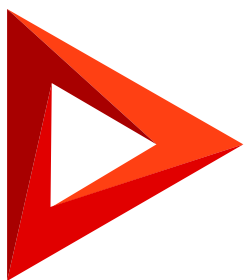


Parameters

Use process parameters

Version 7.18



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Table of Contents

Use process parameters	4
Specify static and variable parameter values	4
Pass parameters between process elements	6
Pass parameters between business processes	8
Add a Collection of records process parameter	13

Use process parameters

PRODUCTS: ALL CREATIO PRODUCTS

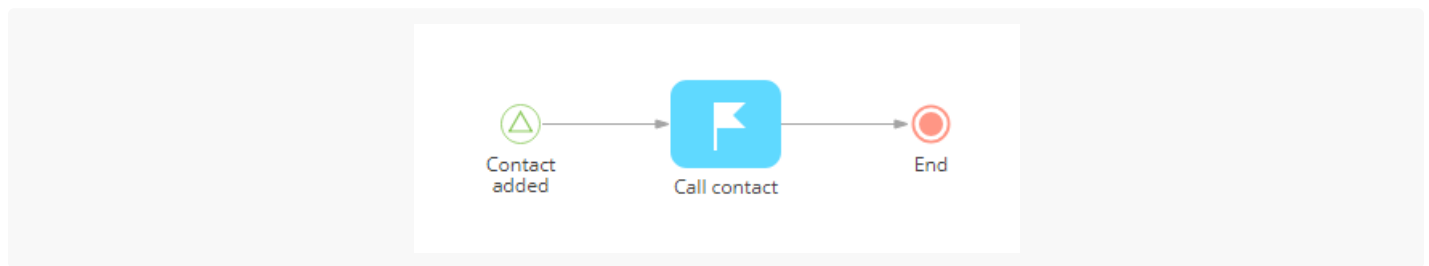
Specify static and variable parameter values

Static parameter values are specified directly when the process is designed and are always the same in all process instances. The parameters are populated in the same way as section record fields of the corresponding type (text, numeric, date, etc.).

Certain parameters allow you to select variables, such as [system settings](#), current date and time or current user contact as their values.

For example, when populating the [*Who performs task*] parameter of the [*Perform task*] element, you can select a specific contact, e.g. "John Smith", or the "Current user" variable, which will populate the parameter with the user who runs the current instance of the business process (Fig. 1).

Fig. 1 Task scheduling process

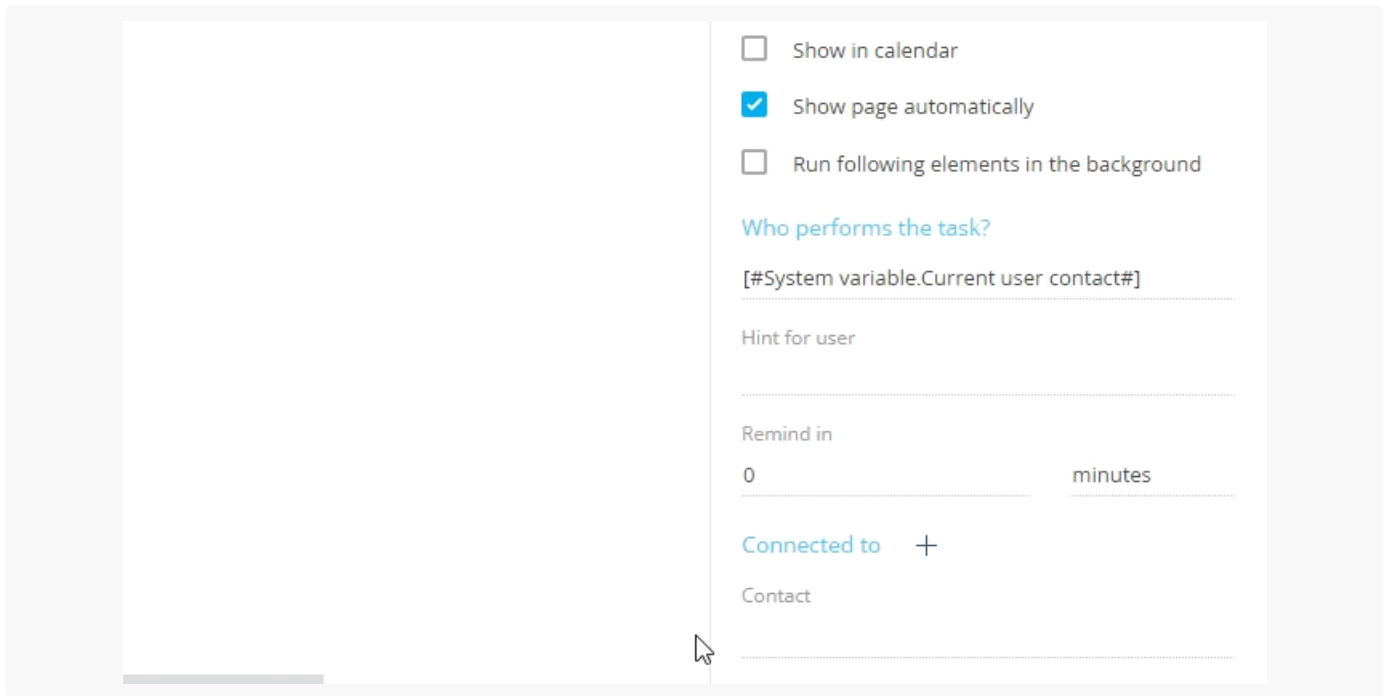


Note. You can also map parameter values to parameters of other elements or [process parameters](#).

To specify a static parameter value:

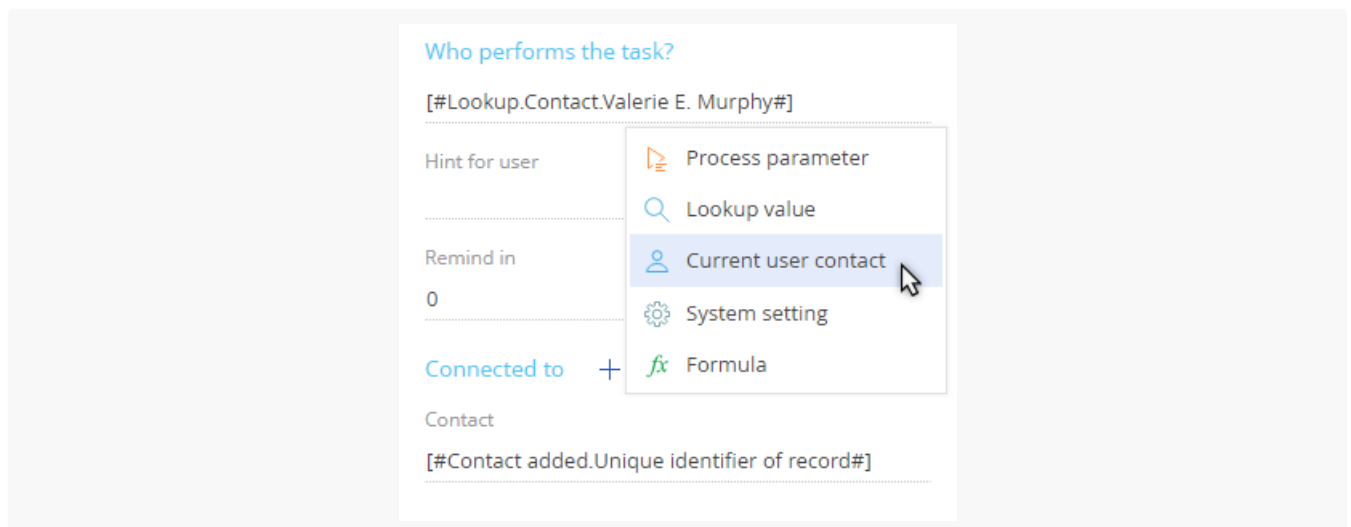
1. Select a process element where the static value should be.
2. Click the ⚡ button next to the field of the parameter. Select one of the following options depending on the desired outcome:

Fig. 2 Selecting a lookup value



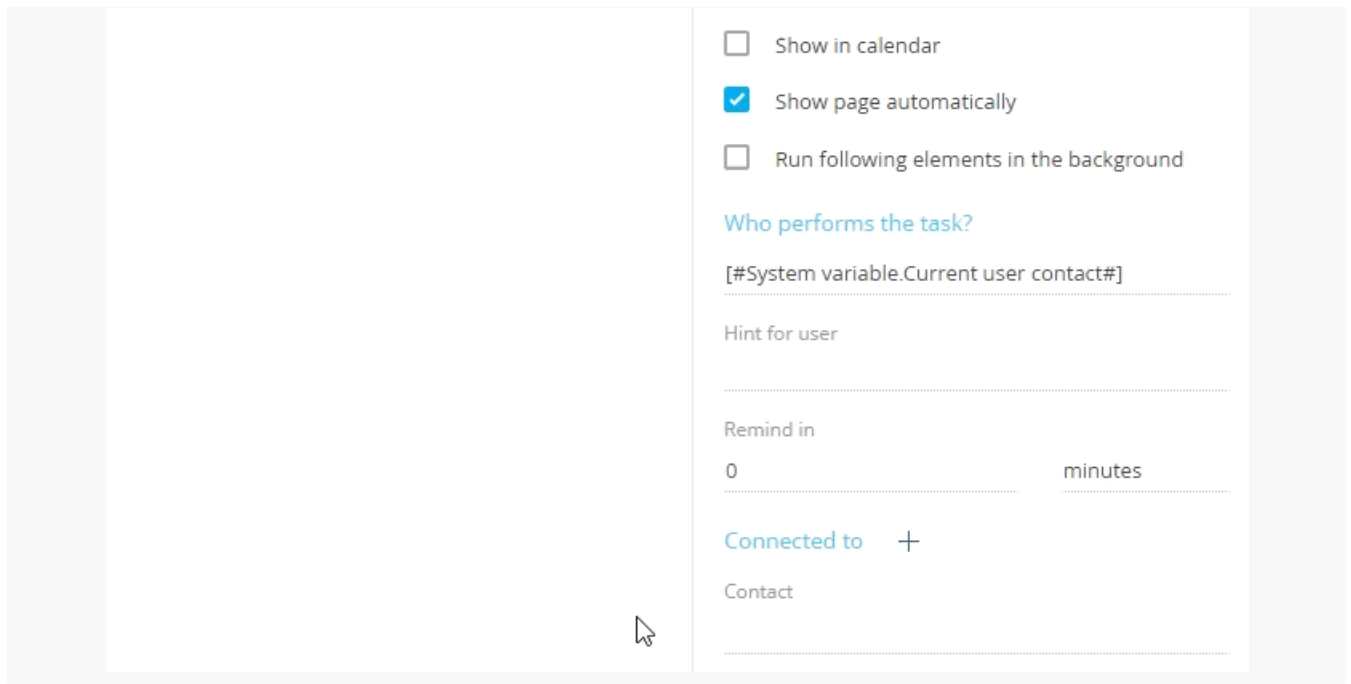
- a. Select [*Lookup value*] if the value has to be populated with lookup data. For example, click the [*Contact*] field in the [*Perform task*] element and select the contact responsible for the task from the list (Fig. 2).
- b. Select a variable, e.g., [*Current user*] or [*Current date*] if the value has to be populated with / for the user who runs the process or date when the element is executed. For example, select [*Current user contact*] option (Fig. 3) in the [*Who performs the task?*] field of the [*Perform task*] element if the task has to be created for the user who started the business process.

Fig. 3 Selecting the current user contact option



- c. Select [*System setting*], and then select the system setting whose value must be written to the parameter when the process is executed. The data type of the selected system setting value must match the data type of the parameter (text, numeric, date, etc.). (Fig. 4).

Fig. 4 Selecting the system settings option



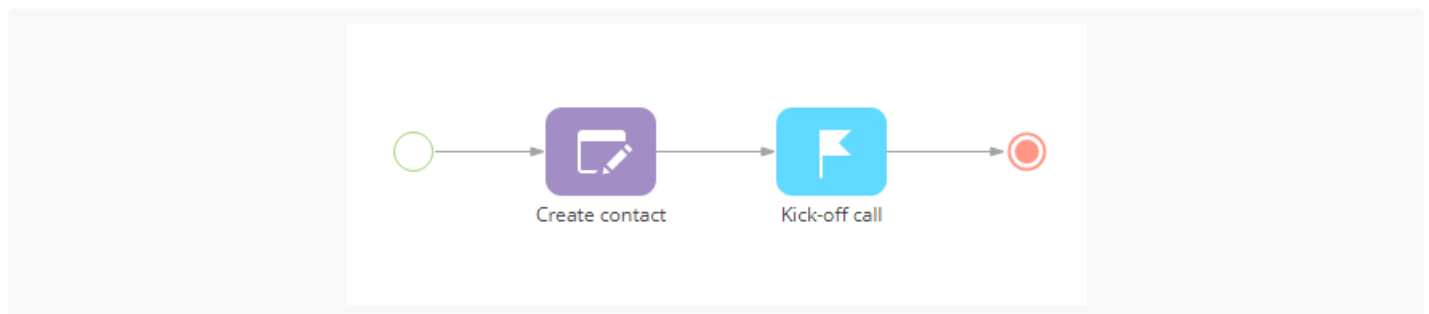
3. Save the process diagram.

