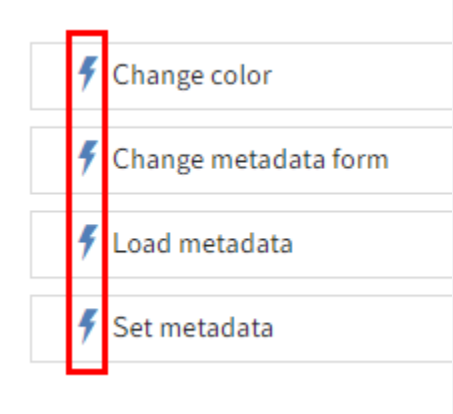
   By selecting the component, you are automatically taken to the next tab with the services available for the component.

3.

Select the desired service.

Information

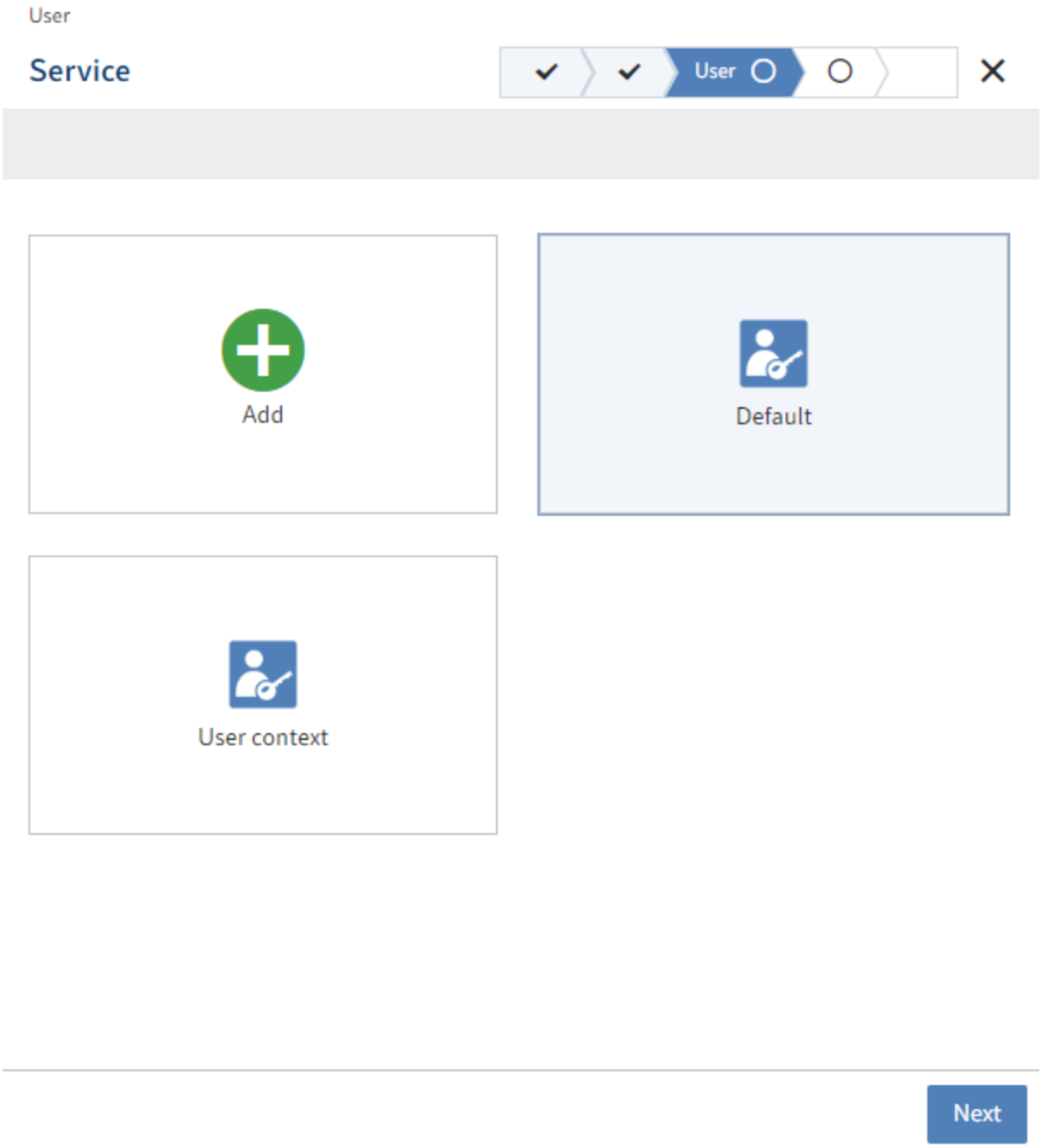




Services can be listed individually (lightning icon) or in groups (folder icon).

Selecting the service automatically takes you to the next tab with the user settings.

4. Select a user connection.

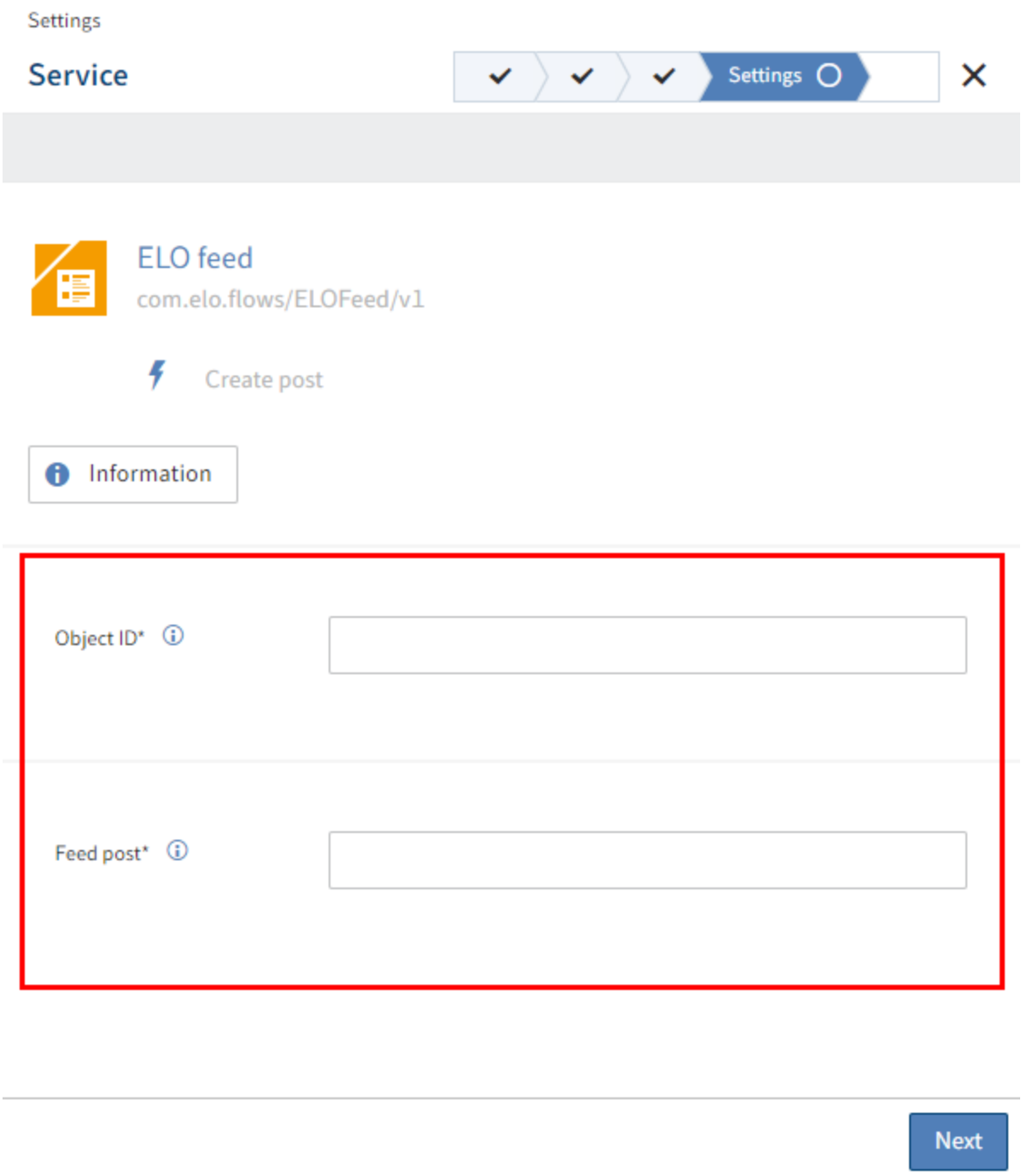


Information

If you've already configured a user under Credentials, you can select the user here. Otherwise, you can select the default connection or add a new one.

5. Confirm your selection with *Next*.

   The *Settings* tab appears.



6. Configure custom data for the service.

Information

You can complete the fields with *static* or *dynamic values*. Dynamic evaluation is performed based on *JSONata*. With this transformation language, you can access event objects of the previous nodes and triggers in addition to using transformations or simple functions.

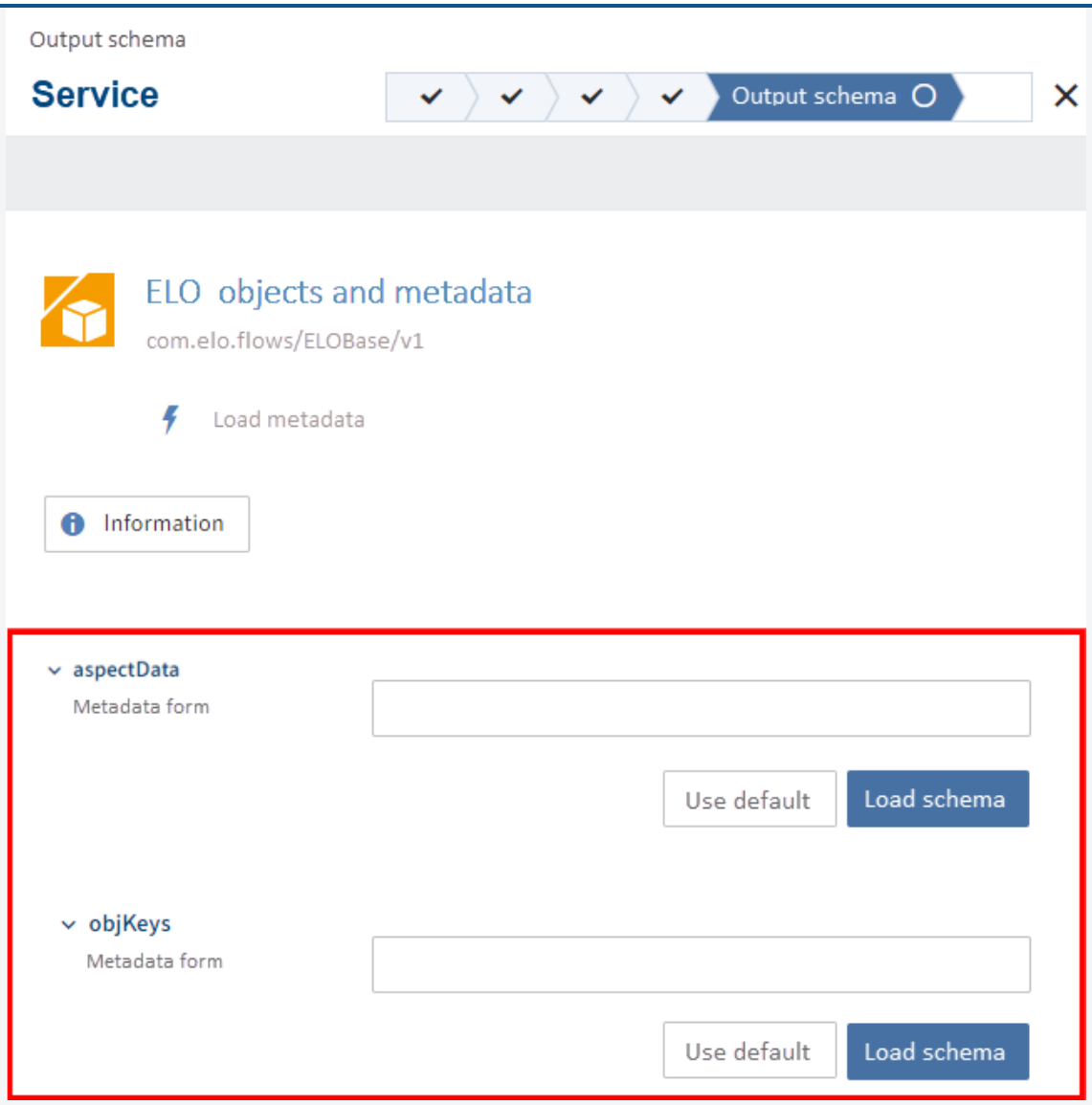
You will find more about the use of JSONata in ELO Flows under JSONata editor.

7. Confirm your selection with *Next*.

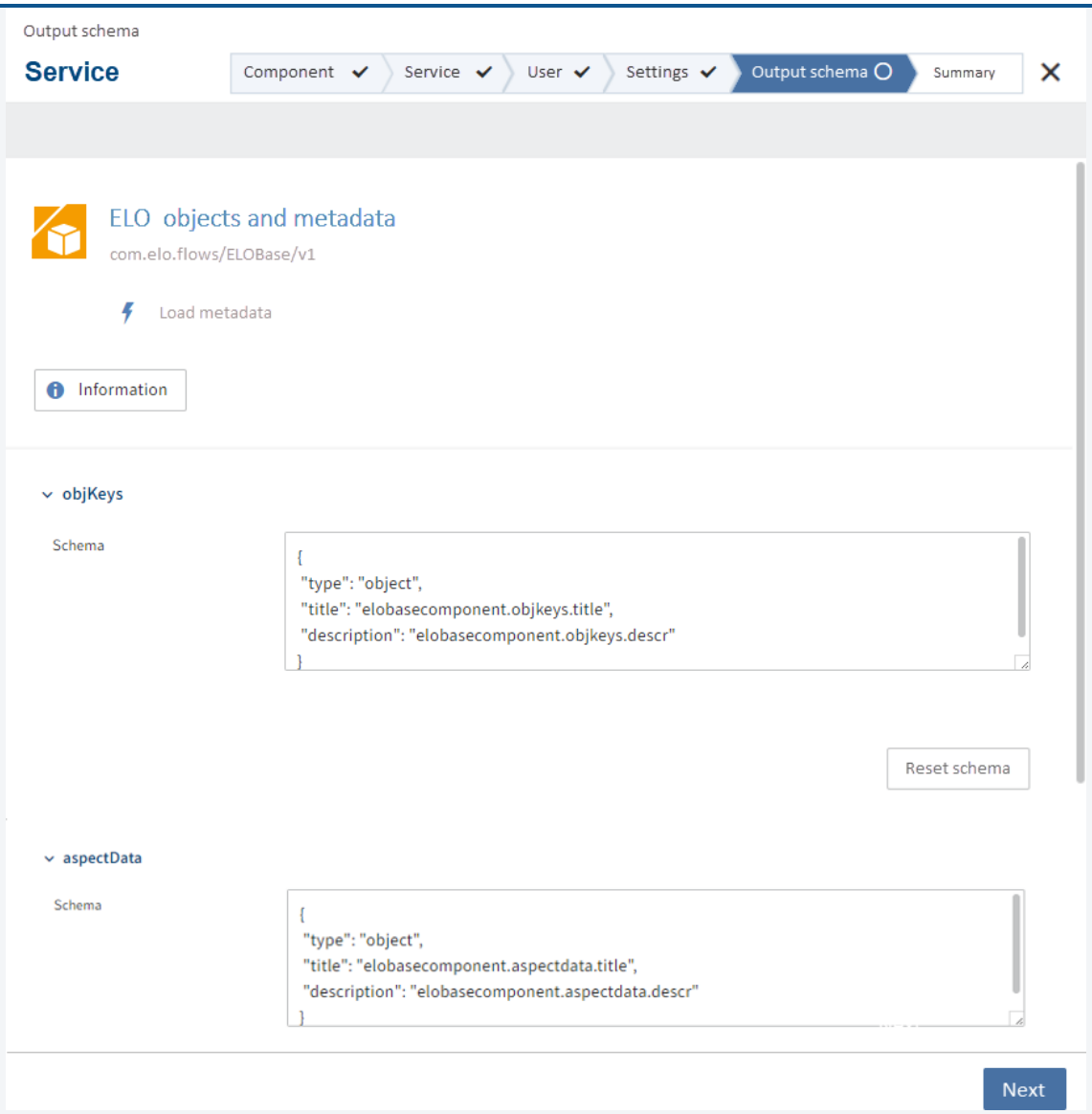
Information

Some services offer you the output schema function. That means that this service offers a dynamic output object. This can be the case with metadata objects, for example.

Example: You file a document with the *E-mail* metadata form and want to access a field of this document in a subsequent node. As ELO Flows initially does not know which fields are relevant for this document, the service only offers the parent object (e.g. objKeys and aspectData) in the subsequent object without access to the individual metadata fields.

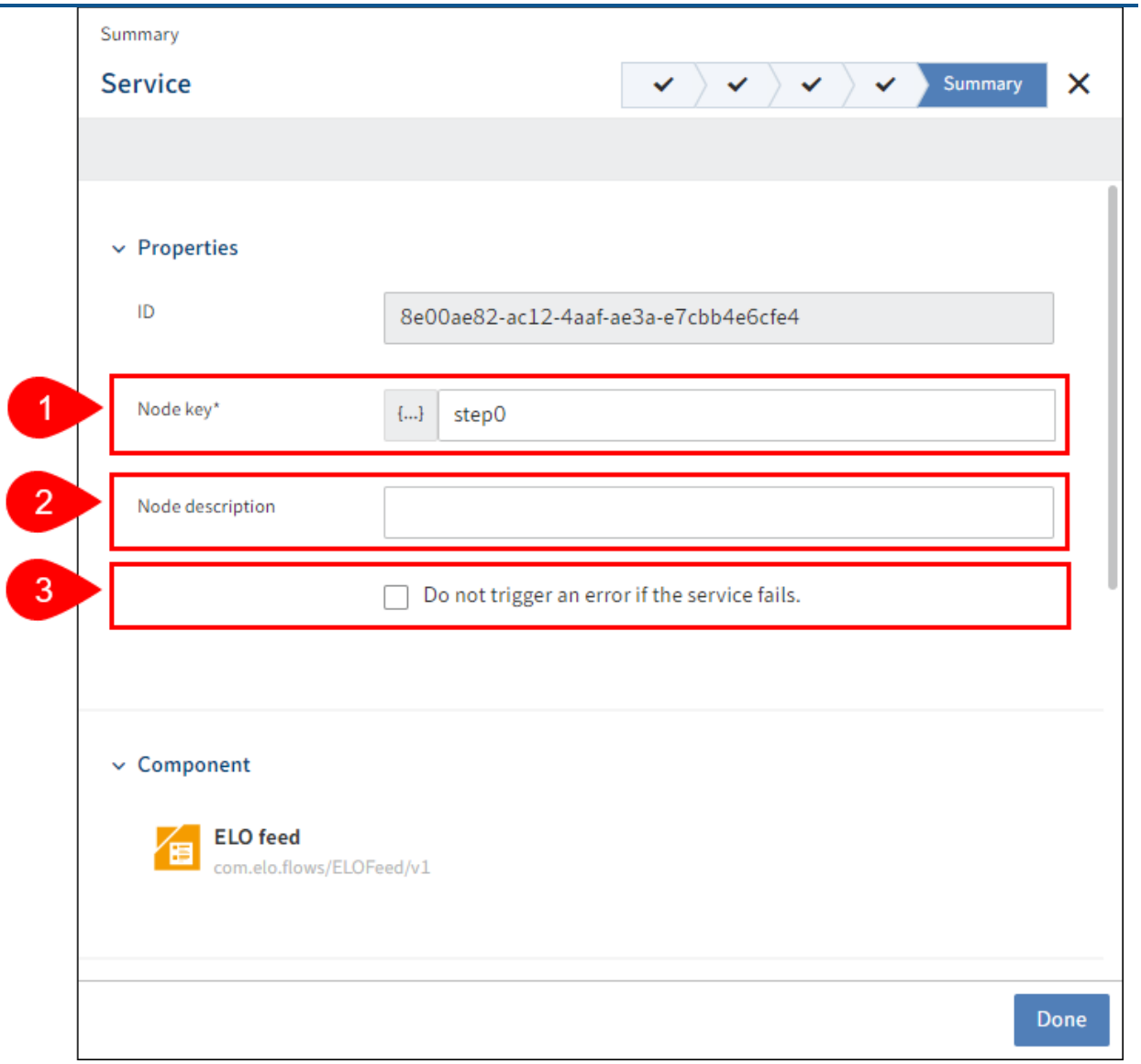


So that ELO Flows knows which metadata fields the output object has, you can either use a *standard metadata form* or load a special one. In the metadata form field, enter *E-mail* and select *Load schema*. The schema is loaded with the appropriate metadata form properties.



