

# Web service integration

Update currency exchange rates with web service integration

Version 7.17



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

|   |   |
|---|---|
| Update currency exchange rates with web service integration | 4 |
|---|---|

# Update currency exchange rates with web service integration

PRODUCTS: ALL CREATIO PRODUCTS

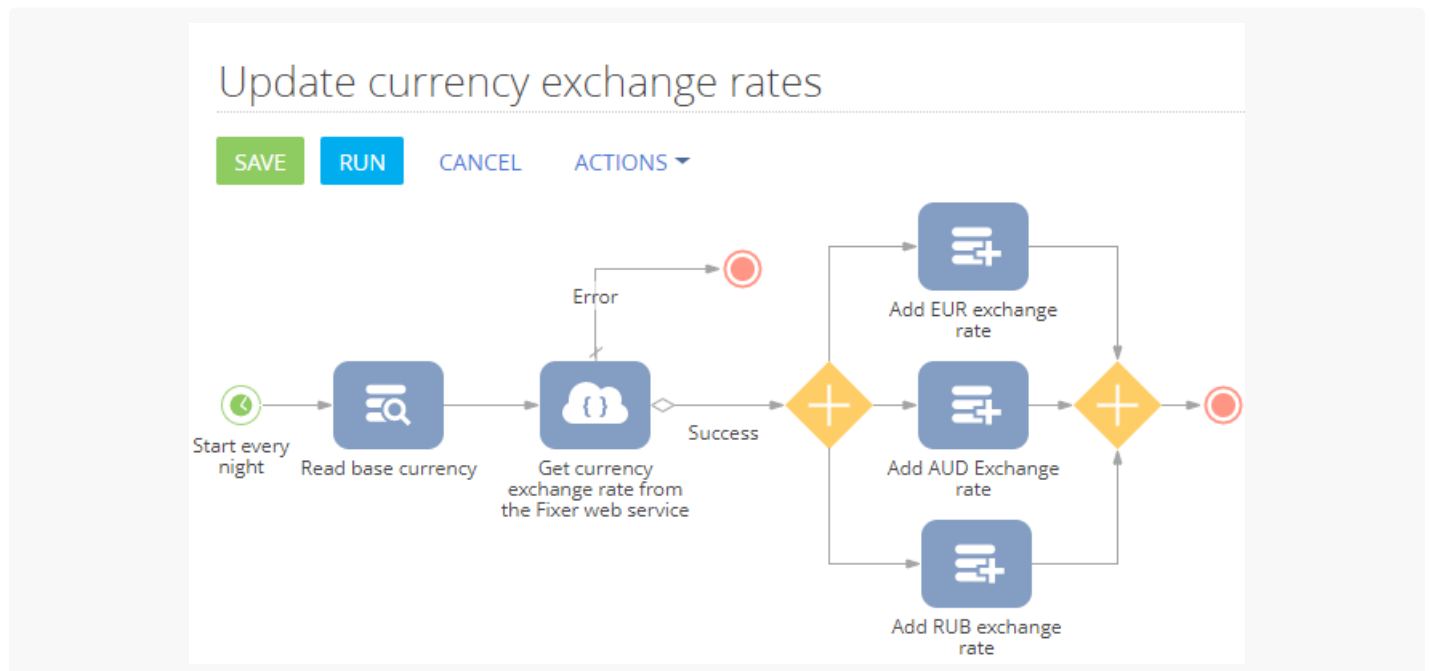
Creatio uses a special business process element to call web services and parse their response.

**Example.** Create a business process that will obtain currency exchange rates from the <http://api.fixer.io/> web service and update exchange rates on the [ *Currency rate* ] detail of the [ *Currencies* ] lookup.

**Attention.** Detailed instructions on setting up integration with the <http://fixer.io/> web service are available in a separate article.