Pass parameters between process elements

Process element parameters determine the behavior of process elements. For example, the [*Perform task*] element contains such parameters as [*Subject*], [*Owner*], [*Duration*], etc., which determine the properties of the “Task” activity that the element creates.

You can assign parameter values manually or map them to other process or element parameters. For example, while scheduling a kick-off call with new contacts, you can pass the Id of the contact record created earlier in the process flow to the [*Contact*] parameter of a subsequent [*Perform task*] element. As a result, the “Task” activity will be connected to that contact (Fig. 4).

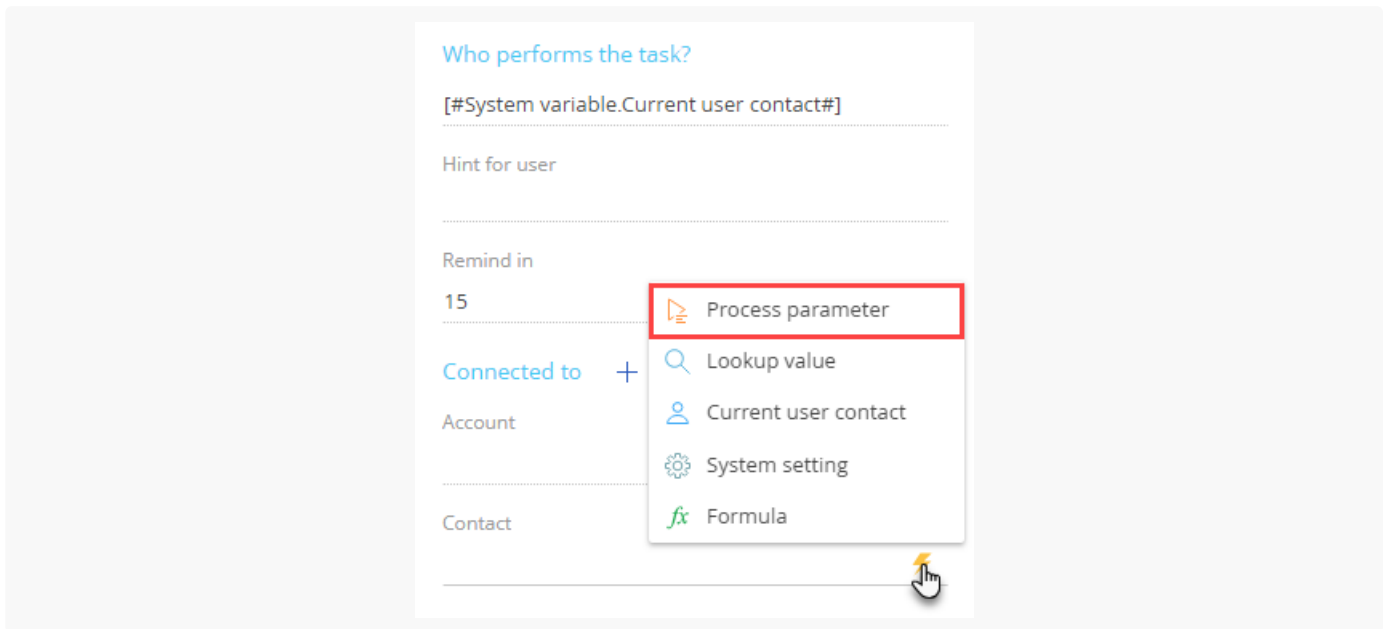
Fig. 4 Simple task scheduling process



To pass parameters:

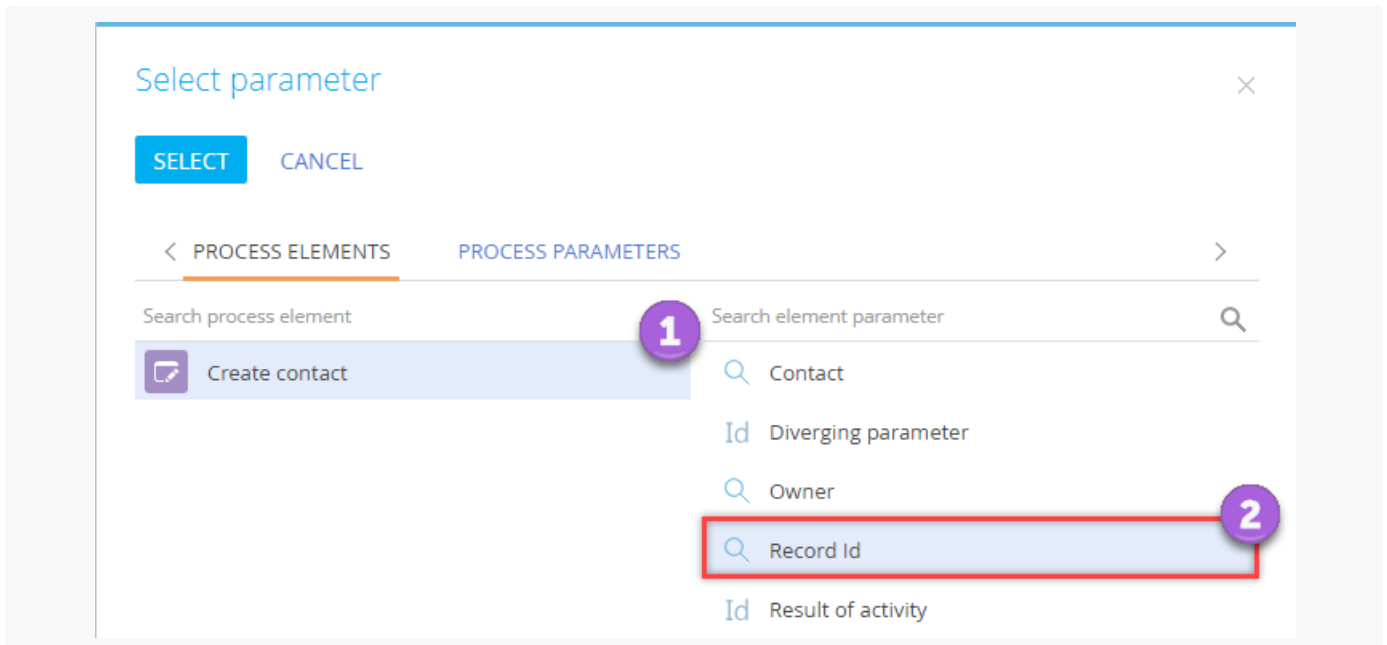
1. Select the element, to which the parameter value will be passed. For example, to pass the contact to the [*Perform task*] element, select it on the diagram.
2. In the element setup area, click the ⚡ button next to the field of the parameter, which must be populated with the value from another element, and select [*Process parameter*] from the menu (Fig. 5). For example, click the [*Contact*] field in the [*Perform task*] element.