Alternative: You can also select *Use default*. The object is then output with the default properties, and the properties of the object have to be accessed manually via JSONata.

If applicable, confirm the settings on the *Output schema* page by selecting *Next*.



The *Summary* tab opens. All the settings are shown once again.

1 Node key: Name of the node shown in square brackets in the flow designer

2 Node description: More detailed description of the node

3 Do not trigger an error if the service fails.: The entire flow runs, even if this service fails. This function is for test purposes. The result data for this service may be faulty or unavailable.

8. Select *Done* to apply the service settings in the configuration area.

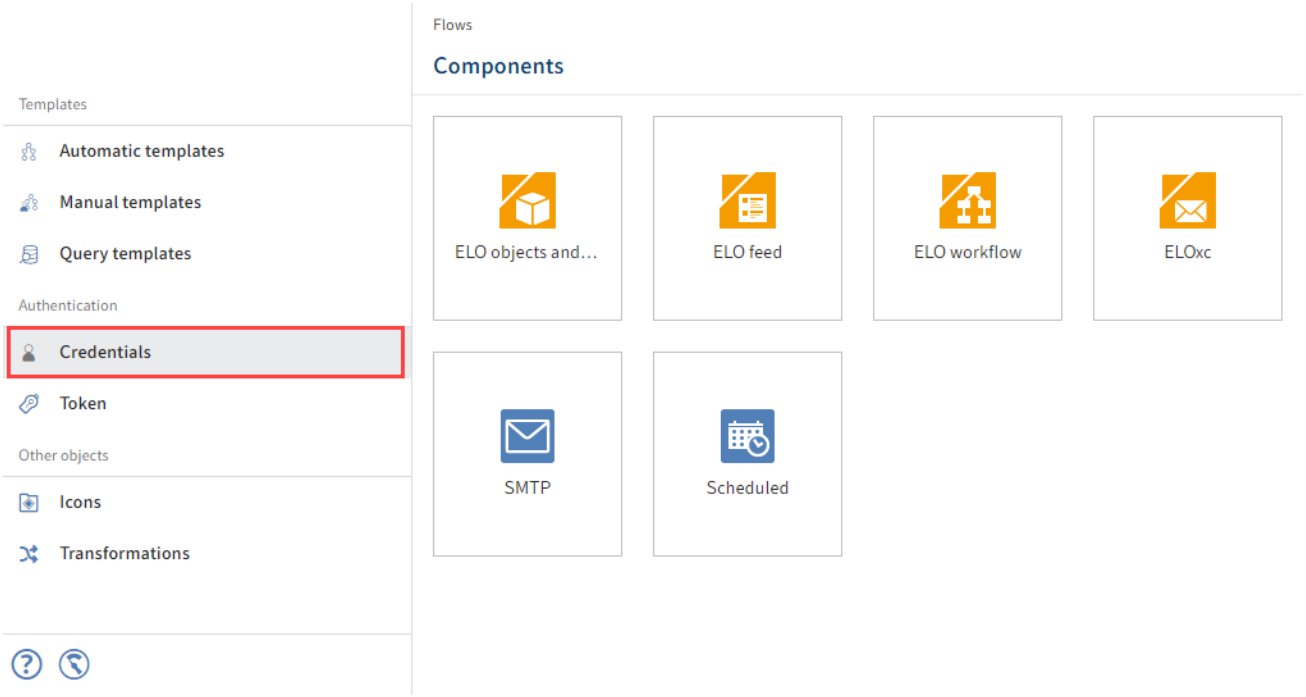
> **Please note**
>
> With the *Done* action, the configurations are only saved in the configuration area and are not yet applied to the flow.

> In the flow designer, select *Save* to apply the entries in the configuration area to the flow.

9. Save the entire flow.

   The status display now changes to *Active*.

Credentials

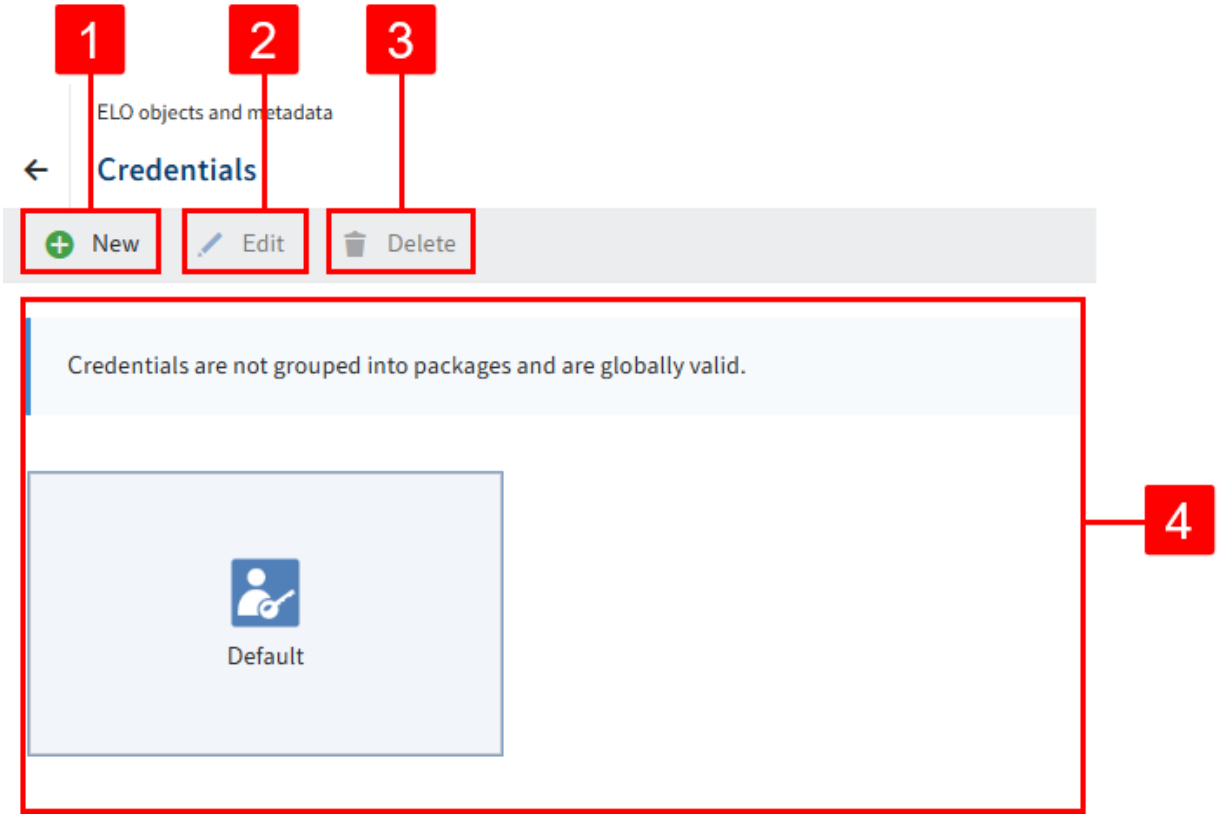


Under *Credentials*, you can configure credentials for each component.

Example applications include integrating accounts from third-party systems or configuring specific roles as users, for example invoice workflow users.

> **Information**
>
> Credentials are assigned to components and are not grouped into packages. They are therefore valid globally for the respective component.

1. Select a component to edit it.



The context menu with *Credentials* opens.

You can configure the *credentials* with the following elements:

1 Configure new credentials

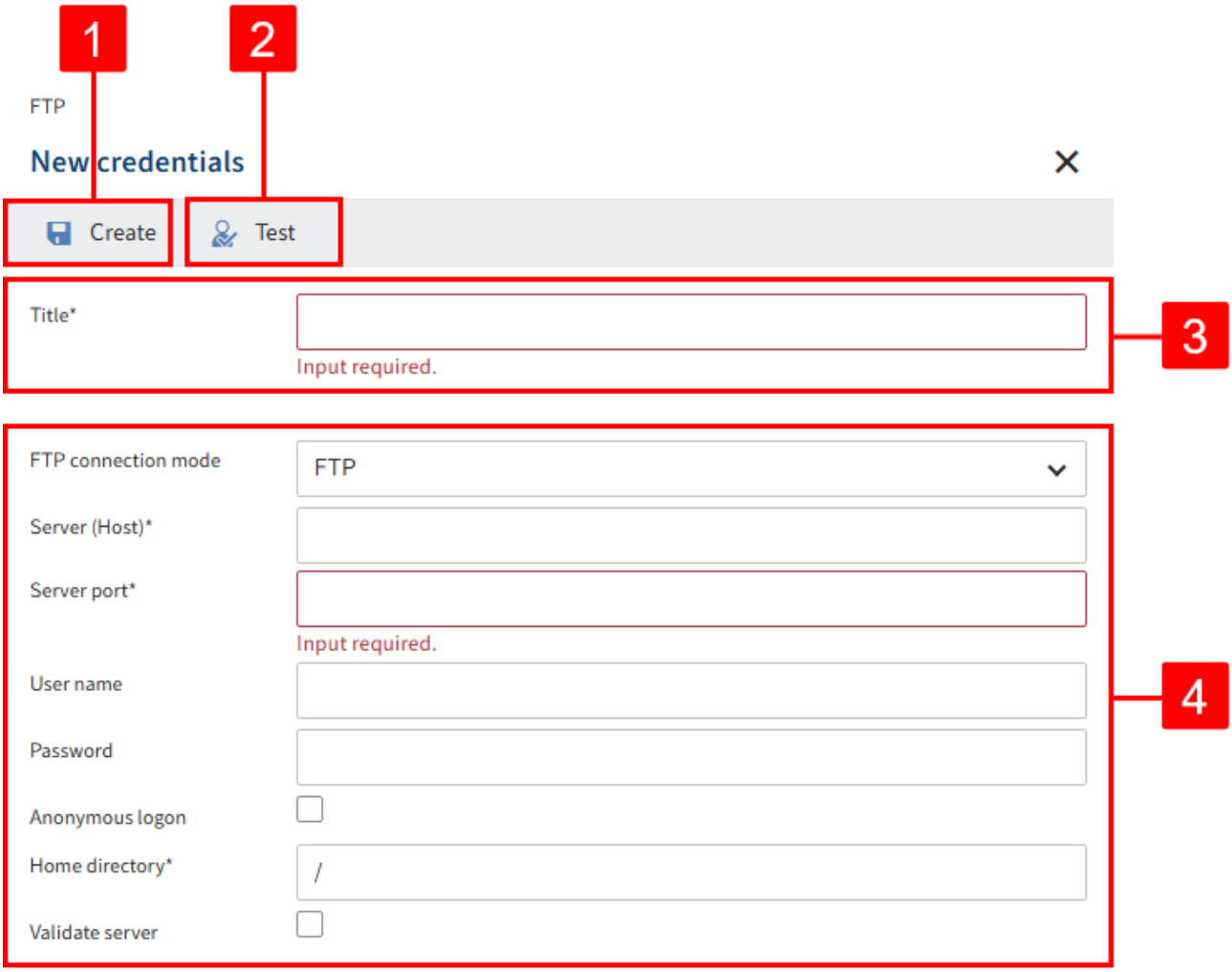
2 Edit a selected credential object

3 Delete a selected credential object

4 Select credential object

> **Information**
>
> There are components that offer the *Default* credential object. In this case, *Default* is shown as standard and assumes the credentials from the config file, e.g. the data of the "Windows service user".
>
> You cannot edit or delete the *Default* credentials object.

Create new credentials



> **Information**
>
> The example image shows the credential object for the component *FTP*.
>
> The structure of a credential object may vary depending on the component requirements. You will find more information under *Components*.

1 Create: Saves the credentials

2 Test: Tests the current entries

3 Title: Shown in the overview for the credential object.

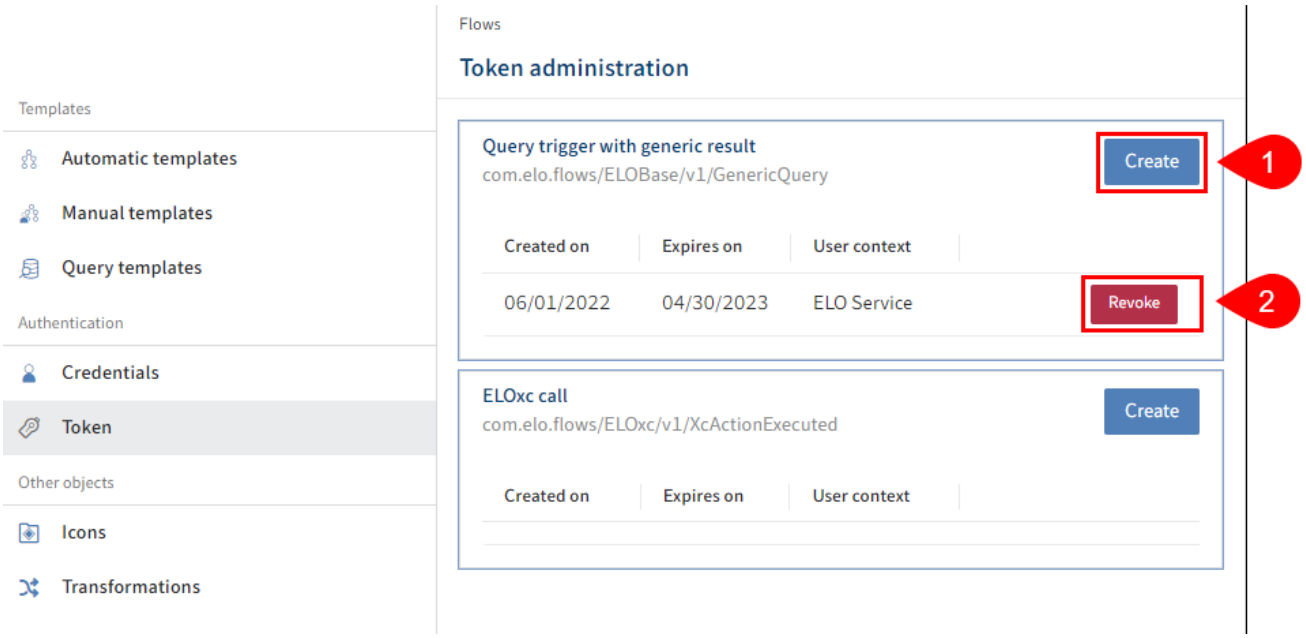
4 Configuration area: Input fields for mandatory data for the credential object

Webhooks and token authentication

You can safeguard trigger events that are triggered by third-party system calls by creating a token with an assigned user context. This check is based on the WebSub standard. Third-party systems that do not implement this standard can be provided the token as a query parameter in the webhook URL.

Token administration

The token administration area provides an overview of all endpoints and their respective tokens.



1. Selecting the *Token* entry in the sidebar opens the token administration area.

   All the endpoints used and their tokens are listed in the token administration area. You can also create, change, and revoke tokens here.

   The endpoint entries are collapsed to begin with for a better overview of the endpoints and their respective tokens. Only when you select an endpoint does the system show a list of the tokens for exactly this endpoint.

1 Create: Create a new token. You will find more under Token creation.

2 Edit: Double-click a token entry to open the editing dialog box. The user context and expiration date of the token can be changed there.

3 Revoke: Third-party systems that use this token are no longer authorized to trigger trigger events.

Token creation

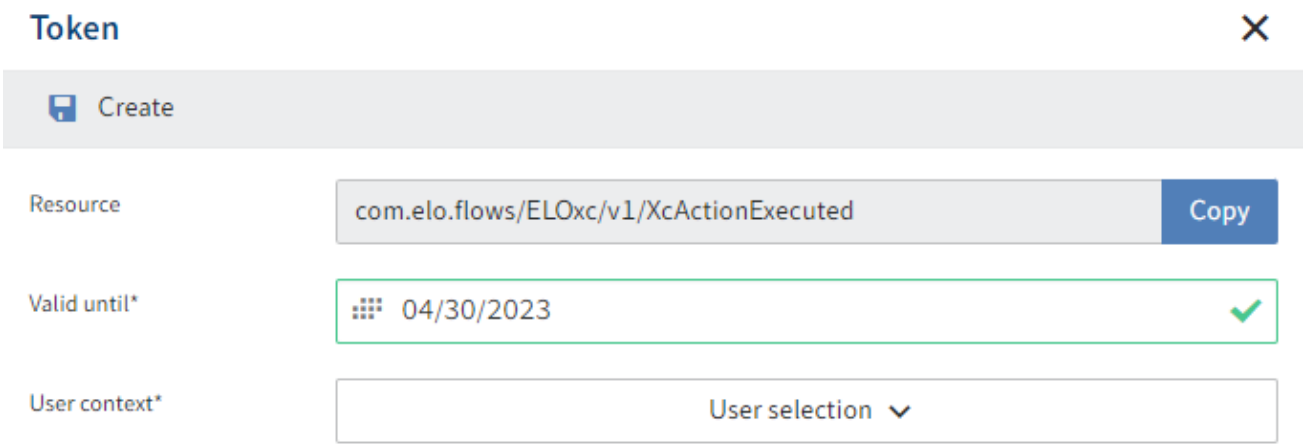
A token requires a user context and an expiration date. This can be configured in the settings dialog box.

1. In the token administration area, select an endpoint.

The endpoint expands.

1. In the endpoint, select *Create*.

The *Token* dialog box opens.