Business process diagram (Fig. 1) elements:

Fig. 1 The “Update currency exchange rates” process



1. Business process custom [parameter](#): “Base currency”, which contains the current value of the [ *Base currency* ] system setting.
2. [Timer](#) start event: “Start every night” – the process starts daily, at the specified time.
3. [Read data](#) system action: “Read base currency” – the process obtains the current base currency name.
4. [Call web service](#) system action: “Get currency exchange rate from the Fixer web service” – the process calls web service request method “latest” to obtain exchange rates for the current base currency. The

element has 2 outgoing [conditional flows](#):

- a. "Success": the process proceeds to updating the exchange rates if response from the web service has been received.
  - b. "Error": the process will terminate if web service call resulted in error or timed out.
5. [\[ Add data \]](#) system actions: add one for each currency whose rate must be modified. For example, "Add EUR exchange rate" – The process adds a record on the [\[ Currency rate \]](#) detail for each currency whose rate must be updated. The [\[ Exchange rate \]](#) field for each record contains the corresponding exchange rate received from the web service.
  6. [\[ Parallel gateway AND \]](#): this gateway will ensure that the process will end only after all currency rates have been updated.

**Attention.** Before using the [\[ Call web service \]](#) element, make sure that you set up integration with the needed web service, using the [\[ Web services \]](#) section in the [\[ Studio \]](#) workplace. Learn more about adding the <http://fixer.io/> web service integration in the [Getting started with low-code web service integration](#) article.

To configure the process:


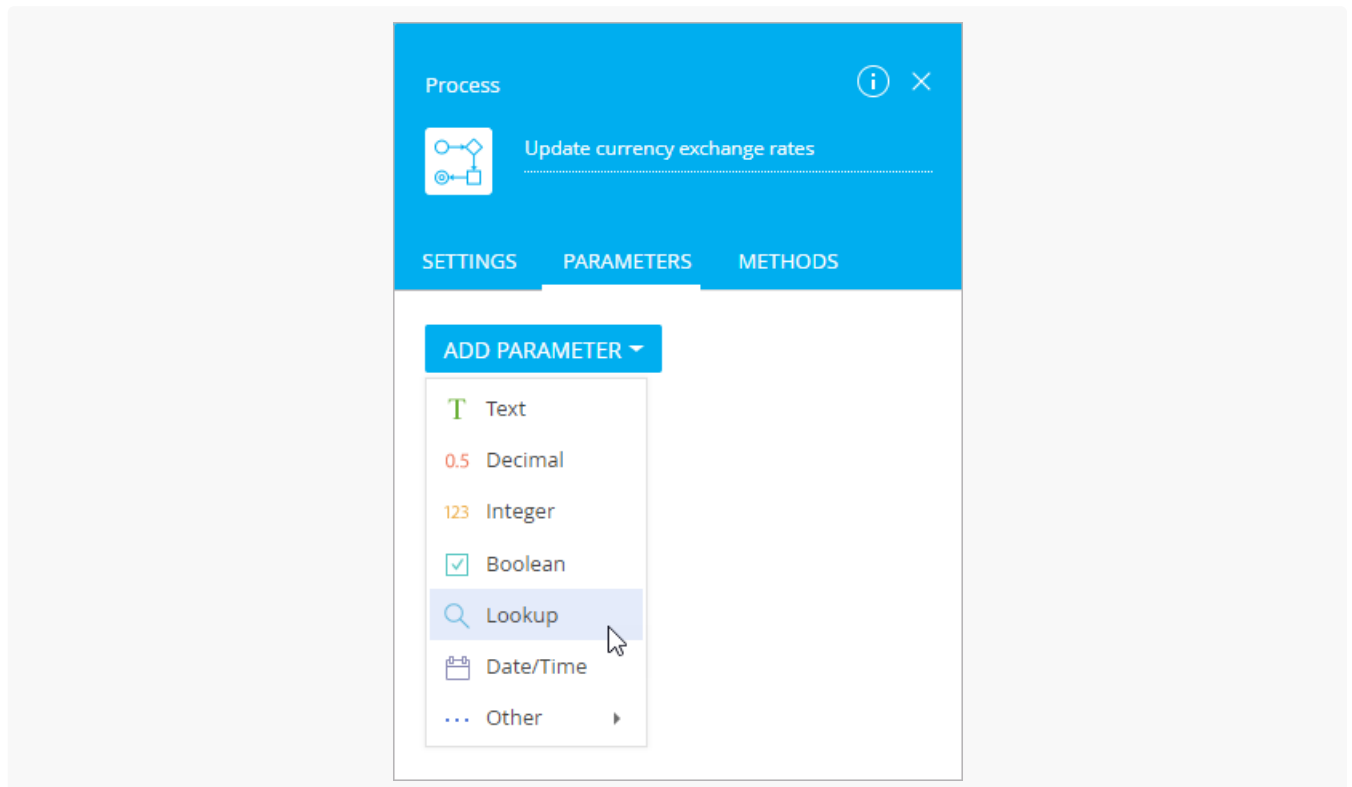
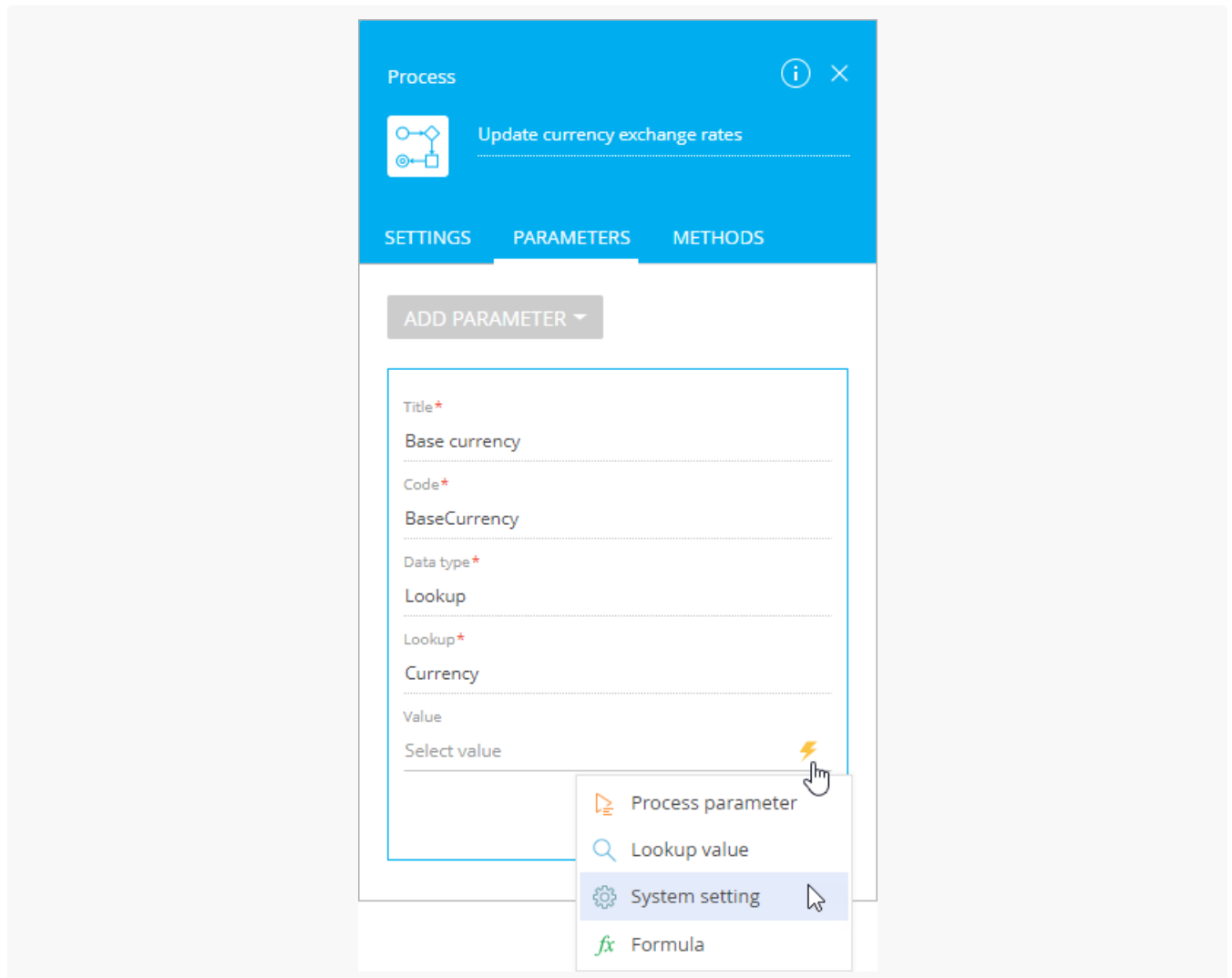
1. Use a custom parameter to pass the current "Base currency" system setting value to the business process.
  - a. Click , then click [\[ Parameters \]](#) tab and [\[ Add parameter \]](#) button. Select "Lookup" parameter type (Fig. 2).