Fig. 5 Opening the parameter selection window



3. In the window that appears, click [*Process elements*].
4. Select an element which contains the necessary parameter (1). In the list of element parameters (2), select the parameter whose value must be obtained. In this case, Id of the contact will be saved in the [*Record Id*] parameter of the [*Open edit page*] element (Fig. 6).

Fig. 6 Selecting the Id parameter



5. Save the process diagram.

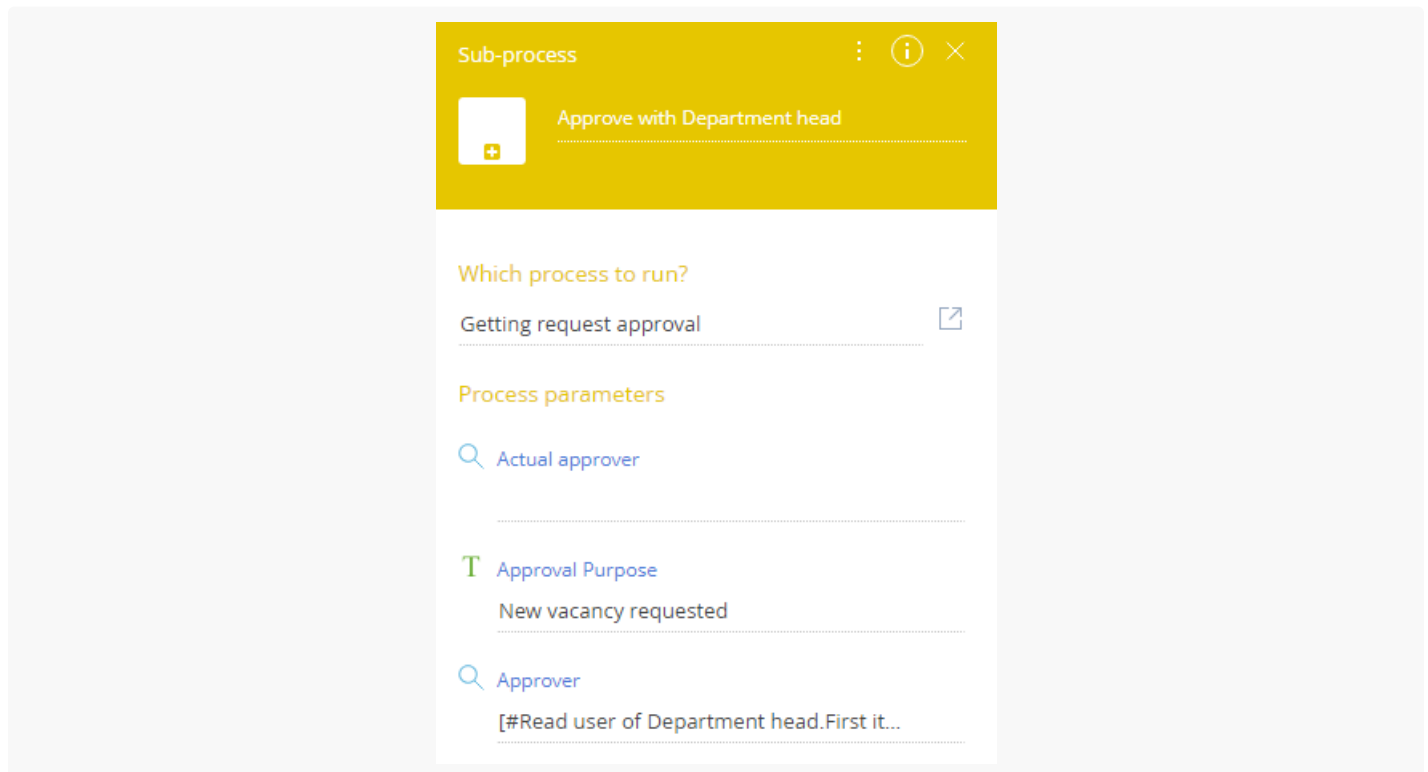
As a result, whenever the process is run, the value of the element parameter will be populated automatically from the corresponding parameter of a different element. For example, the [*Contact*] parameter of the [*Perform task*] element will be populated with the contact, which was created when the [*Open edit page*] element was executed.

Pass parameters between business processes

Creatio business processes can exchange information with each other using process parameters. Such exchange is possible between a sub-process and corresponding parent processes only.

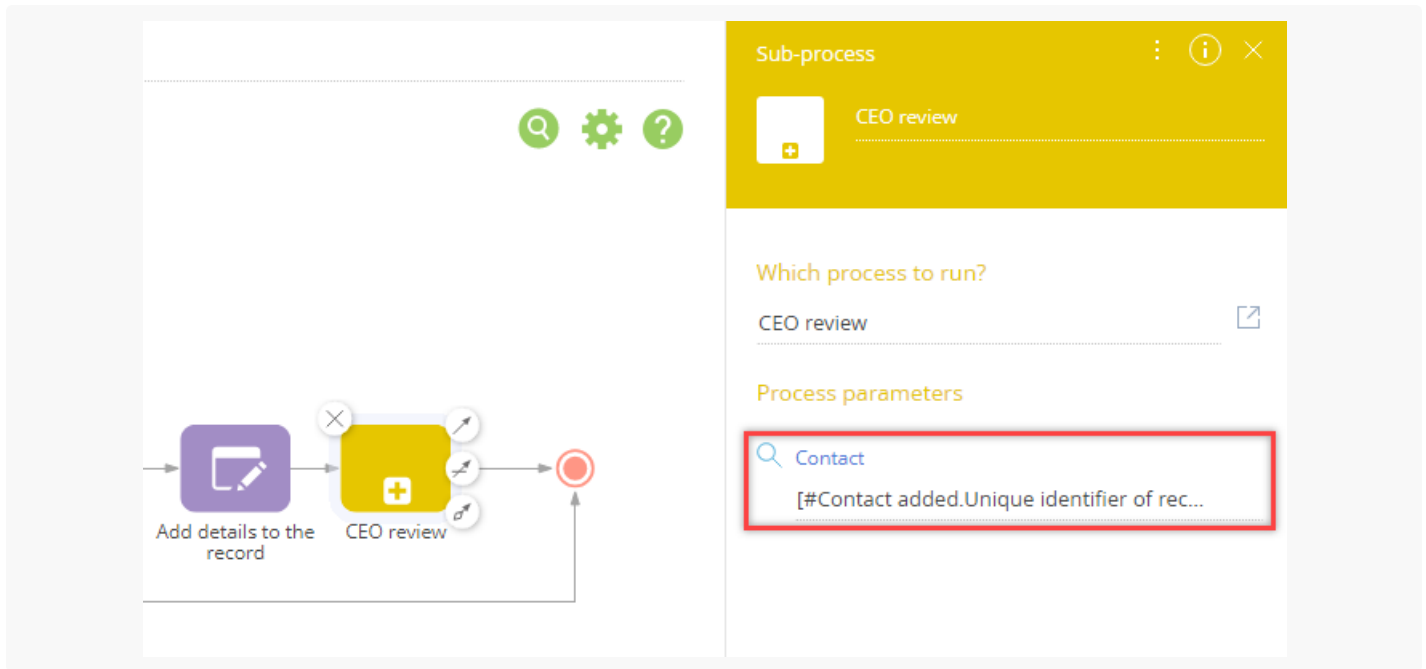
Parameters of the subordinate process become parameters of the corresponding [*Sub-process*] element of the parent process. When you populate the [*Which process to run?*] field of the [*Sub-process*] element, its setup area displays the parameters of the selected process (Fig. 7). The values of these fields can then be populated with static values or mapped to other parameters from the parent process. Likewise, parameters of the parent process and its elements can be mapped to parameters of the [*Sub-process*] element.