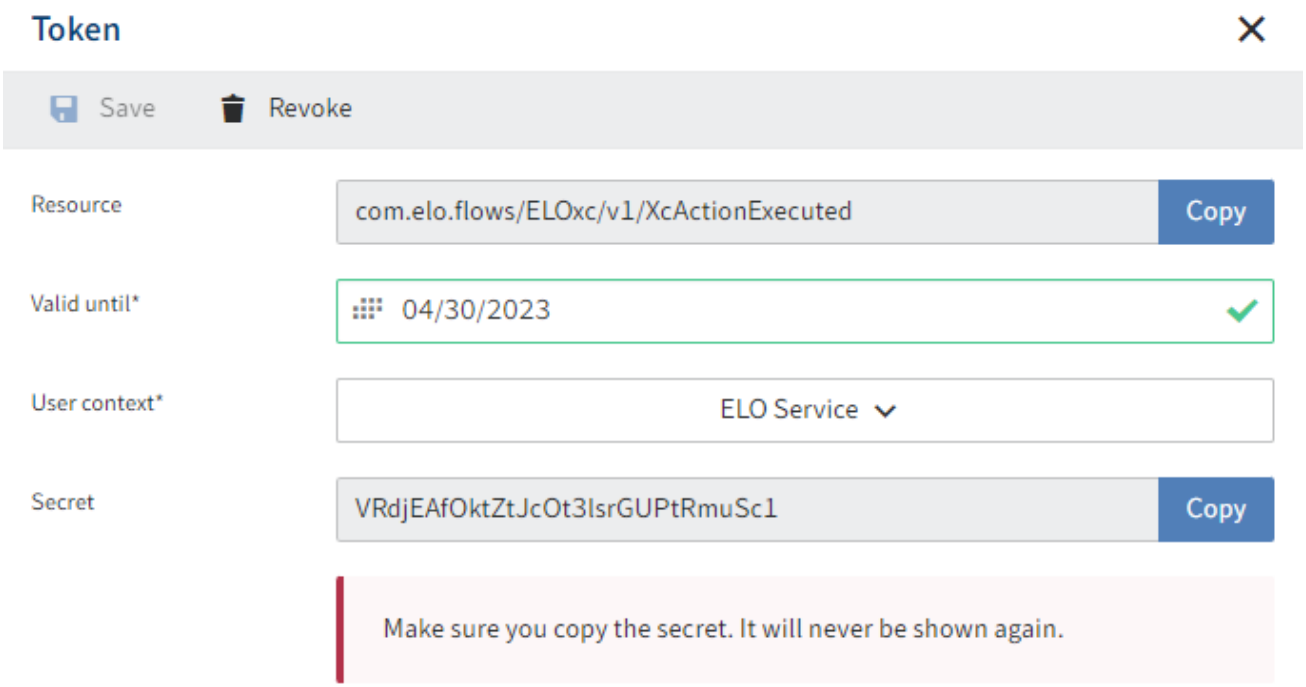


1. In the *Valid until* field, enter how long the token should be valid.
2. Select *Select users*.

   A search field opens.
3. Search for the user you want the token to run through.

   > **Information**
   >
   > The user must be created in the *Credentials* menu item.
4. Select a user.
5. Select *Create*.

ELO Flows creates a new token.



After creation, the secret used for webhook registration is shown in the dialog box.

Important

The secret cannot be viewed again. Create a copy for this reason.

Query parameter

Not every third-party system supports the WebSub standard. To be able to use this kind of system anyway, you can provide a query parameter containing the token secret to the webhook URL. This URL is then used to register the webhook in the third-party system. As a result, the secret is provided from the third-party system when the trigger event is triggered for authentication.

A possible endpoint can look like this, for example:

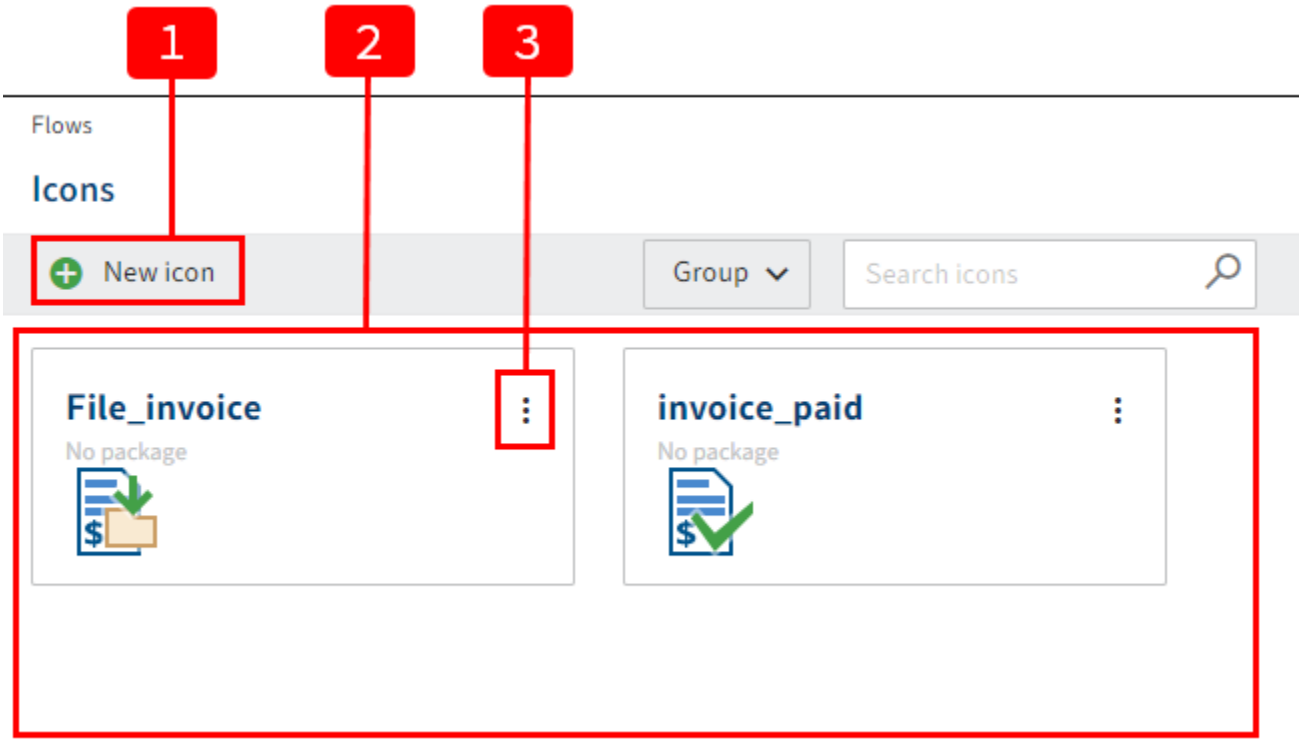
```
http://<server>:<port>/ix-<repository>/plugin/de.elo.ix.plugin.proxy/
   flows/api/v1/trigger/com.elo.flows/ELOBase/v1/ObjectCreated?token=07Cz3CCkYiW32OXrRuJZ0bFjyou
```

Please note

Query parameters should only be used in the event of a possible HTTPS connection.

Icons

Select *Icons* to see all the icons available for ELO Flows. These icons can be used for *user actions* for buttons in the ELO clients.



You have the following configuration options:

1 Add new icon

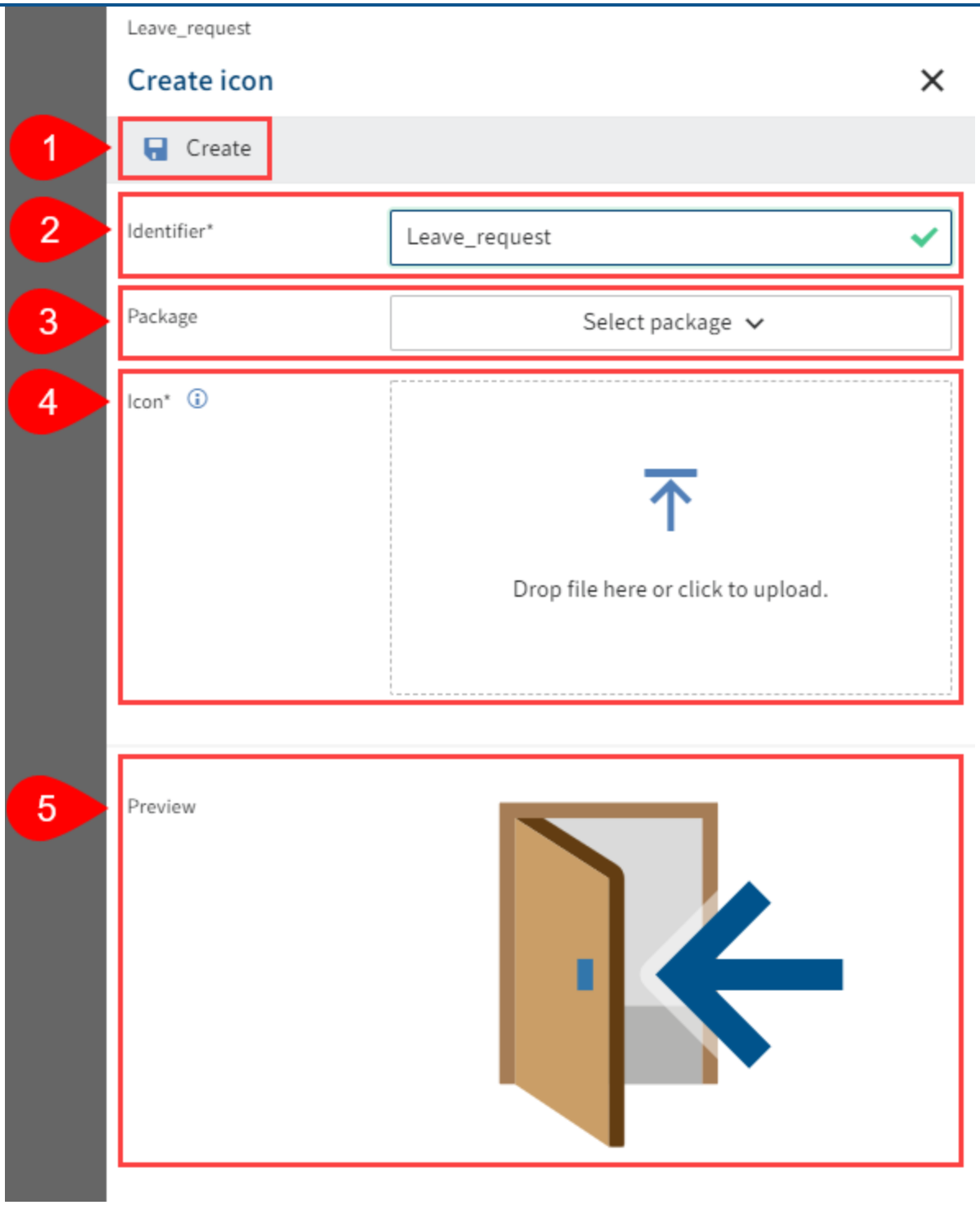
2 Select existing icons

3 Edit or delete an existing icon

> **Information**
>
> You cannot change the name. Create a new entry if you need an icon with another name.

Create icon

1. Select *New* to create a new icon.



You have the following configuration options:

1 Create: Saves the configuration

2 Name: Unique, technical name of the icon

Please note

The name cannot be changed later.

Information

Do not use special characters.

The following characters are allowed:

- a-z
- A-Z
- 0-9
- äöü
- ÄÖÜ
- ß
- _

3 Select package: Assigns the icon to a package as needed

4 Icon: Upload image file.

Information

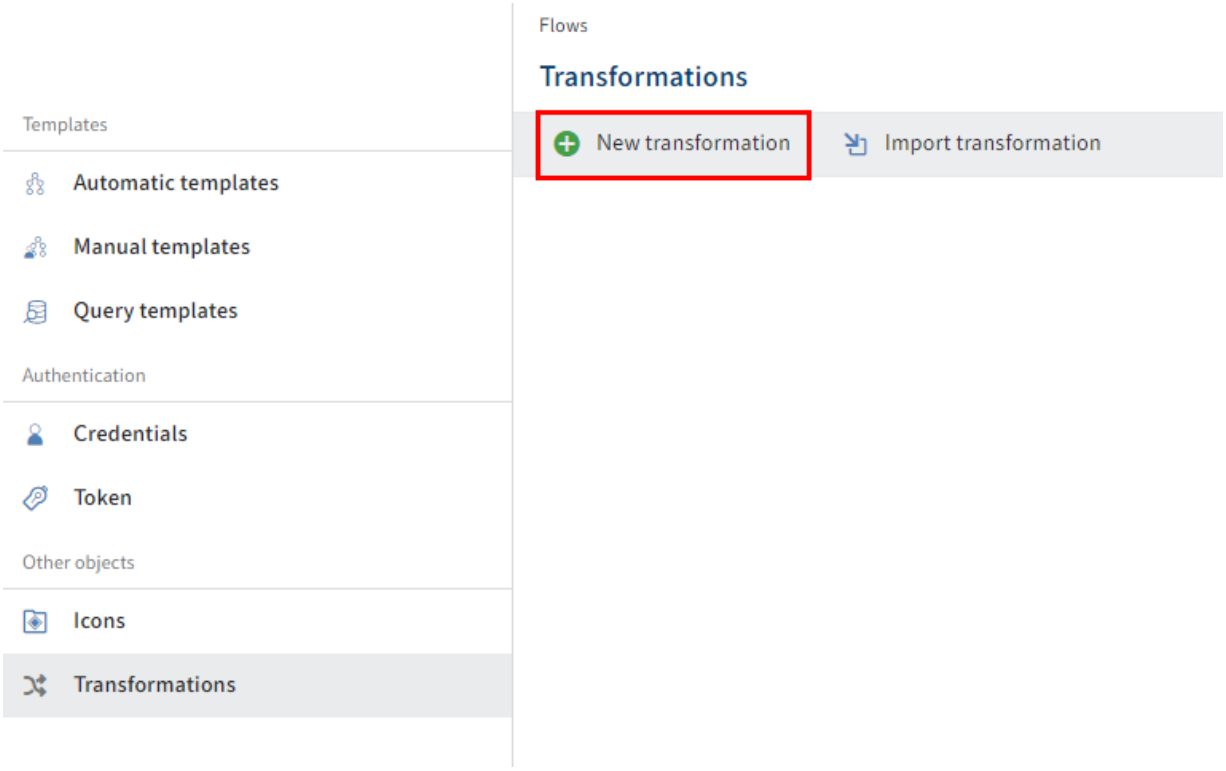
- Maximum size: 1 MB
- Format: SVG

Transformations

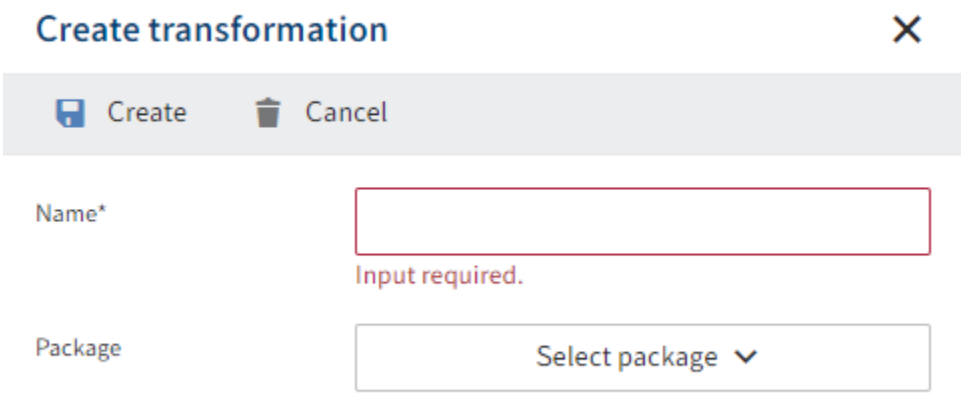
Transformations allow you to map complex objects from third-party systems to services in ELO Flows.

Create transformation

1. Open the *Transformations* menu item.



2. Select *New transformation*.



The *Create transformation* dialog box opens.

3. Edit the following fields:
   - Name: In the *Name* field, enter a unique name for the transformation.

     Please note

     The name cannot be changed later.

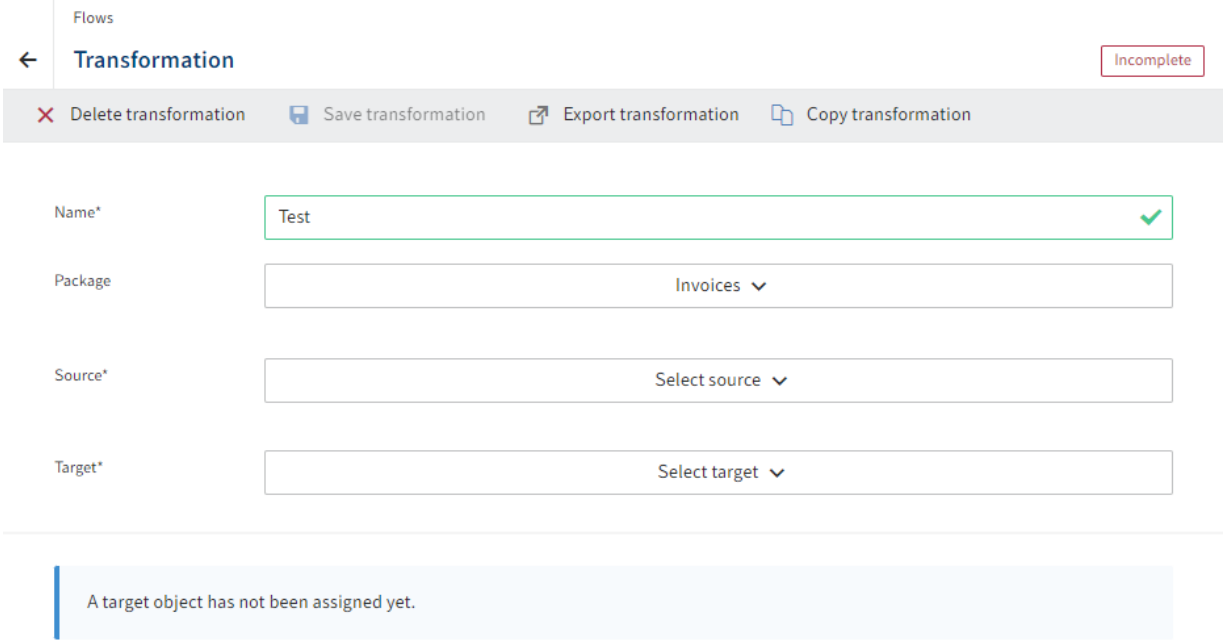
     Information

     Do not use special characters.

     The following characters are allowed:

     - a-z
     - A-Z
     - 0-9
     - äöü
     - ÄÖÜ
     - ß
     - _

   - Select package: You can assign the transformation to a package via the *Select package* drop-down menu as needed.
4. Select *Create*.



The *Transformation* dialog box opens.