Fig. 2 Add a custom "Lookup" type process parameter



## b. Populate parameter properties (Fig. 3):

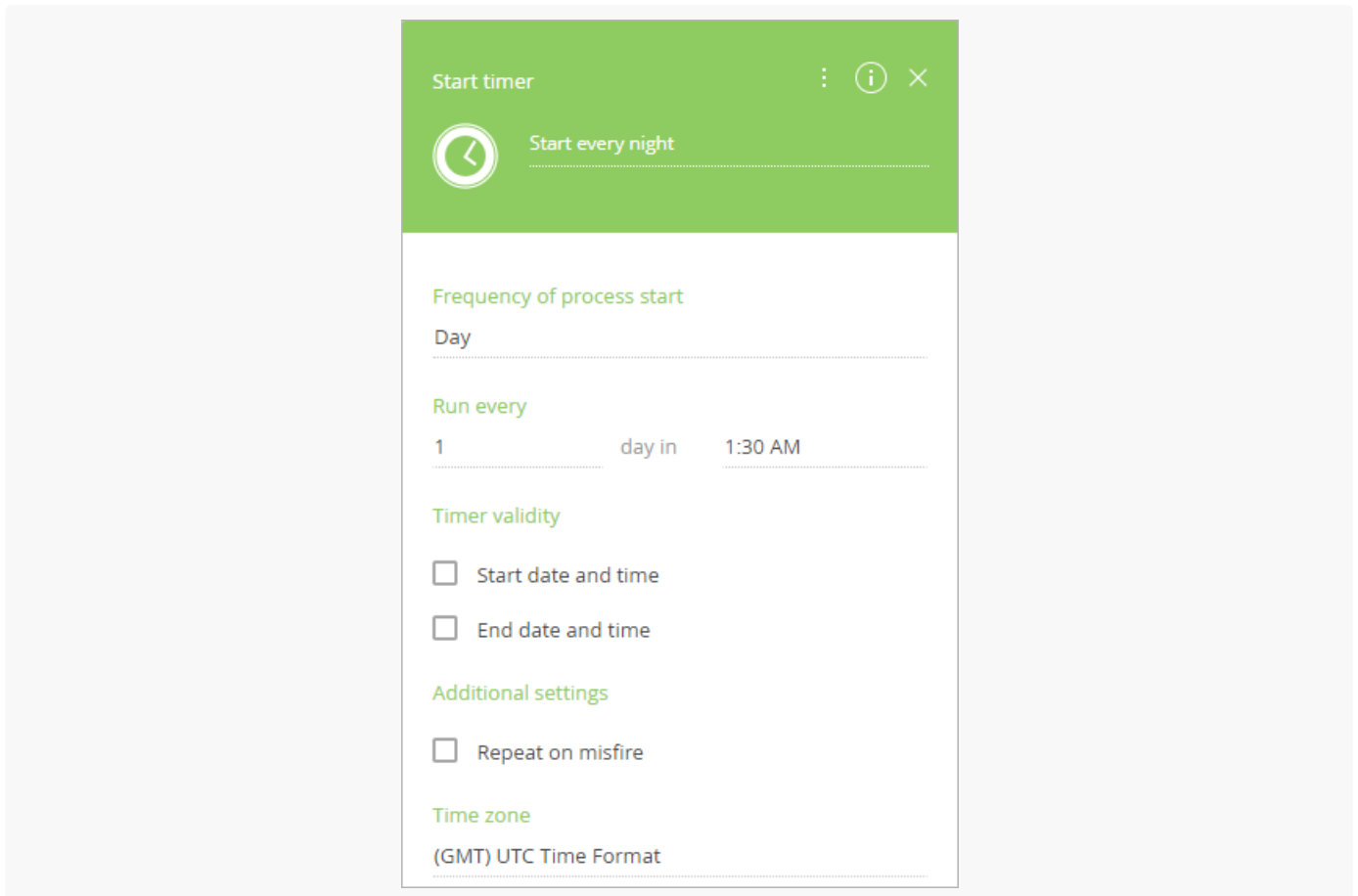
Fig. 3 Mapping a custom process parameter to a system setting value



- Populate [ *Title* ] and [ *Code* ] properties.
- In the [ *Lookup* ] field, select the [ *Currency* ] lookup.
- In the [ *Value* ] field, click ⚡ > [ *System setting* ] and select the [ *Base currency* ] system setting.