Fig. 7 An example of process parameters displayed in the setup area of the [*Sub-process*] element



For example, a process for job applicant review includes a sub-process, during which the applicant is reviewed by the company management. You can pass the contact's Id to the sub-process using the [*Lookup*] process parameter (Fig. 8).

Fig. 8 Mapping sub-process parameters



To transfer parameter values between a sub-process and its parent process:

1. Add corresponding parameter(s) to the sub-process.
2. Map the parameters of the parent process and/or its elements to the parameters of the [*Sub-process*] element.

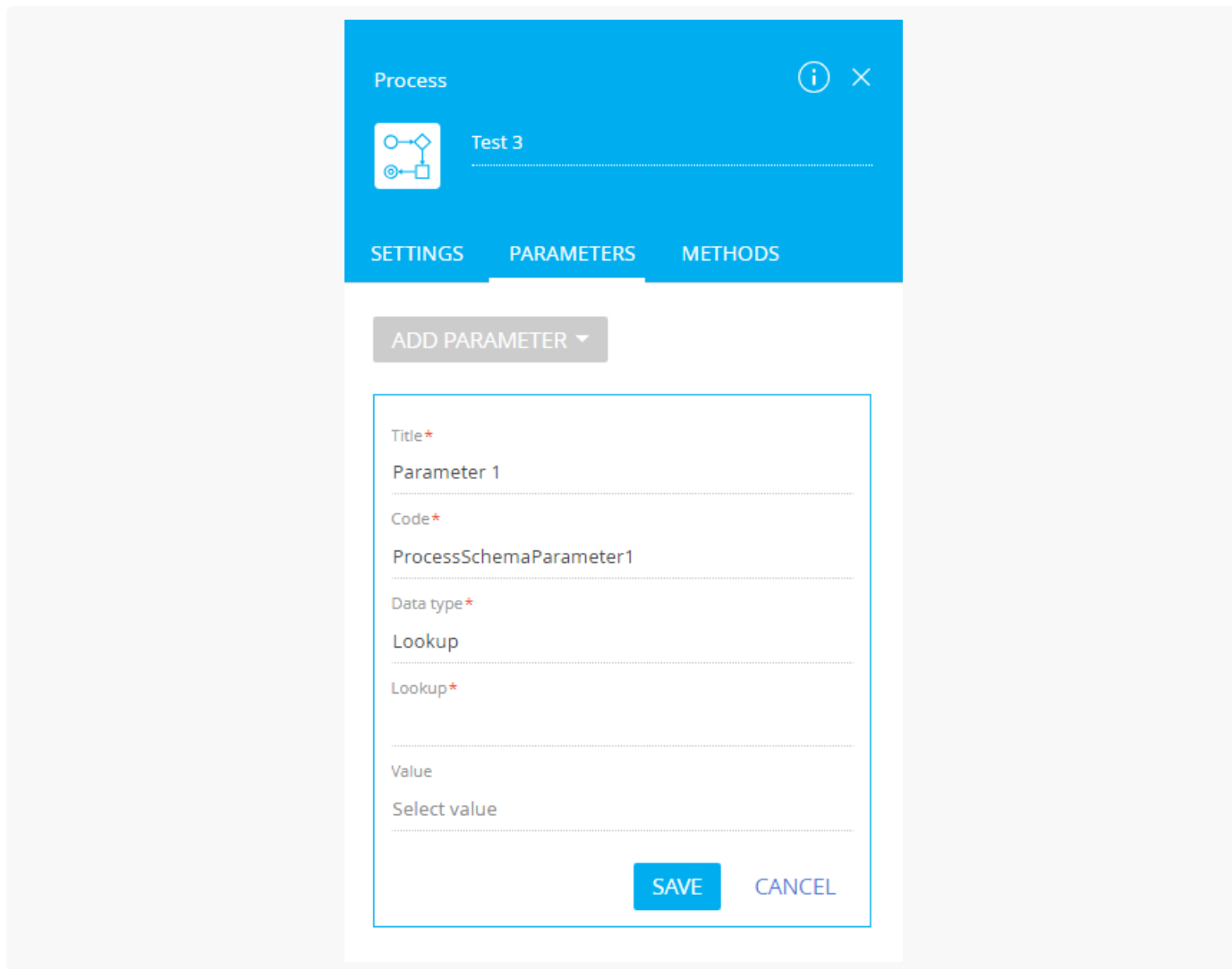
To add a parameter to the business process:

1. Open the sub-process in the Process Designer.
2. In the process setup area, go to the [*Parameters*] tab.
3. Click [*Add parameter*] and select its type. For example, if you need to pass a specific record as parameter value, you would need a [*Lookup*] type parameter.

Note. The record will be passed to this process, therefore this parameter will be populated with a value from the parent process. The setup of the parent process is described below.

4. Populate the [*Title*] and [*Code*] fields so that you can easily identify this parameter in the future, e.g., when mapping its values.
5. For a lookup parameter you need to specify the object whose records can be used to populate this parameter. In the [*Lookup*] field, select the object whose records will be used to populate parameter values. For example, if your parameter is supposed to link a process to a contact record, select the “Contact” (Fig. 9).