5. Select the source object you want to link from the *Select source* drop-down menu.
6. Select the target object from the *Select target* drop-down menu.
7. Select *Save transformation*

JSONata editor

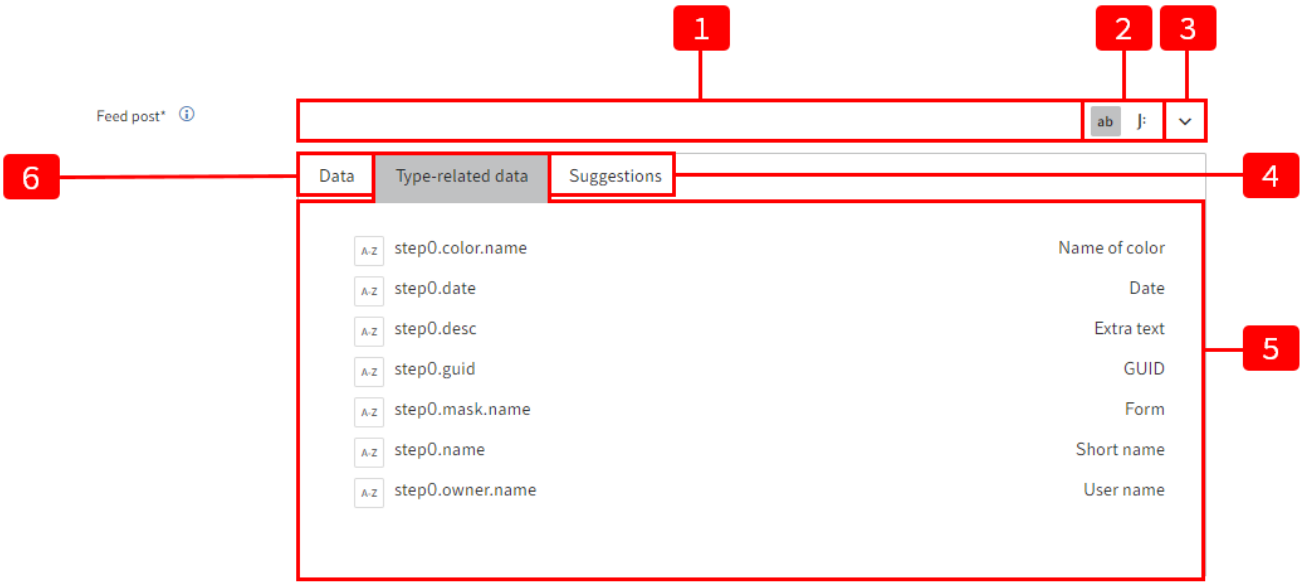
The *JSONata editor* in ELO Flows is responsible for defining *service properties*. If you create a service, the service will likely need data it can use to execute the respective action.

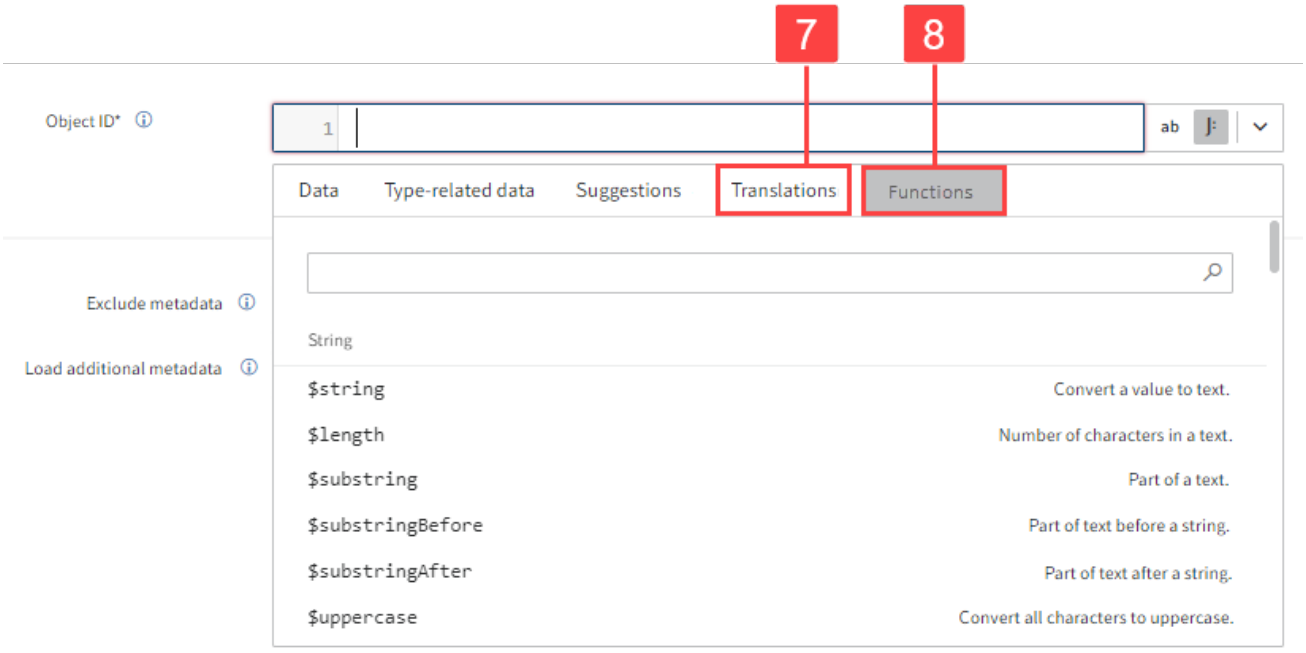
Example

One example is the object ID of an ELO object. This is used in the *Create post* service of the ELO feed component:

The service needs to know which object a feed post should be written for. The object is addressed via the object ID, which you can define or identify dynamically.

ELO Flows uses the transformation language JSONata for dynamic data access with data transformation.





You have the following configuration options:

1 Enter value: Input field for fixed values as a string or dynamic values as a JSONata expression

2 Select JSONata editor mode: The input field is in text mode by default. Text mode works with string values, which you can select from configured data (3). Expert mode (J icon) enables you to input JSONata commands.

3 Expand advanced input options: The button with the arrow icon expands the functions of the JSONata editor.

4 Use preconfigured suggestions

5 Select type-related data: Data is provided depending on the type of object you've selected. For example, the fields of a metadata form.

6 Select data from previous nodes

7 Select translation variables available in the package (expert mode)

8 Select pre-configured functions (expert mode)

> **Information**
>
> For more information about the structure and possibilities with JSONata, refer to jsonata.org

Advanced flow

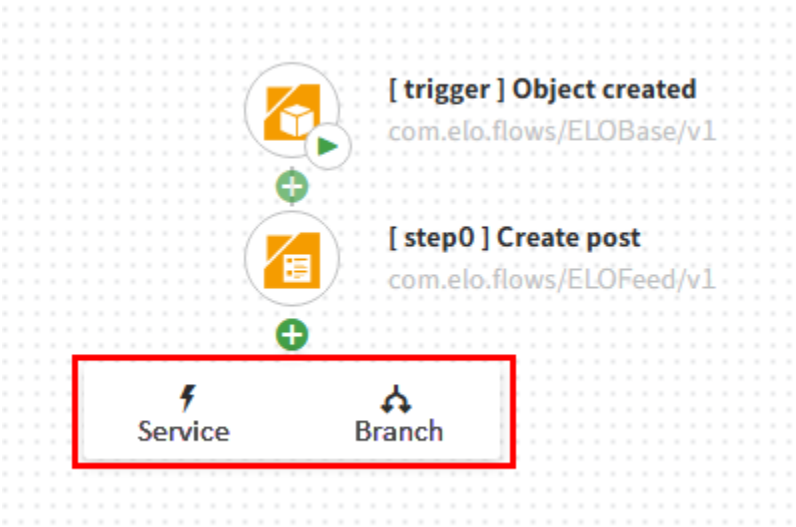
Branches

In ELO Flows, branches enable you to generate more complex workflows with a flow template via a true/false query. At the branches, ELO Flows checks a binary condition and runs the relevant flow based on the feedback.

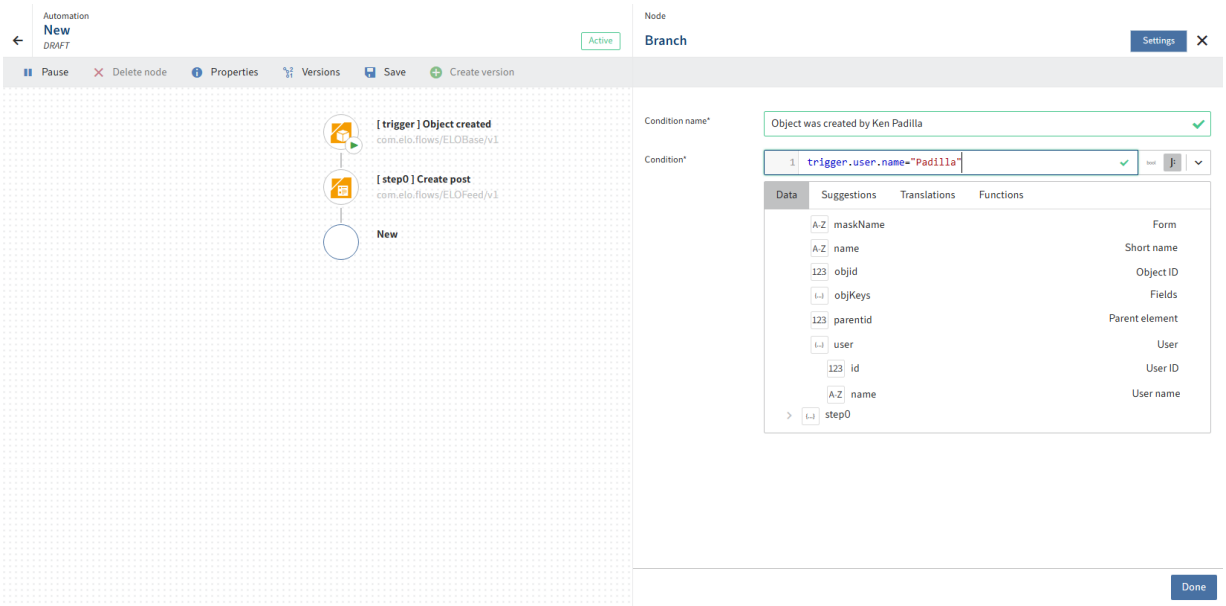
Configure branches

When creating your flow, decide whether you want to create a branch or a service after each node.

Method



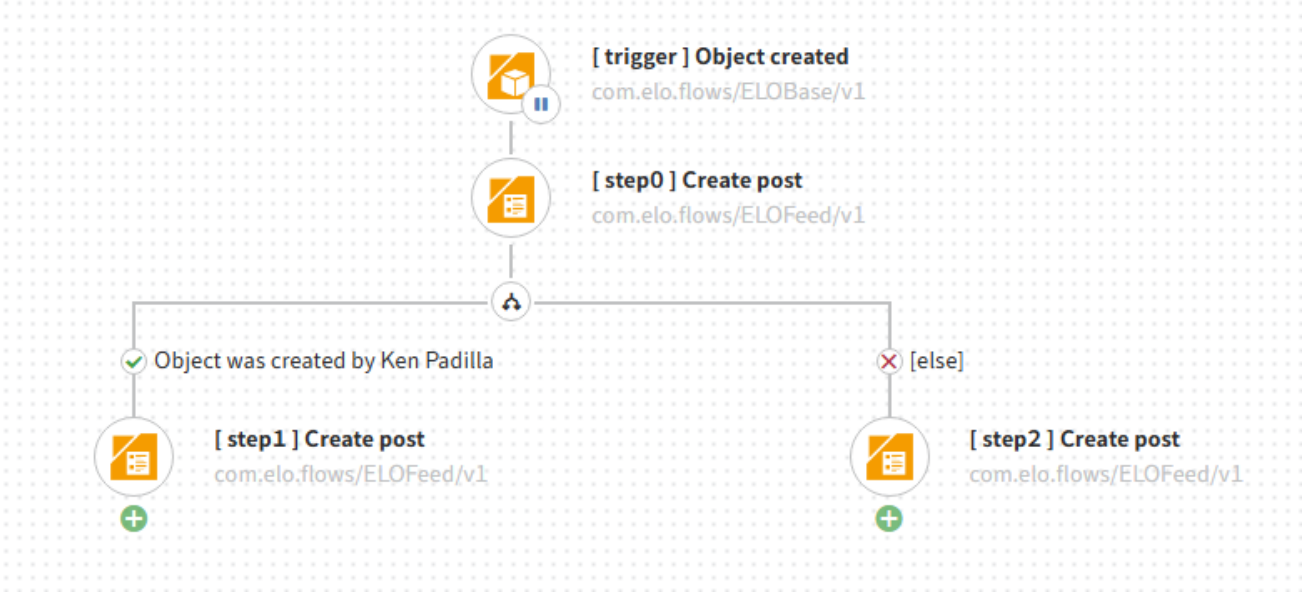
1. Select *Add node* (button with plus icon) and select *Branch*.



The context menu for the new *branch* opens.

2. Enter a title for the branch condition. After you confirm the change in the context menu, this title appears for the true condition in the flow designer.
3. In the JSONata editor, configure the *condition* that the binary branch refers to. These conditions determine which branch is run over the course of the flow.
4. Confirm with *Done*.

Result



> **Information**
>
> You can create the branch node at the end of the existing flow or between two existing nodes.
>
> In a synchronous flow, an end node is automatically placed at the end of each branch when creating a branch.
>
> Refer to the Synchronous flows section in this documentation for more information.

The context menu closes and you are back in the flow designer.

The branch with the true condition now has the name of the title entered for the branch condition in the flow designer.

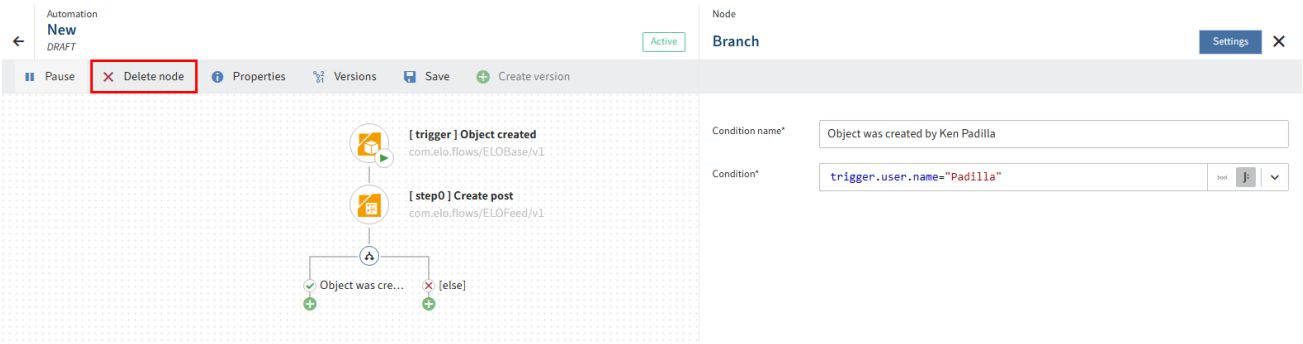
You can continue configuring your flow.

Select *Add node* to create either additional branches or services at the end of the branch.

> **Information**

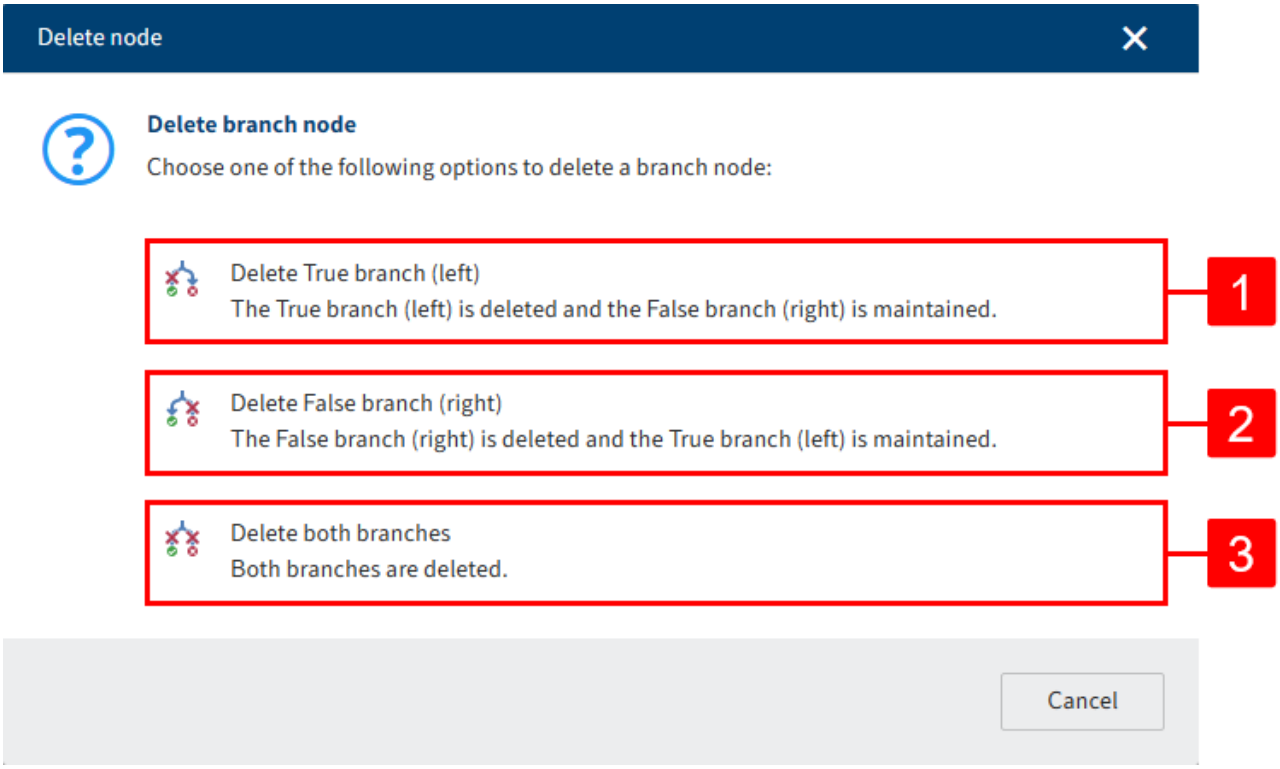
You do not have to create nodes at both ends of the branch. If an empty branch is run at runtime, the flow ends at this point. If both branches are empty, the binary branch is ignored.

Delete branches



As with all nodes in the flow designer, you can configure or delete a selected branch node after it has been created.

Selecting *Delete node* for a branch node opens a dialog box. Here, you can choose whether you want to delete single branches (1, 2) or both branches (3). In case 3, all steps following the branch are deleted.



Information

When creating a flow with branches, there are three possible situations that can have an effect on how the branch node is deleted.

Both ends of the branch are empty: If both ends of the branch are empty, the entire branch node is deleted. The flow ends with the last configured node before the branch was inserted.

One end of the branch is empty: If only one end of the branch contains more nodes, the branch is inserted in the flow in place of the branch node.

There are configured nodes at both ends of the branch: If both ends of the branch have configured nodes, you can decide whether to delete the entire branch node or a specific end of the branch. If you delete the entire branch node, the nodes that follow the branch are also deleted. If you delete one branch, the other ranch is inserted in the flow in place of the branch node.

Synchronous flows

ELO Flows offers two options for execution: synchronous and asynchronous flows.