## 2. Set up the “Start every night” element properties:

Fig. 4 The “Start every night” timer event element



- a. In the [ *Frequency of process start* ] field, select "Day".
  - b. In the [ *Run every* ] field block, specify "1 day in 1:30 AM".
  - c. Select your time zone and configure other optional properties, if needed.
3. Set up the "Read base currency" element properties (Fig. 5):

Fig. 5 The "Read base currency name" element properties

Read data

Read base currency

Which data read mode to use?  
Read the first record in the selection

Which object to read data from?  
Currency

How to filter records?

Actions ▾

Id = Base currency

AND

+ Add condition

How to sort records?

Name Ascending

+ Add

What record data should the process read?

Read data from selected columns only

Short name

+ Add column

- In the [ *Which data read mode to use?* ] field, select “Read the first record in the selection”.
- In the [ *Which object to read data from?* ] field, select the “Currency” object.
- Set up the following filter in the [ *How to filter records?* ] area: “Id = Base currency”. In this case, “Base currency” is the name of the custom process parameter that you added earlier.

**Note.** To set the filter: click [ *Add condition* ], select the [ *Id* ] column; click [ *<?>* ] and select [ *Compare with parameter* ] command; in the [ *Select parameter* ] window, click [ *Process parameters* ] tab and select the “Base currency” parameter that you added earlier.

- In the [ *What record data should the process read?* ] field, select “Read data from selected columns only”.
  - Click [ *Add column* ] link and select the [ *Short name* ] column to have the process read the base currency short name (USD, EUR, etc.) from the lookup.
4. Set up the “Get currency exchange rate from the Fixer web service” element properties (Fig. 5):





endpoint will be enough. Enter “latest” value manually.

5. Add an outgoing conditional flow to the [ *Parallel gateway (AND)* ]. Set up the condition for moving down this flow only if the outgoing [ *Success* ] parameter of the “Get currency exchange rate from the Fixer web service” element is “true”:
  - a. In the [ *Condition to move down the flow* ] field, click ⚡ .
  - b. In the opened [ *Formula* ] window, on the [ *Process elements* ] tab, select the [ *Get currency exchange rate from the Fixer web service* ] element.
  - c. In the right side of the window, double-click the [ *Success* ] parameter.
  - d. Add “==true” text after the parameter variable, so that the whole formula text looks like this: [ *#Get currency exchange rate from the Fixer web service.Success#* ]==true
6. Add an outgoing condition flow to the [ *Terminate* ] signal. Set up the condition for moving down this flow only if the outgoing [ *Success* ] parameter of the “Get currency exchange rate from the Fixer web service” element is “false”. This way, if the web service call ends in error, the process will terminate (Fig. 6).
7. Add an [ *Add data* ] element for each exchange rate that must be updated. For example, to update Euro exchange rate, “Add EUR exchange rate” element properties (Fig. 7):

Fig. 7 – The “Add EUR exchange rate” element properties

The screenshot shows a configuration window titled "Add data" with a close button (X) and an information icon (i). Below the title bar, there is a header "Add EUR exchange rate" with a plus icon. The main content area is divided into several sections, each with a question and a text input field:

- Which object to add data to?**  
Exchange rate
- What is the data adding mode?**  
Add one record
- Which column values to set?**  
Currency  
[#Lookup.Currency.Euro#]
- Exchange rate  
[#Get currency exchange rate from the Fixer web service.RatesEUR#]
- Start  
[#System variable.Current Time and Date#]

At the bottom, there is a "+ Add field" button.

As a result, the process will automatically start every day, at 1:30 AM, and:

The process will terminate if the web service call results in an error.

- check current base currency
- call web service for exchange rates in relation to the base currency
- write the updated rates for the selected currencies (in this particular case, Euro, Australian dollar and Ruble) on the [ *Currency rates* ] detail of the [ *Currency* ] lookup (Fig. 8).