Fig. 9 Setting up a process parameter of the “Lookup” type



6. Save the parameter and the process.

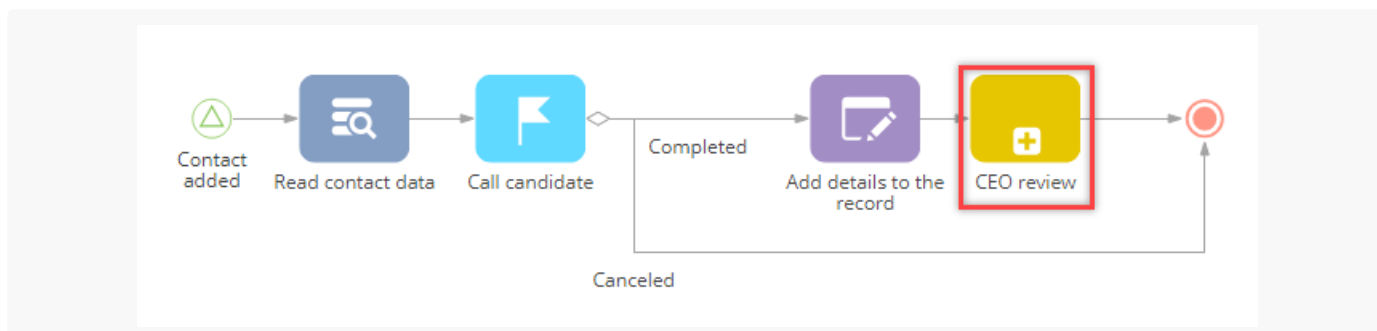
Once the lookup parameter is added to the business process, it has to be mapped in the parent process.

Mapping parameter values of parent process to parameters of a sub-process

To map the parent process parameter:

1. Add a [*Sub-process*] element to the process diagram (Fig. 10).

Fig. 10 Sub-process on the diagram

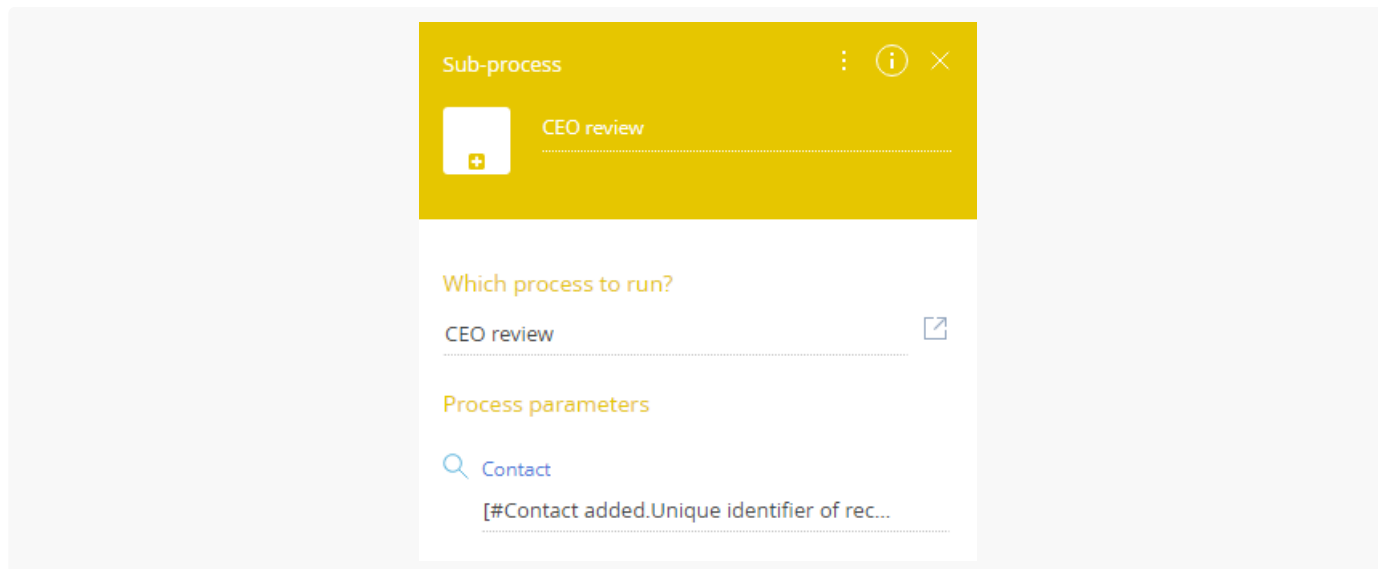


2. In the [*Which process to run*] field, select the subordinate process, which will be run as part of the current

process. For example, in the “Onboarding” process, add sub-process element and select the “CEO review” process as the sub-process.

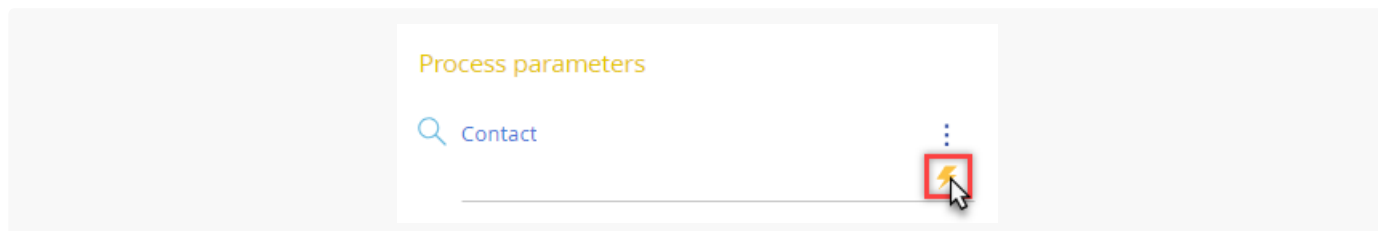
As a result, the parameters of the selected process will appear in the setup area of the [*Sub-process*] element. For example, the [*Contact*] parameter will appear once you select the “CEO review” process (Fig. 11).