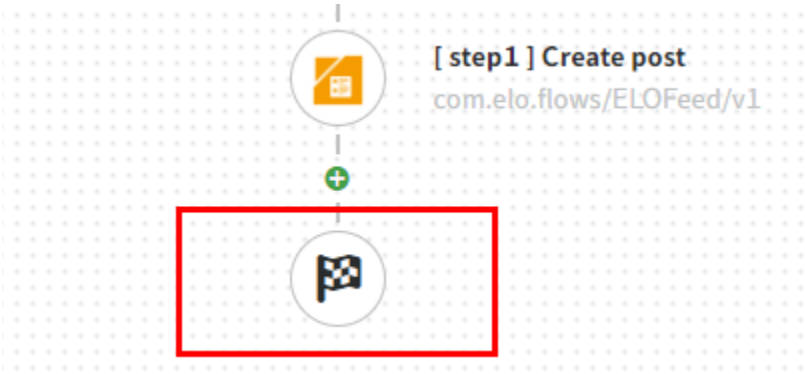
Flows configured as synchronous flows are executed with feedback. This enables you to show an info box in the ELO client once a flow has been completed, for example.

All flows are asynchronous by default. This means that the flows are executed without final feedback. This may make sense if you want to prevent the user interface from being blocked, for example.

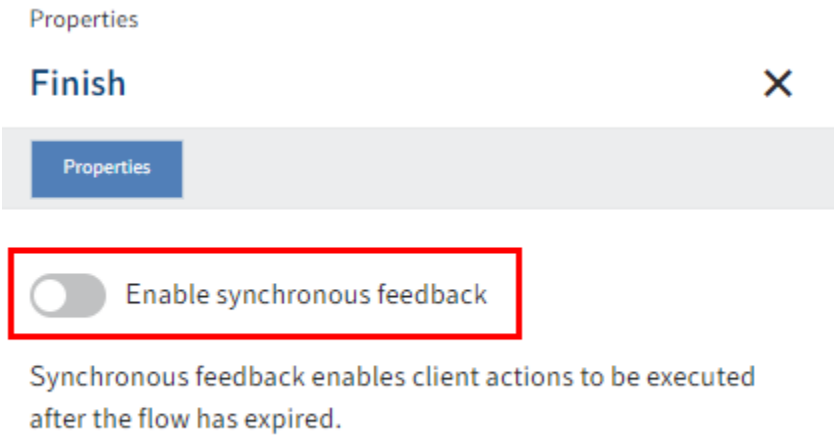
The following section explains how you can enable synchronous flows for user actions and recognize synchronous flows with automations.

User actions with synchronous flows

1. Open a flow within the user actions.



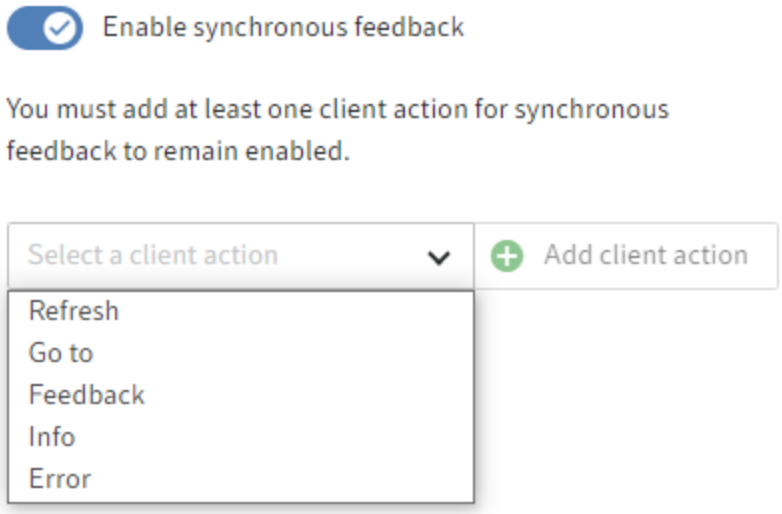
2. To enable the synchronous feature, select the flag symbol in the flow designer.



   The *Properties* page opens.

3.

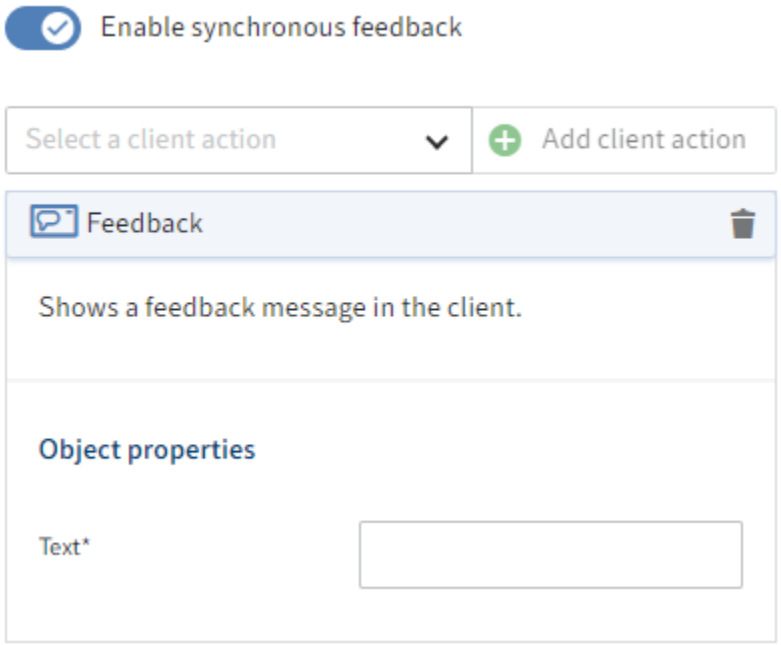
Enable the *Enable synchronous feedback* option.



A drop-down menu with different actions opens.

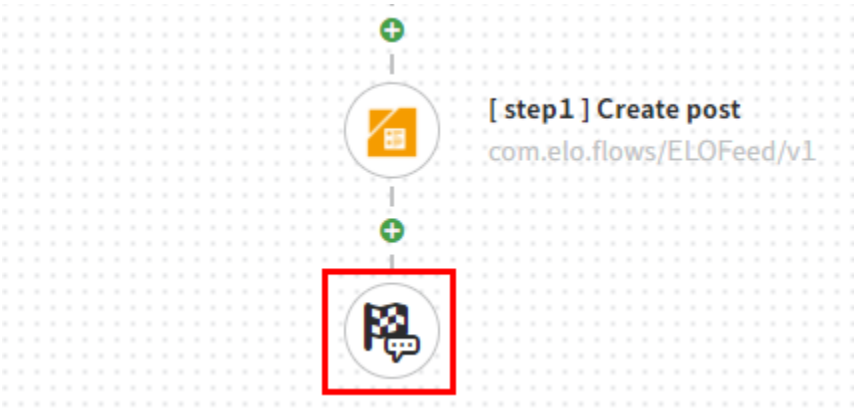
- Refresh: Refreshes the ELO client once the flow is processed.
- Go to: The ELO client goes to the configured object.
- Feedback: The ELO client shows a toast with the configured message.
- Info: The ELO client shows an info box with the configured message.
- Error: The ELO client shows an error box with the configured message.

4. Select an action.
5. Select *Add client action*.



> **Information**
>
> You can link multiple client actions.
>
> However, not all combinations make sense.
>
> Example: A *Go to* action implies a *Refresh* action. So, you don't have to link them.

6. Configure the desired object properties.
7. Select *Done*.



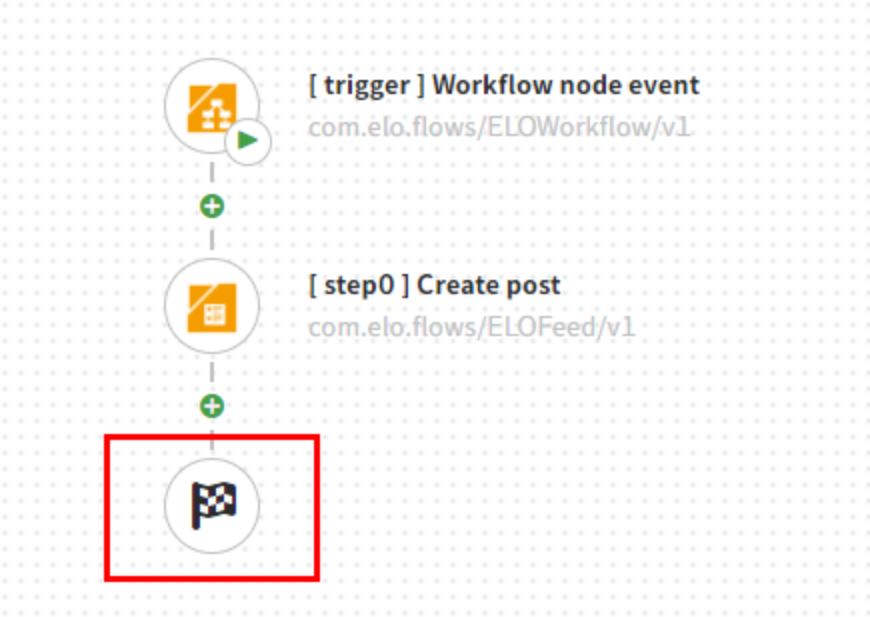
   The selected action(s) are linked with the flow. The flag symbol changes.

8. Select *Save.*

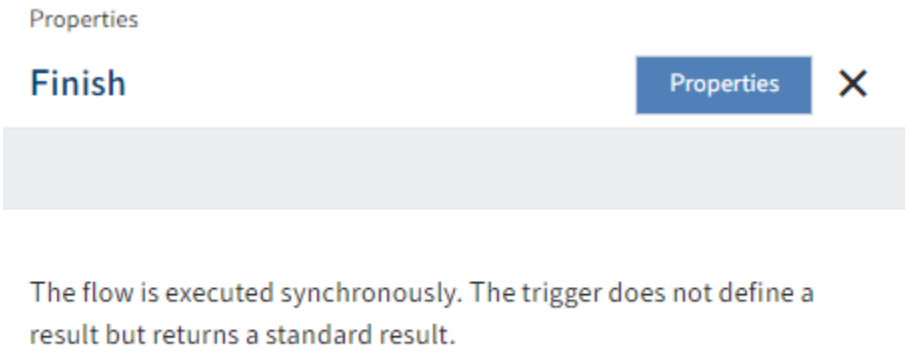
   The flow is saved.

Automations with synchronous flows

If a trigger with synchronous feedback is used, the flag symbol appears in the flow designer in the automations.



Whether the flag symbol appears depends on the component used. If the selected trigger does not support synchronous feedback, the flag symbol is not shown.

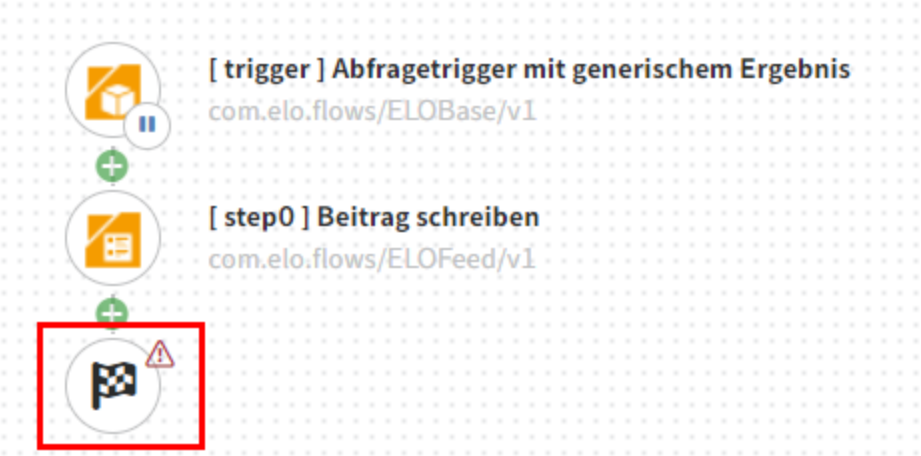


Information

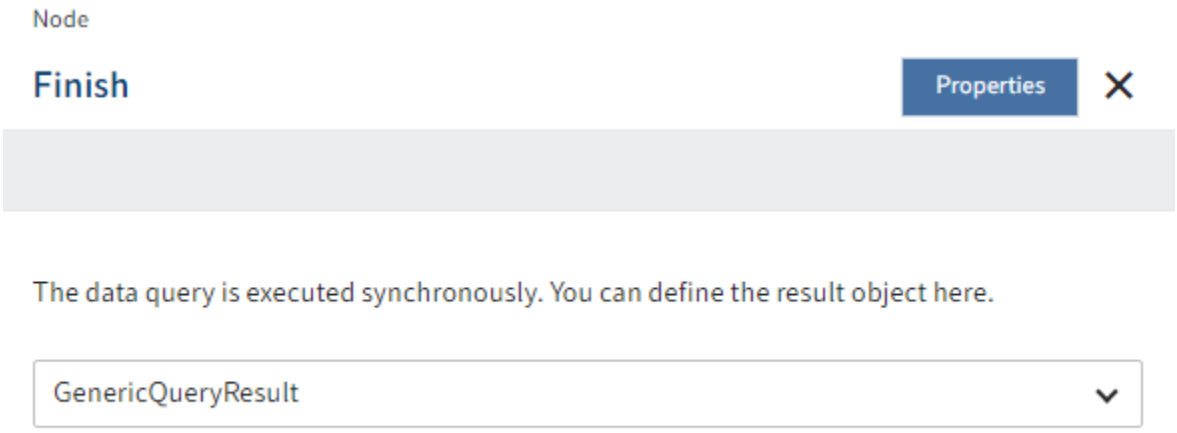
Unlike with *user actions*, no properties can be configured for synchronous feedback in *automations*. In this case, the trigger defines the result. The flag symbol here is only intended as an indicator of synchronous feedback taking place.

Data queries with synchronous flows

Data queries with synchronous flows require a result definition.



1. Select the end node (flag symbol).



   The *Finish* page opens.

2. Select a result type from the drop-down menu.

   Optional: Edit the object properties if necessary.

3. Select *Done* to apply the result definition.

4. Select *Save*.

   The status display changes to Active.

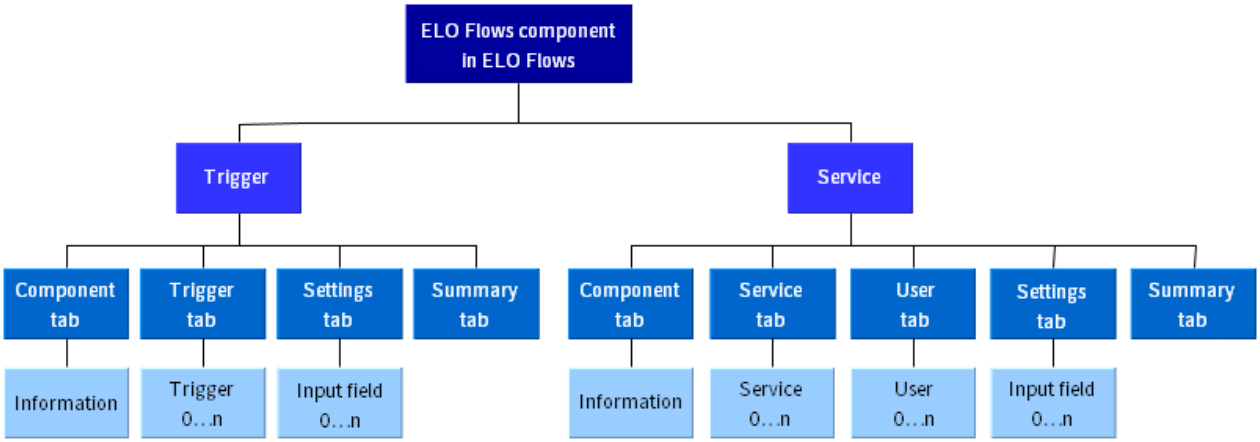
Components

What are components in the context of flows?

In flows, components are self-contained software elements that extend ELO's standard functions.

The components can be programmed independent of customer systems and integrated into the respective customer system as needed via standardized interfaces.

In the context of flows, a component is output as a JAR file and provided for further use in the ELO Flows worker.



Components can be triggers, services, or a combination of the two. In addition, information fields can be integrated into flows.

ELO provides several components as standard. You can also develop custom solutions.

The following describes the components provided by ELO as standard.

Install new or custom components

To provide custom components in the flow administration area, roll out the JAR file in the flow worker (Apache Karaf). Deployment is realized via the *deploy* directory of the flow worker.

Copy the JAR file of the new component to the *deploy* directory.

A possible path to this directory could be: `ELO\servers\ELO-Flows-Worker-1\deploy`.

For more information on developing components, refer to the ELO Flows component development documentation.

General information

Components are configured using input fields among other things. The following goes into more detail about the input variants.

Determine the ARCPATH syntax and object ID

With an ARCPATH, you can specify a start or target element via a folder path. The path can also be specified relative to a SORD.

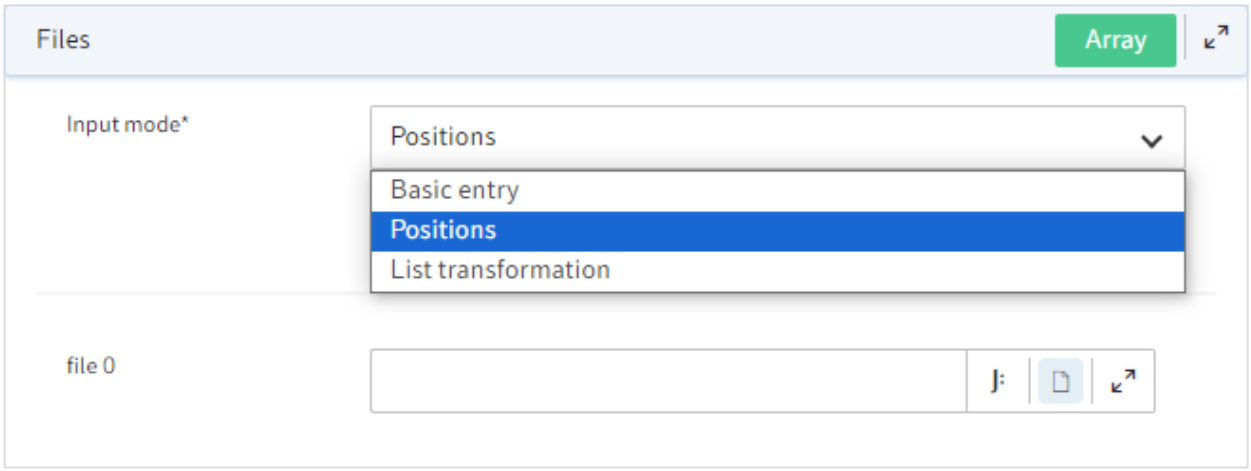
Type	Description	Example
<ID> <GUID>	Enter a static ID or GUID.	Static ID: `1234` GUID: `(E10E1000-E100-E100-E100-E10E10E10E00)`
ARCPATH: <path>	After the colon, enter the filing path. The first (any) character always separates the level in the folder structure.	`ARCPATH:¶Accounting¶New`
ARCPATH[<GUID>]:<path> ARCPATH[<ID>]:<path>	Enter an object ID or GUID in square brackets before the colon. A path is then identified relative to the start point.	`ARCPATH[(E10E1000-E100-E100-E100-E10E10E10E00)]:¶Business Solutions Custom¶Configuration`
OKEY:<field name>=<value>	Searches for an exact entry in a field. The value should be unique. Only an object ID is returned.	`OKEY:ELOINDEX=incinvoice`
LMATCH:<field name>=<fixed value>%<search string>	Similar to OKEY. Searches for the exact fixed value and longest matching search string in the named field.	`LMATCH:CUSTOMER_NAME=Contelo%Holdings`

Using gen. 1 and gen. 2 metadata

With flows, you can use and process gen. 1 and gen. 2 ELO-specific metadata.

Type	Metadata gen.	Description	Example
Metadata form	(gen. 1)	Identify the metadata form with the name of the metadata form: `<MASK_NAME>`	`INVOICE`
	(gen. 2)	Identify the metadata form with the name of the package. Enter the name of the package as the prefix, separated from the name of the metadata form with a period: `<PACKAGE_NAME>.<MASK_NAME>`	`BASIC.PERSON`
Data field	Index field (gen. 1)	Set the index field via the field group name (*Key*) `<FIELDGROUP_NAME>` and the field value (*Value*) `<FIELD_VALUE>` In ELO Flows, index fields are also referred to as *ObjKey fields*.	Key: `INVOICE_NO` Value: `42`
	Aspect field (gen. 2)	Identify the aspect field with the name of the aspect: `<ASPECT_NAME>.<ASPECT_FIELD_NAME>`	`PERSONAL_DATA.FIRSTNAME`

'Input mode' in the context of lists and objects

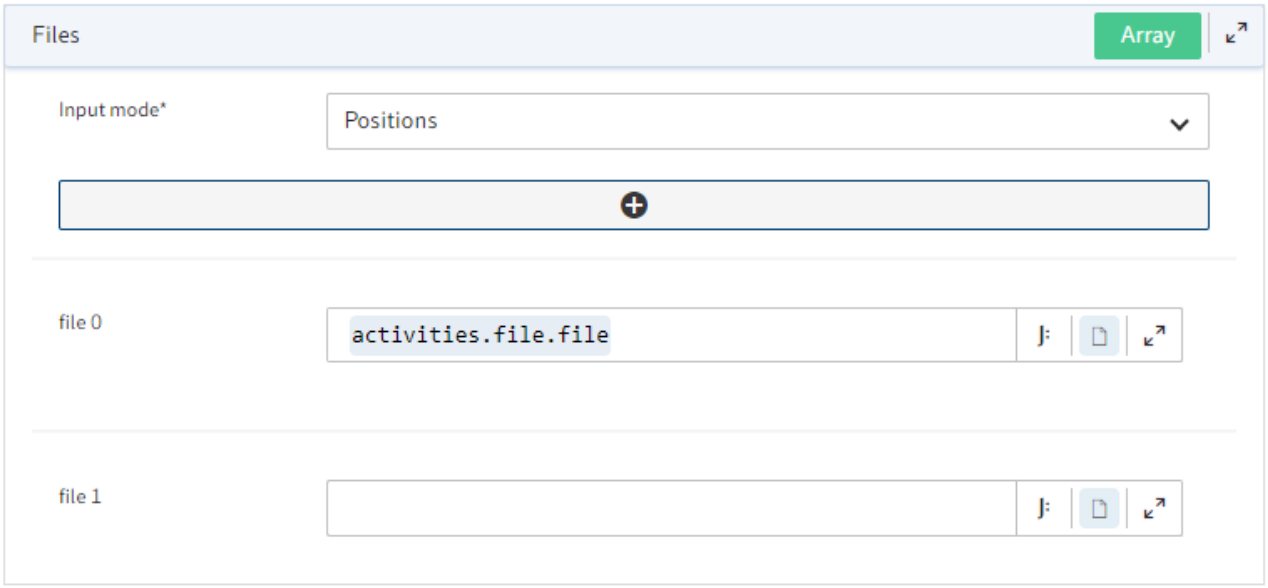


Generate new lists and objects in the context of flows.

ELO Flows offer you *Input modes* in the configuration area to support you when designing lists or objects.

When selecting an input mode, you can decide on the support you want when you are making the entry. The input modes *Positions*, *List transformation*, and *Basic entry* are available.

Positions

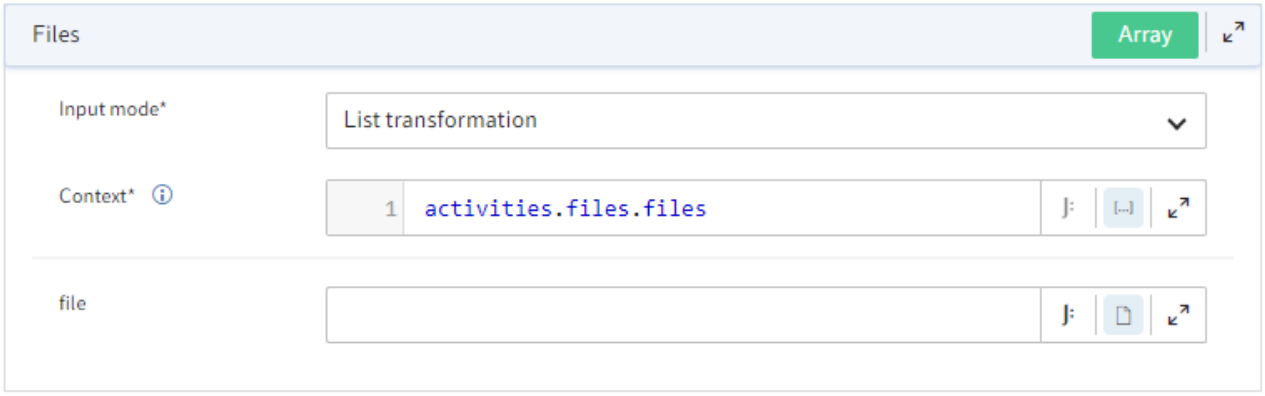


Create a list or an object with a fixed number of entries manually. You can fill the entries statically or dynamically. You determine the size of the list or object statically.

Add the desired number of entries using the plus button. Use the JSONata editor in the input fields to integrate dynamic field values.

You can find more information about the JSONata editor in ELO Flows in the *JSONata editor* section.

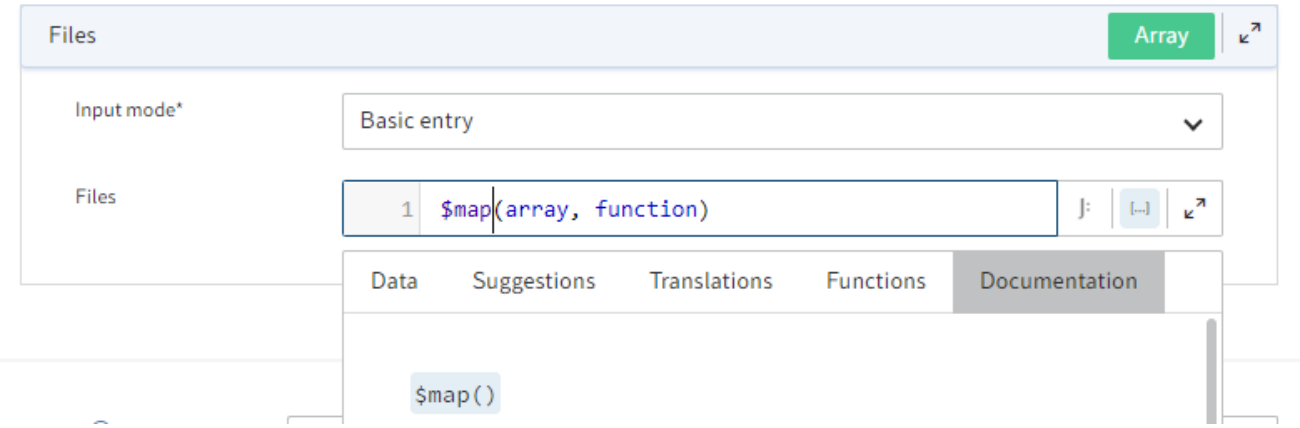
List transformation



Design a new list dynamically on the basis of an initial list.

Enter the initial list in the *Context* field and in the next field, define how the elements will be transformed in the initial list.

Basic entry



Generate a new list or a new object entirely using JSONata commands. JSONata allows you to use both static and dynamic processes.

You can find more information about the JSONata editor in ELO Flows in the *JSONata editor* section.

ELO objects and metadata

Goal and use

The *ELO objects and metadata* component provides you with basic functions for communicating with an ELO repository.

Credentials and connection data

Enter connection data for the *ELO objects and metadata* component. A *default connection* is available. This is entered in the config files. The *user context* assumes the session of the user logged into the ELO client. You can also create additional connections to run services with defined rights.

Triggers

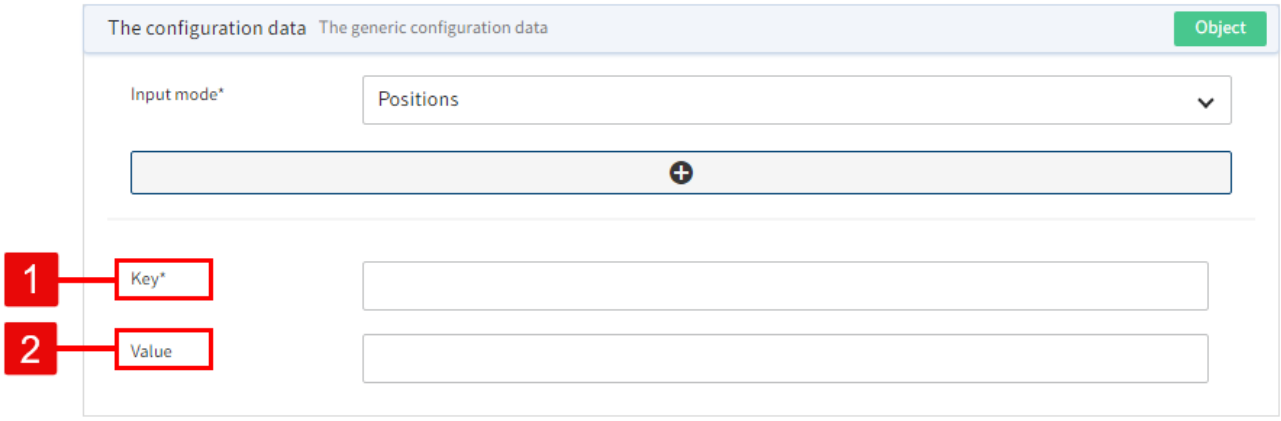
Triggers are the starting points of a flow. The *ELO objects and metadata* component offers the following triggers:

Name	Event	Additional information
Query trigger with generic result	Started when the endpoint is called with an HTTP POST. The trigger can be started with gen. 1 and gen. 2 metadata.	Synchronous flow
Dynamic keyword list called	Gen. 1 metadata: Configure dynamic keyword lists with a flow call in the *Form designer* or in the *Field templates*. Gen. 2 metadata: Configure dynamic keyword lists with a flow call in the aspects.	Synchronous flow
Object created	Started when an object is filed in ELO.	Asynchronous flow
Object edited	Started when an object in ELO is changed (metadata, versions, etc.).	Asynchronous flow

Trigger configuration 'Query trigger with generic result'

You can transfer configuration data via a REST call or forego the transfer of configuration data entirely. To do this, do not make any additional configurations on the *Settings* tab.

Advanced settings



Optional: Enter configuration data via the *Positions* input mode, e.g. data from a third-party system. Enter your user-defined data in the key-value list. The key (1) is a mandatory field, while the value (2) is optional.

Example entry

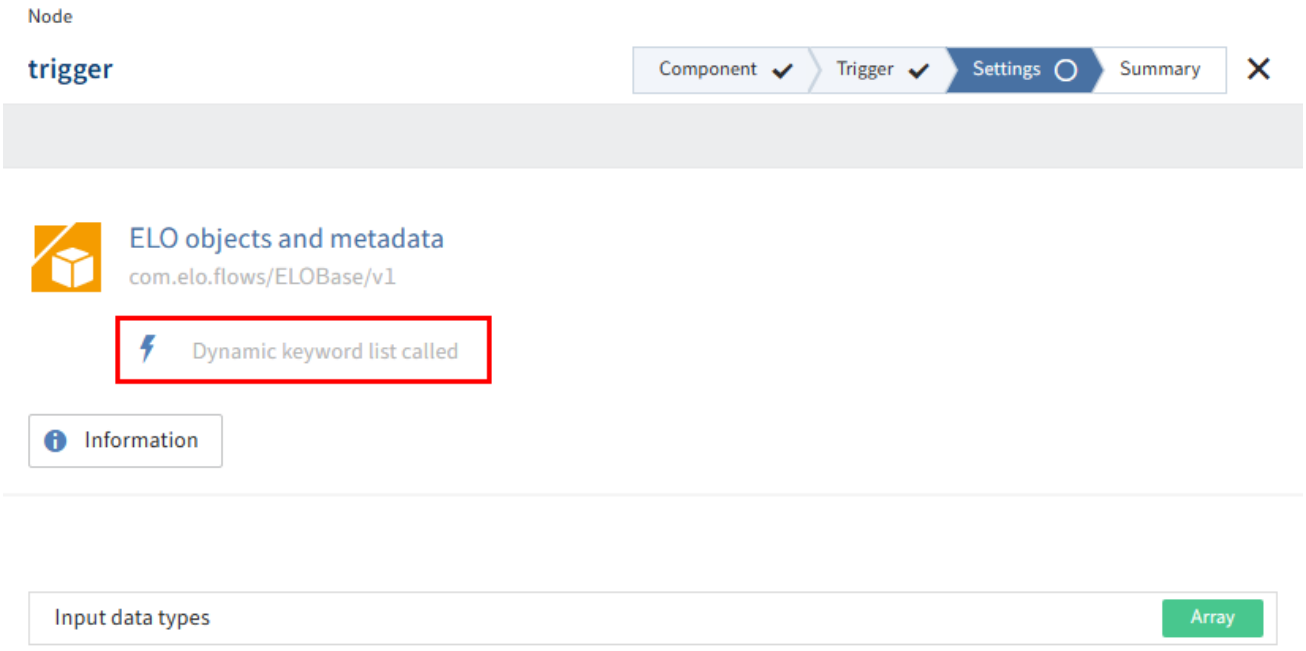
Key - `user`

Value - `renz`

Next steps with the 'Query trigger with generic result' trigger

This trigger is started when the endpoint is called with an HTTP POST. You can get the endpoint under *Summary > Service*.

'Dynamic keyword list called' trigger configuration



Call dynamic keyword lists with a flow call. The trigger can start in the context of gen. 1 and gen. 2 metadata. The call is structured as follows: `flows-plugin/trigger/trigger ID` and may look like this, for example: `flows-plugin/trigger/08e40e34-2e3e-4725-baed-1442d1e25143`.

For gen. 1 metadata, the data entered is not automatically transferred formatted. You can configure formatting on the *Settings* tab as needed. Gen. 2 metadata automatically provide formatting.

Configure flow call for dynamic keyword lists

To call dynamic keyword lists with flows, the flow call has to be configured in the metadata. The following explains more for both gen. 1 and gen. 2 metadata.

Gen. 1 metadata:

The trigger is started when an ELO Flows call is used to store dynamic keyword lists in the *Form designer (gen. 1)* or in the *Field templates (gen. 1)*.

You can read the trigger ID in the flow designer for the trigger under Summary > Properties > ID.

Configure in the Form designer:

1. Select a field.
2.

Open the properties for the selected field.

3. Under Keyword list, select the option Dynamic Keyword Map.
4. In the *Script name* input field, enter the call according to the convention `flows-plugin/trigger/Trigger-ID`.

Store in the Field templates:

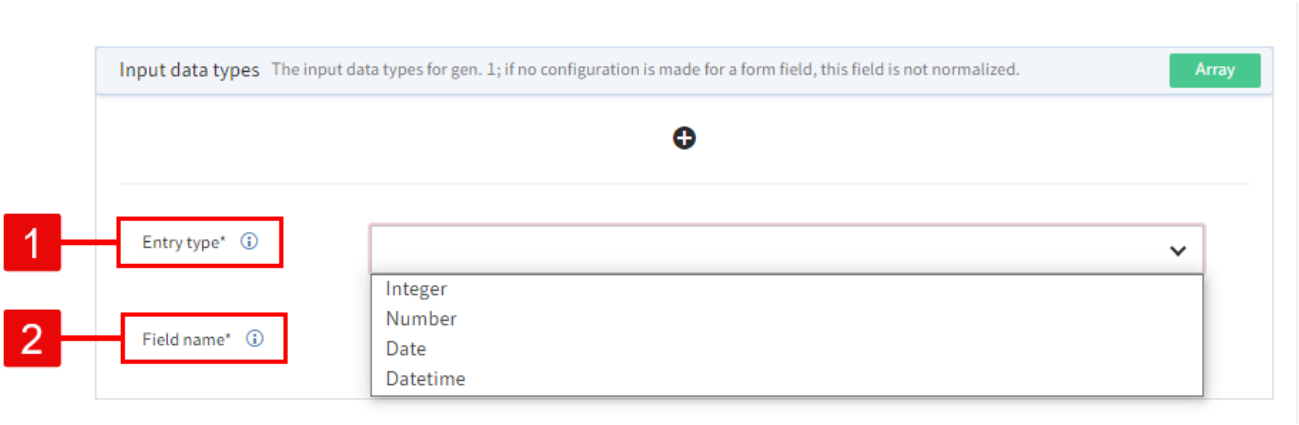
1. Select a field.
2. Enter the call under Keyword list > Dynamic keyword list.

Gen. 2 metadata:

The trigger is started when a flow call is used to store dynamic keyword lists in an aspect field.

1. Within your package, navigate to *Aspects*.
2. Select an aspect or add a new one.
3. Within the aspect, navigate to the *Fields* area.
4. Select a field or add a new one. The settings for the field open.
5. In the *Dynamic keyword list* input field, enter the call according to the convention `flows-plugin/trigger/Trigger-ID`.

Advanced settings



Optional: Manual formatting of gen. 1 metadata entries

1 Entry type: Select an entry type from the drop-down menu. The following options are available: Integer, Number, Date, Datetime.