Fig. 11 “CEO review” sub-process element parameters



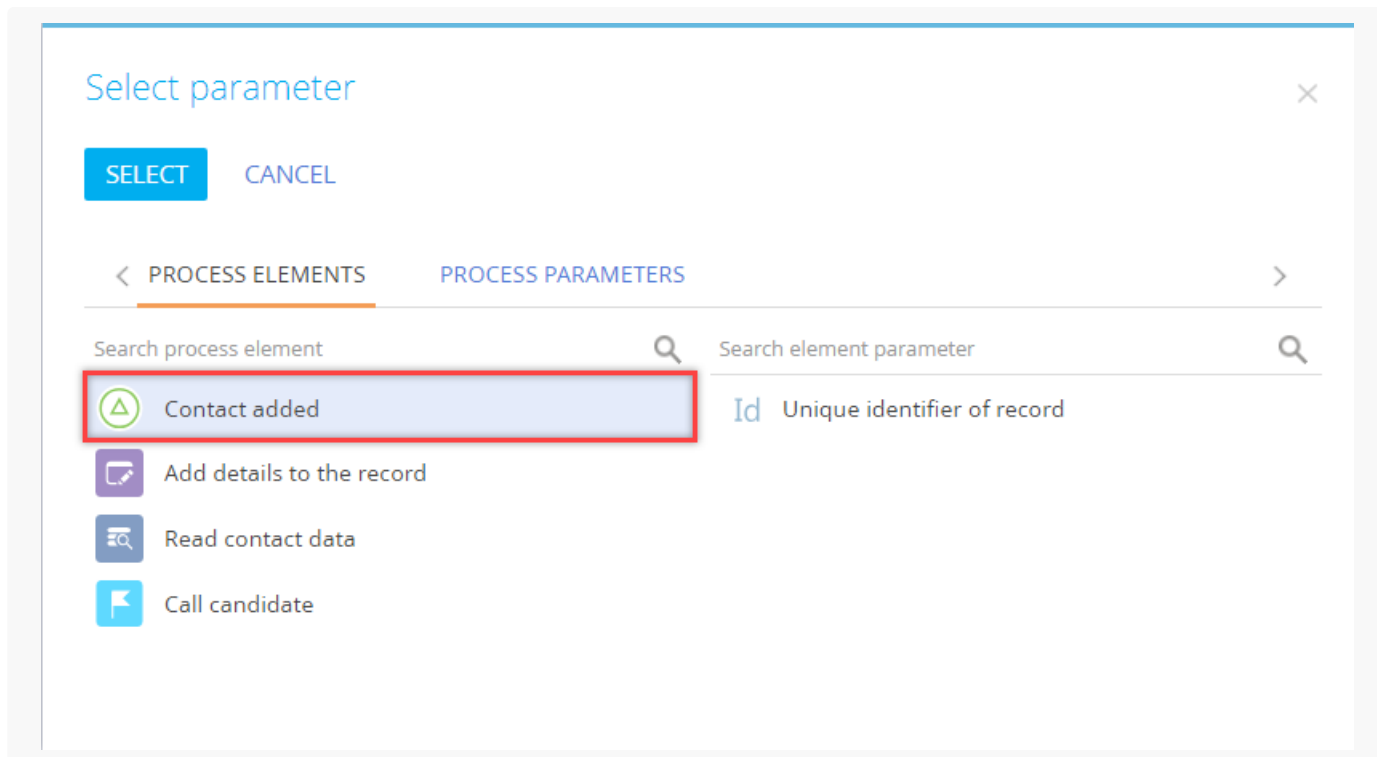
3. In the [*Process parameters*] area, click the ⚡ button next to the parameter to map (Fig. 12).

Fig. 12 Selecting the parameter to map



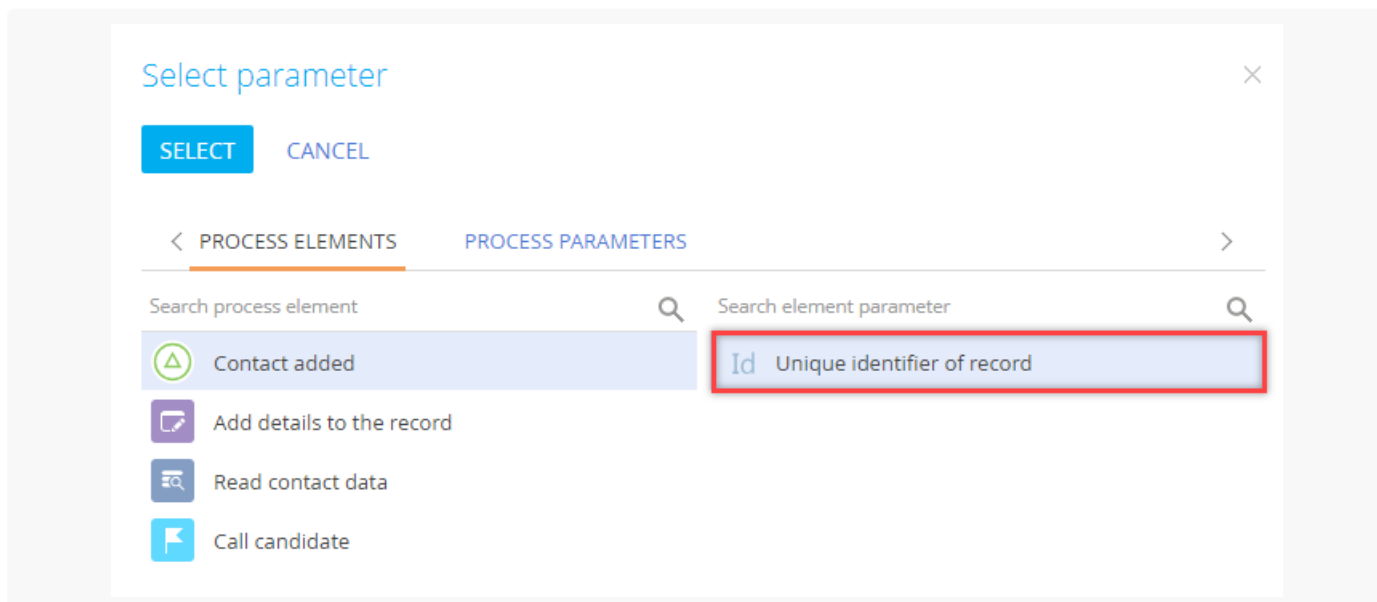
4. In the window that appears, select the element that contains the needed parameter. For example, if the parent process is triggered by an object signal and you need to map the parameter to the record that triggered it, select the corresponding signal element (Fig. 13).

Fig. 13 Selecting the element that contains the needed parameter



5. A list of the element parameters will appear to the right. Select the parameter that contains the value. For example, if you are mapping a lookup parameter to an object signal, select the [*Unique identifier of record*] parameter (Fig. 14).

Fig. 14 Mapping the parameter to the child process



6. Save the process diagram.

As a result, whenever the sub-process is executed, its parameters will obtain values from the parent process. For example, the contact that triggered the start of the parent “Onboarding” process will be passed to the child “CEO review” process.

Add a [*Collection of records*] process parameter

When working with collections, use the [*Collection of records*] parameter as a medium between the developer code and low-code elements in a business process. You can add a process parameter for storing collections of records and set up the structure of the collection by adding nested parameters. This way you can work with collections using low-code tools, such as [multi-instance sub-processes](#).

You can use the [*Collection of records*] process parameter when running business processes from a C# or JS code or when using the [\[*Script task* \] process element](#), e.g., for implementing integration with third-party applications. You can also use it to pass record collections between business processes.