2 Field name: Enter the field name with dynamic keyword list from the context of forms with gen. 1 metadata.

Example entry

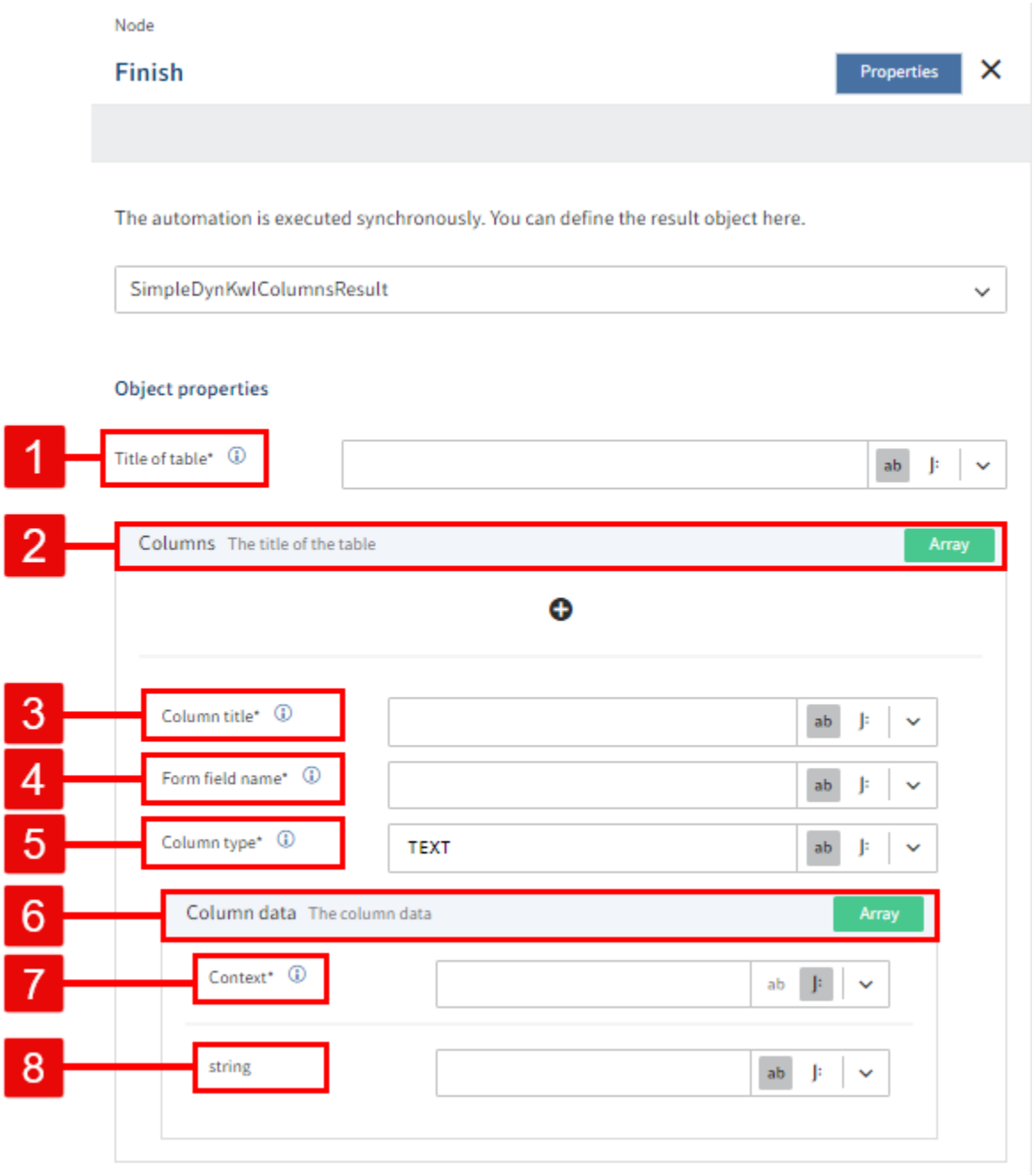
Entry type - *Date*

Field name - `Filing date`

Next steps for 'Dynamic keyword list called' trigger

Use the result *SimpleDynKwlColumnsResult* or *DynKwlColumnsResult* at the end of the flow template under *synchronous feedback*.

Result definition via 'SimpleDynKwlColumnsResult'



Configure the result object manually with the following object properties:

1 Title of table: The *Title of table* field is mandatory. Enter the title of the generated results table.

2 Columns: Open the advanced configuration for the columns by selecting the plus icon.

Information

Manual configuration of individual columns of the result table is optional. If you decide to configure columns, the fields *Column title, Form field name*, and *Column type* are mandatory.

3 Column title: Enter a title to be displayed for the column. If you don't want to show a column title in the table, enter `null`.

4 Form field name: Enter the name of the form field the data should be transferred to. If you don't want to transfer the data from this column to a form field, enter `null`.

5 Column type: Enter the data type of the column. `TEXT` is entered by default. Alternatively, you can enter `INTEGER`, `NUMBER`, `DATE`, or `DATETIME`.

6 Column data: Configure a custom list transformation.

Information

Manual configuration of the list transformation is optional. In this case, the *Context* field is mandatory.

7 Context: Enter the data context of the transformation, e.g. with relation to the previous *steps* within your flow.

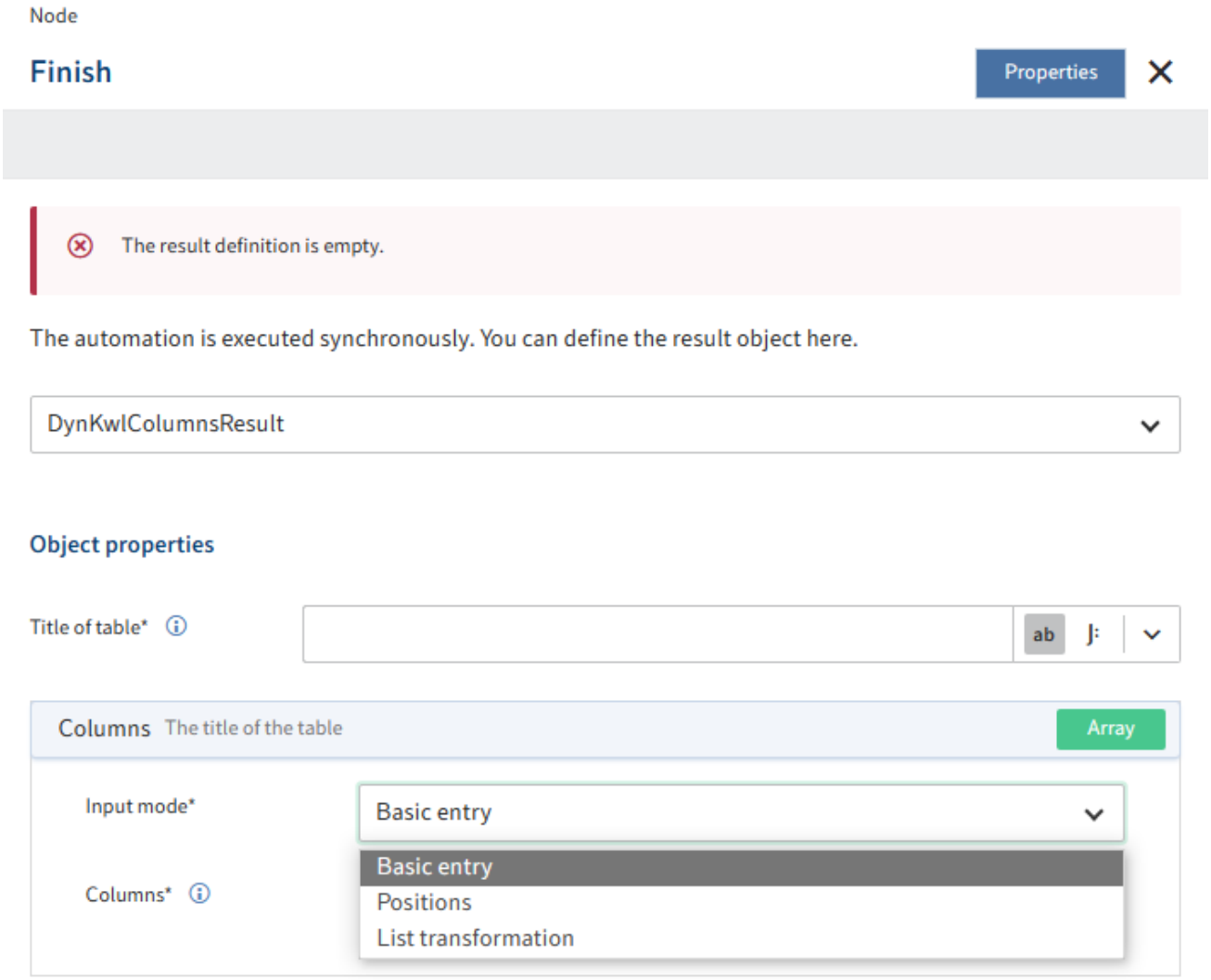
8 string: Enter the value you want to select.

Information

If you want to perform concatenation for the list transformation, you will have to make your entry in the JSONata flow editor. You can find more information under Known issues > Concatenation in transformation fields.

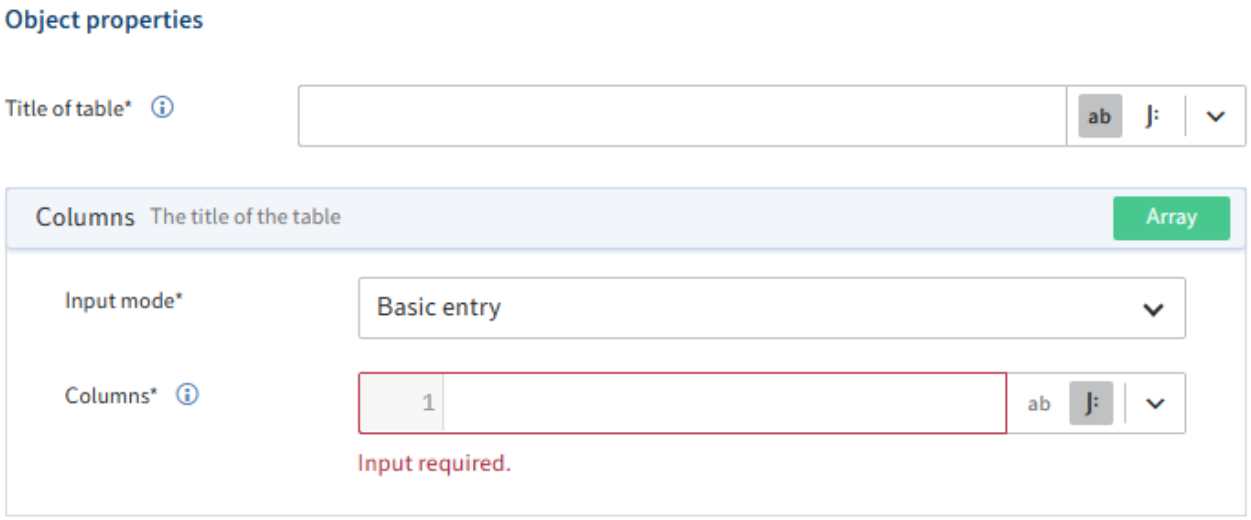
- Example entry
  - *Title of table* - `User list`
  - *Column title* - `User name`
  - *Form field name* - `PERSON.PERSONAL_DATA`
  - *Column type* - `TEXT`
  - Intended transformation: Transform List<Person> into a list of names List<String>.
  - *Context* - `stepBefore.persons`
  - *string* - `$.name`

Result definition via 'DynKwlColumnsResult'



Result definition via *DynKwlColumnsResult* provides advanced configuration options.When configuring the columns, you can choose between the modes *Basic entry*, *Position*, and *List transformation*.

'Basic entry' input mode

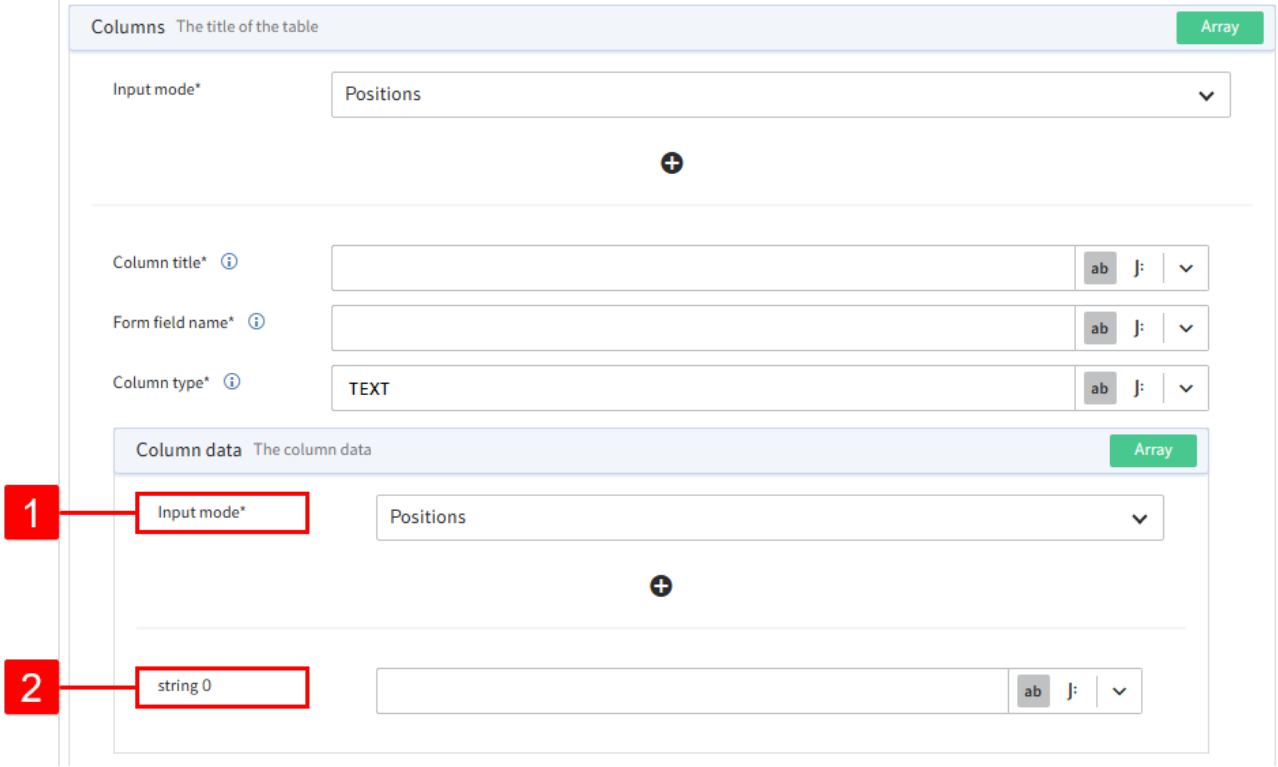


With the input mode *Basic entry*, you can customize the columns in the form of JSONata commands. Use the *Columns* field in the JSONata editor.

Example entry

```
//Example of a static variant for basic entry
[{
    "title": "Column1",
    "formFieldName": "Field1",
    "columnType": "TEXT",
    "data": [
        "Column1-Field1",
        "Column1-Field2"
    ]
},
{
    "title": "Column2",
    "formFieldName": "Field2",
    "columnType": "TEXT",
    "data": [
        "Column2-Field1",
       "Column2-Field2"
    ]
}]
```

'Positions' input mode

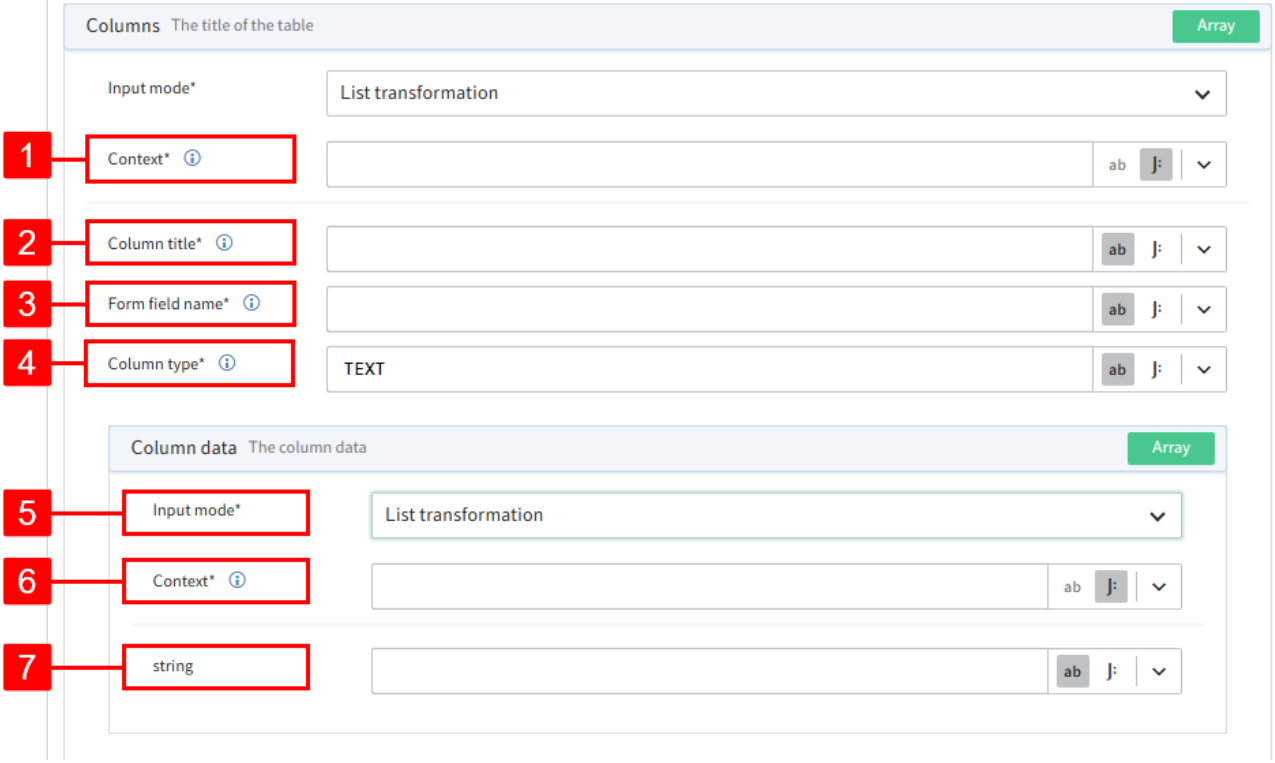


Information

The *Positions* mode corresponds to the setting options of the result definition via *SimpleDynKwlColumnsResult*.

Within the column data, you can also select the input mode (1) *Positions* and add string fields (2). Enter a fixed value in the string field. You can select preconfigured suggestions by selecting the arrow icon next to the field, for example.

'List transformation' input mode



The *List transformation* input mode focuses on the transformation of list data from previous steps of the flow.

1 Context: Define the context for the transformation, e.g. a previous step in the flow.

2 Column title: Enter the title of the column where the data from the transformed list should appear.

3 Form field name: Enter the name of the form field the data should be transferred to. If you don't want to transfer the data from this column to a form field, enter `null`.

4 Column type: Enter the data type of the column. `TEXT` is entered by default. Alternatively, you can enter `INTEGER`, `NUMBER`, `DATE`, or `DATETIME`.

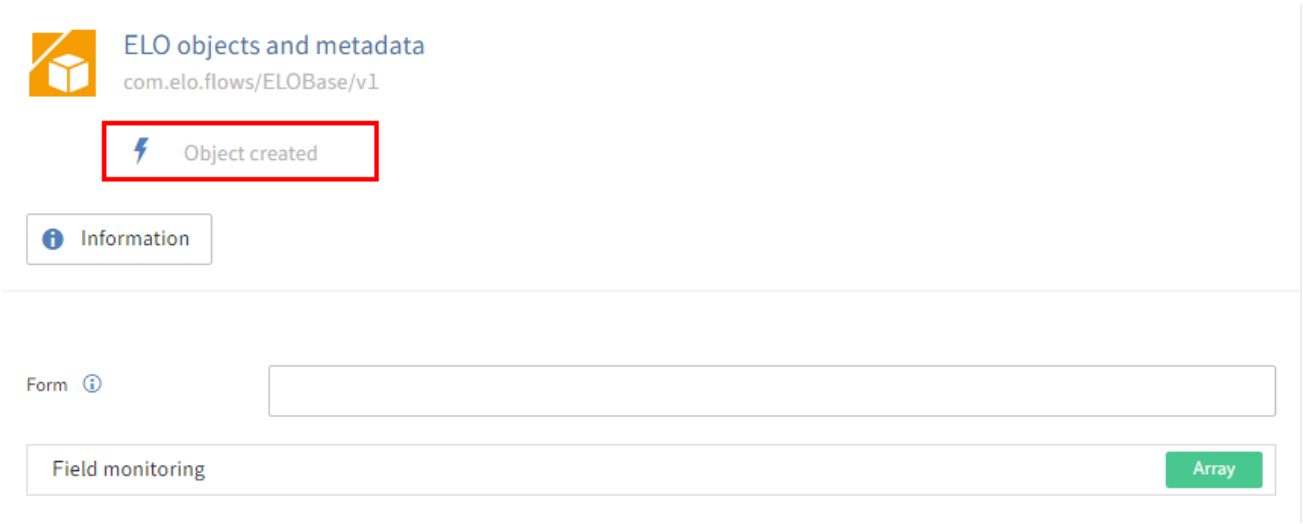
5 Column data: For further configuration, choose between the input modes Positions, Basic entry, and List transformation.

> **Information**
>
> Advanced configuration of the list transformation is optional. In this case, the *Context* field is mandatory.

7 Context: Enter the data context of the transformation, e.g. with relation to the previous *steps* within your flow.

8 string: Enter the value you want to select.

'Object created' trigger configuration



The *Object created* trigger starts when an ELO object (*SORD*) is created.

Advanced settings

Optional: You can restrict the trigger to specific metadata forms and data fields that are in the context of the created object.

> **Information**
>
> You can use gen. 1 and gen. 2 ELO metadata. For more information on how to use metadata in ELO, refer to General information > Using gen. 1 and gen. 2 metadata.

1 Form: Enter the metadata form the trigger listens to. The trigger is only started for this metadata form

2 Field monitoring: Enable field monitoring if you only want to start the trigger within the context of specific metadata fields.

3 Check mode: Choose between the check modes *Field corresponds to a comparison value* and *Field was changed*.

- *Field corresponds to a comparison value* - triggers the flow if a specific value is set.
- *Field was changed* - triggers the flow only if a specific field has been changed.

4 Field name: Enter the date field you want to monitor.

5 Field value: If you select the check mode *Field corresponds to a comparison value*, enter the value to be checked.

Example entry

Form: `Invoice` (gen. 1 metadata form) or `BASIC.PERSON` (gen. 2 metadata form)

Check mode: *Field corresponds to a comparison value*

Field name: `INVOICE_NO` (index field) or `INVOICE_POSITION.STATUS` (aspect field)

Field value: `12345`

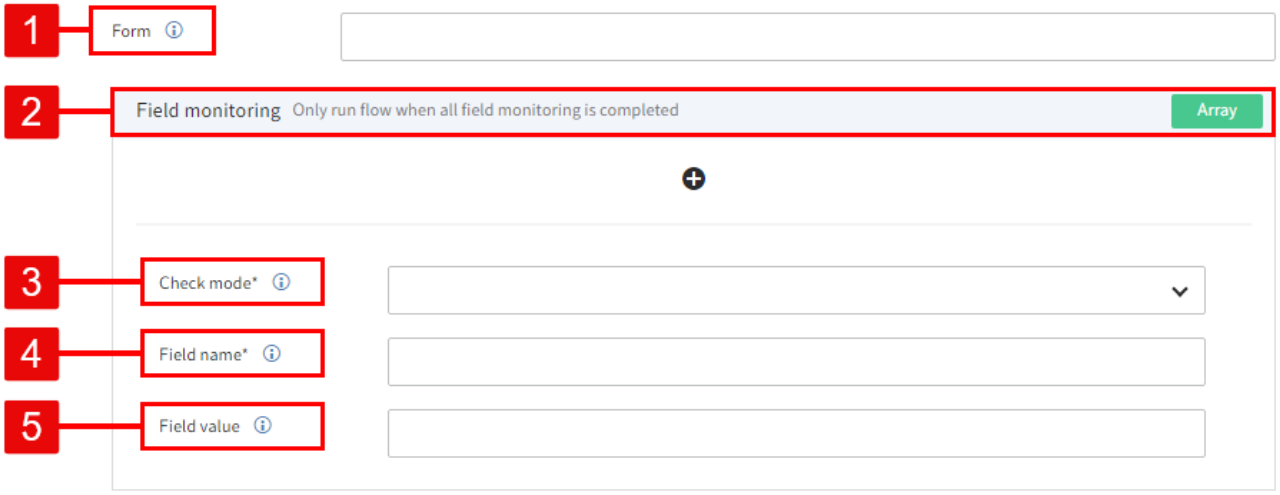
Next steps

The trigger *Object created* is an asynchronous flow so you don't have to configure an end node.

'Object edited' trigger configuration

The *Object edited* trigger starts when an ELO object (*SORD*) is edited.

Advanced settings



Optional: You can restrict the trigger to specific metadata forms and data fields that are in the context of the edited object.

> **Information**
>
> You can use gen. 1 and gen. 2 ELO metadata. For more information on how to use metadata in ELO, refer to General information > Using gen. 1 and gen. 2 metadata.

1 Form: Enter the metadata form the trigger listens to. The trigger is only started for this metadata form

2 Field monitoring: Enable field monitoring if you only want to start the trigger within the context of specific metadata fields.

3 Check mode: Choose between the check modes *Field corresponds to a comparison value* and *Field was changed*.

- *Field corresponds to a comparison value* - triggers the flow if a specific value is set.
- *Field was changed* - triggers the flow only if a specific field has been changed.

4 Field name: Enter the date field you want to monitor.

5 Field value: If you select the check mode *Field corresponds to a comparison value*, enter the value to be checked.

Example entry

Form: `Invoice` (gen. 1 metadata form) or `BASIC.PERSON` (gen. 2 metadata form)

Check mode: *Field was changed*

Field name: `INVOICE_NO` (index field) or `INVOICE_POSITION.STATUS` (aspect field)

Field value: `12345`

Next steps

The trigger *Object edited* is an asynchronous flow so you don't have to configure an end node.

Services

Services are actions that can be run within a flow. The *ELO objects and metadata* component offers the following services:

'Permissions' group

Name	Function
Remove all permissions	Removes all permission entries from an object.
Remove permissions	Removes permissions from an object.
Add permissions	Adds permissions to an object.

'Documents' group

Name	Function
Download working version	Downloads a version of an object and provides it in flows.
Upload document version	Loads a flow file in an object as a document version.

'Links' group

Name	Function
Create ECD link	Creates a new ECD link based on an object ID of a document.
Create external link	Creates a new external link based on an object ID.

'Metadata' group

Name	Function
Load metadata	Reads the metadata of an object and provides it in the flow.
Change color	Sets the color of an object.
Change metadata form	Sets the metadata form of an object.
Set metadata	Sets defined metadata on an object.

'New' group

Name	Function
New documents	Creates several new documents.

Name	Function
New document	Files a new document to the repository.
New folder	Creates a new folder in the repository.

'Organize' group

Name	Function
Delete all references	Deletes all references to an object.
Copy	Copies an ELO object and files the copy.
Delete	Deletes an ELO object.
Move	Moves an entry in the repository.
Reference	Creates a reference to an ELO object in the repository.

'Additional metadata' group

Name	Function
Load map fields as list	Loads map fields of an object as a list.
Load map fields	Loads map fields of an object and provides them in flows.
Write map fields	Writes values to map fields for an object.

Without group assignment

Name	Function
Load configuration file	Loads a configuration file from the ELO repository.

ELO feed

Goal and use

The *ELO feed* component offers basic functions for creating and editing feed entries.

Credentials and connection data

Configure connection data for the *ELO feed* component. A *default connection* is available. This is entered in the config files. The *user context* assumes the session of the user logged into the ELO client. You can also create additional connections to run services with defined rights.

Triggers

Triggers are the starting points of a flow. The *ELO feed* component does not offer any triggers.

Services

Services are actions that can be run within a flow. The *ELO feed* component offers the following services:

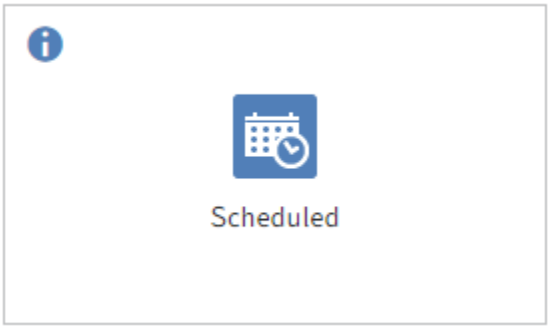
Name	Function
Create post	Writes a feed entry to an object.

Use

The *Create post* service accepts a string as a parameter. Use the following format to use a hashtag or mention:

- #[HASHTAG]
- @[User]

Time



Goal and use

The *Time* component allows you to schedule start times for flows.

Credentials and connection data

This component does not require credentials.

Triggers

Triggers are the starting points of a flow. The *Time* component offers the following triggers:

Name	Event
Scheduled trigger	Started via cron expression.

Use:

Define when to start the flow via a cron expression. ELO Flows uses the Spring cron format with a seconds field. This expression has six fields.

Structure

	Second	Minute	Hour	Day of month	Month	Day of week
Mandatory	YES	YES	YES	YES	YES	YES
Permissible values	0-59	0-59	0-23	1-31	1-12 or JAN-DEC	1-7 or SUN-SAT\
Permissible characters	, - * /	, - * /	, - * /	, - * ? / L	, - * /	, - * / L

> **Please note**
>
> Fields cannot be allocated twice. Double allocation such as `0 * * * * *` does not work here.

You can increase the complexity of the cron expression with the following characters:

Character	Meaning	Example
*	All values	* in the minutes field = Every minute
?	No specific value	`0 0 * ? * *` = Every hour, no matter the day
-	Range between two values	`0 0 10-12 ? * *` = Every hour between 10:00 and 12:59
,	Additional values	`0 10,44 14 ? * *` = At 14:10 and 14:44
/	Incremental values	`0 0/5 * * * ?` = Every 5 minutes
L	Last value	`* * * L * ?` = Every second on the last day of the month

Examples

Cron expression	Meaning
`0 0 * ? * *`	Every hour
`0 * * ? * *`	Every minute
`0 * 14 * * ?`	Every minute, between 14:00 and14:59
`0 0/5 14 * * ?`	Every 5 minutes, between 14:00 and 14:59
`0 */5 20-23,0-7 ? * 1-6`	Every 5 minutes at 20:00 until 23:59 and 00:00 until 07:59, Monday to Saturday
`0 10,44 14 ? 3 WED`	At 14:10 and 14:44, only every Wednesday, only in March
`0 15 10 ? * 6L`	At 10:15, on the last Saturday of the month
`0 11 11 11 11 ?`	At 11:11, on day 11 of the month, only in November
`0 0 12 1/5 * ?`	At 12:00, every 5 days
`0 15 10 L-2 * ?`	At 10:15, 2 days before the last day of the month
`0 15 10 L * ?`	At 10:15, on the last day of the month
`0 15 10 15 * ?`	At 10:15, on day 15 of the month

Services

Services are actions that can be run within a flow. The *Time* component does not offer any services.

ELO workflow

Goal and use

The *ELO workflow* component offers functions for starting an ELO workflow from within a flow. Flows can also be triggered by workflows.

You will find information on ELO workflows and using the workflow designer under Administrators > Processes and automation > ELO workflow.

Credentials and connection data

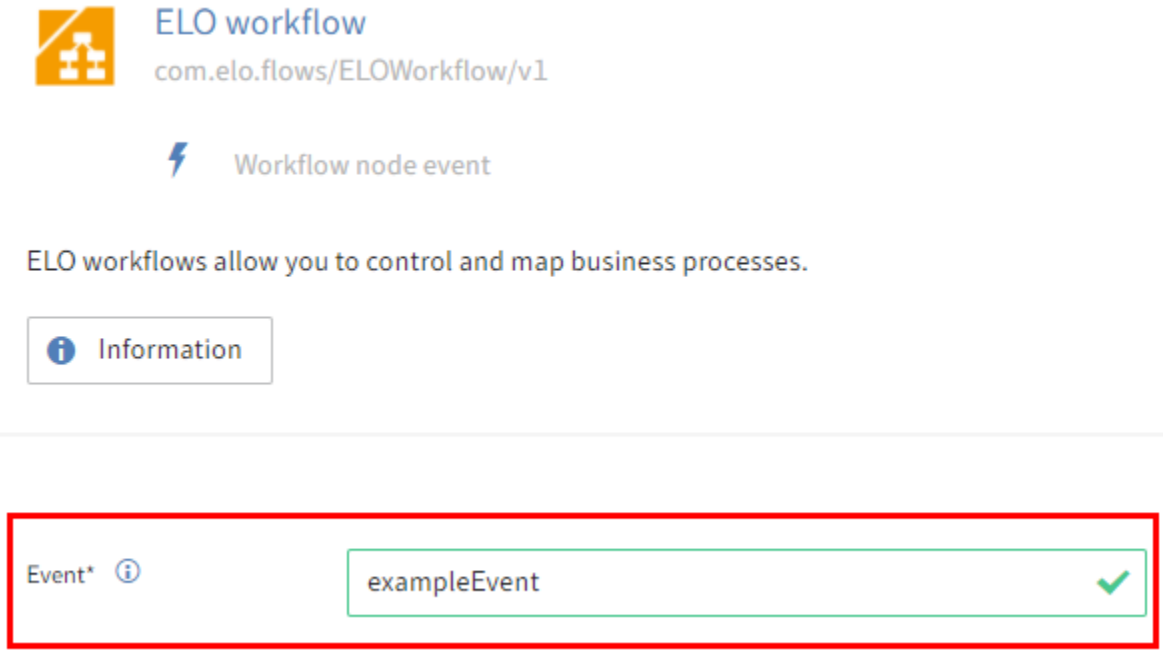
Configure connection data for the *ELO workflow* component. A *default connection* is available. This is entered in the config files. The *user context* assumes the session of the user logged into the ELO client. You can also create additional connections to run services with defined rights.

Triggers

Triggers are the starting points of a flow. The *ELO workflow* component offers the following triggers:

Name	Event	Additional information
Workflow node event	This trigger can be used to start a flow from a workflow node.	Synchronous flow
Workflow node direct call	This ensures that only the defined flow is started.	Synchronous flow

'Workflow node event' trigger usage



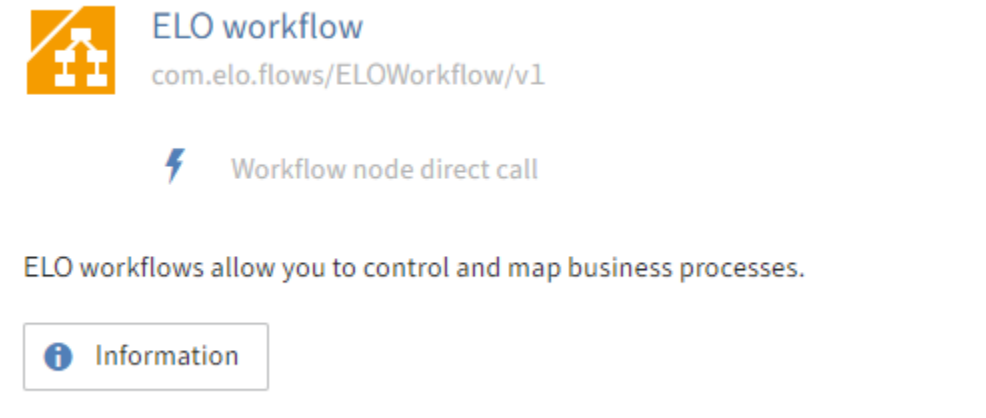
In the flow designer, select the trigger *Workflow node event*. Under *Event*, enter a name. For example, *exampleEvent*.

In the ELO workflow designer, select a node you want to trigger a flow with.

The ELO workflow component enables integration of ELO Flows in ELO workflows.

Additional information regarding the use of flow nodes in the workflow designer can be found under ELO workflow > Default workflow > Flow node.

'Workflow node direct call' trigger usage



In the ELO workflow designer, select a node you want to trigger a flow with.

Services

Services are actions that can be run within a flow. The *ELO workflow* component offers the following services:

Name	Function
Start workflow	Starts the defined workflow on an ELO object
Start workflows	Starts each defined workflow on multiple transferred ELO objects
Read workflow map fields	Reads map fields of a workflow
Write workflow map fields	Writes values to map fields of a workflow

SMTP Mail

Goal and use

The *SMTP Mail* component allows you to send e-mails (as HTML or text) with the help of a mail server.

Credentials and connection data

This component requires access to an SMTP server.

Triggers

Triggers are the starting points of a flow. The *SMTP Mail* component does not offer any triggers yet.

Services

Services are actions that can be run within a flow. The *SMTP Mail* component offers the following services:

Name	Function
Send e-mail	Sends a text e-mail.
Send e-mail (HTML)	Sends an HTML e-mail (this service allows you to apply different formats using HTML tags in the message text).

FTP

FTP

Goal and use

The *FTP* component provides functions for communicating with an FTP server. It enables you to perform folder and file operations, such as uploading and downloading documents to and from an FTP server or loading data and metadata.

Credentials and connection data

Configure connection data for the *FTP* component. Each connection entry stands for a connection to the FTP server. Different FTP protocols can be used: insecure FTP, FTPS, and SFTP.

Triggers

Triggers are the starting points of a flow. The FTP component does not offer any triggers.

Services

Services are actions that can be run within a flow. The *FTP* component offers the following services:

'Document' group

Name	Function
Download document	Downloads a document from an FTP server. This allows the document to be used for other services.
Load metadata for document	Reads the metadata of a document on an FTP server. This allows the available metadata, such as the path, size, and timestamp to be used for other services.
Upload document	Stores a document on an FTP server.

'Folder' group

Name	Function
Create directory	Creates a new folder on an FTP server. It is also possible to create multiple child directories.
Load metadata for folder	Reads the metadata of a folder on an FTP server. This allows the available metadata, including path, size, and timestamp to be used for other services.

Name	Function
Load directory contents	Reads the content of a folder. You can also restrict or expand the selection with different filters and options.

'Organize' group

Name	Function
Copy	Copies a document or folder to an FTP server. This creates a duplicate, which may be stored at a different location on the FTP server.
Delete	Removes a document or folder on an FTP server. All contents in the folder or document will also be deleted permanently. You cannot delete the root directory of the FTP server.
Move	This service moves a document or a folder on an FTP server to another location on the FTP server. You cannot move the root directory of the FTP server. You also have the option to simultaneously rename the document/folder.
Rename	Renames a document or folder on an FTP server. The typical name restrictions apply.