To add the [*Collection of records*] process parameter:


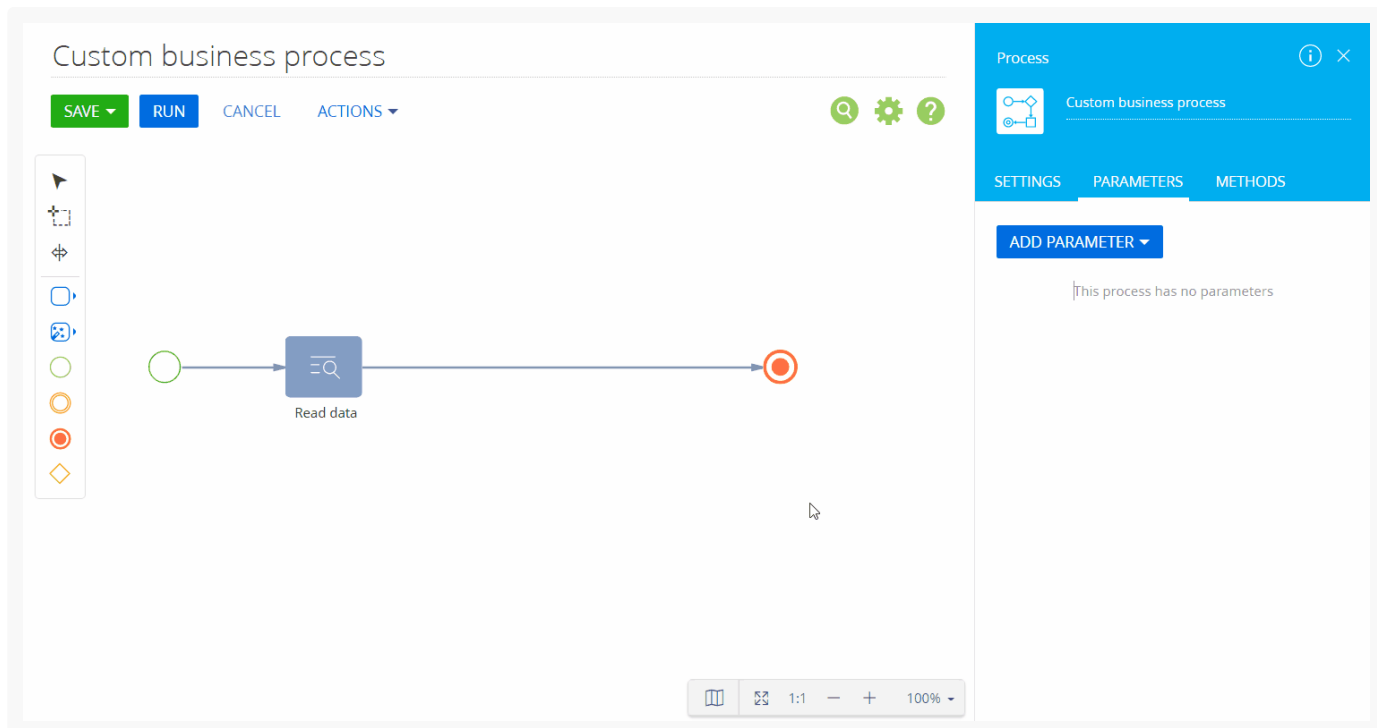
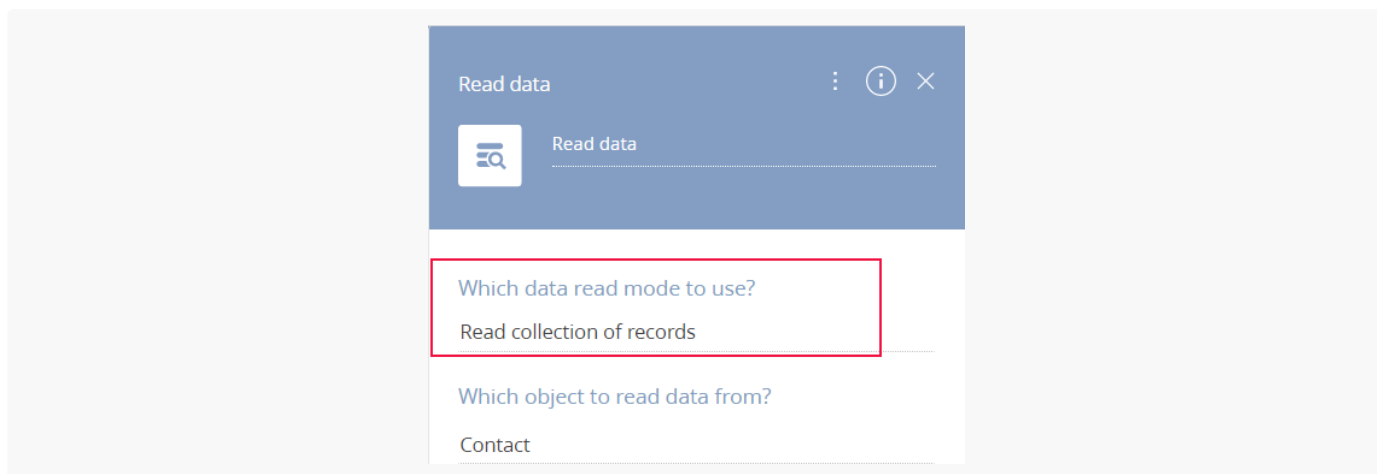
1. Click  → [*Process library*] section and open the needed business process in the Process Designer.
2. Go to the [*Parameters*] tab in the element setup area.
3. Click [*Add parameter*] → [*Other*] → [*Collection of records*].
4. In the window that opens, populate the process parameter standard fields:
 - a. [*Name*] – specify the parameter name, e.g., “Conact Collection.”
 - b. [*Code*] – specify the parameter code, e.g., “ProcessSchemaConactCollection.”
 - c. Leave the [*Value*] field unpopulated.
5. Click [*Save*].
6. Set up the nested parameters to define the collection structure. For example, to include the name and age for each record in a collection of contacts, add [*Full name*] and [*Age*] nested parameters:
 - a. Click [*Add nested parameter*] → select the needed parameter type, e.g. select the “Text” type for the “Name” field to map with the [*Full name*] column of the contact page.
 - b. Click [*Add nested parameter*] → select the needed parameter type for another field, e.g. select the “Integer” type for the “Age” field to map with the [*Age*] column of the contact page.
7. Map each of the nested parameters created at the previous step (in our case, two parameters created at step 5) to the corresponding nested parameters of a collection obtained in the process flow. Read more: [Use parameters](#).

Fig. 9 Adding the [*Collection of records*] process parameter in a business process



Note. To map the nested parameters of the [*Collection of records*] process element with corresponding values, specify the “Read collection of records” value in the [*Which data read mode to use?*] field of the process element you are using for mapping within a business process.

Fig. 10 Setting up the [*Read data*] element for reading a collection of records



As a result, you will create a parameter that can store a collection of records (e.g., contacts) and the corresponding data for each record in the collection (e.g., contact’s name and age). You can use the collection data further in the process and pass these parameter values to other business process elements.