

Process webhooks

Process webhooks in Creatio

Version 8.0



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

Process webhooks in Creatio	4
Retrieve a webhook	4
Add a record based on the webhook	5
Search for a contact	7
Add tracking data to the contact connected to record based on a webhook (optional)	8
Delete the webhook	8

Process webhooks in Creatio

PRODUCTS: [ALL CREATIO PRODUCTS](#)

This functionality is available for Creatio 8.0.5 and later.

Webhook is an extension of a web app functionality via callbacks. Third-party users and developers can maintain, change, and manage the callbacks.

Since version 8.0.5 Atlas, you can integrate any webhook-capable third-party app with Creatio, including certain landing page builders, for example, Landingi. Creatio processes the webhook after retrieving it from a webhook service that is integrated with Creatio. To **integrate an external webhook service with Creatio**, follow the instructions in separate articles: [Set up external webhook service integration](#), [Landingi service integration](#).

In general, webhook management in Creatio includes the following **steps**:

- Retrieve a webhook. [Read more >>>](#)
- Add a record based on the webhook. [Read more >>>](#)
- Search for a contact. [Read more >>>](#)
- Add tracking data to the contact connected to record based on a webhook (optional). [Read more >>>](#)
- Delete the webhook. [Read more >>>](#)

Retrieve a webhook

Creatio retrieves a webhook from an external service in a `raw` form. A webhook in Creatio is stored in the "Webhook" object.

When Creatio service receives the webhook successfully, the service sends the **200** status code in response to the request. You cannot specify a different response. If Creatio is not available for any reason at the time of receiving the webhook, the Creatio service makes additional attempts to pass the webhook to Creatio 1 minute, 5 minutes, 60 minutes, 6 hours, 24 hours, 36 hours, 48 hours, 60 hours, and 72 hours later.

Creatio displays the "Webhook" objects in the [*Webhook*] lookup. Learn more in a separate article: [Create new lookups](#).

The `webhook` database table stores webhook data. View the primary columns of the `webhook` table below.

Primary columns of the `Webhook` table

Column	Type	Description
"RequestBody"	<code>nvarchar(max)</code>	The webhook body (<code>raw</code> body).
"QueryParameters"	<code>nvarchar(max)</code>	Parameters received in the request string within the response.
"Headers"	<code>nvarchar(max)</code>	Webhook headers from an external service.
"ContentType"	<code>nvarchar</code>	The data type of the webhook body.
"WebhookSource"	<code>nvarchar</code>	The ID of the external webhook service, for example, Landingi, Instapage, etc.
"SourceUrl"	<code>nvarchar(max)</code>	The URL of the external webhook service, for example, Landingi, Instapage, etc.
"ApiKey"	<code>nvarchar(max)</code>	The API key required to integrate the external webhook service, for example, Landingi.
"Status"	<code>uniqueidentifier</code>	The webhook processing status.

Status indicates the webhook lifecycle stage. Available webhook **statuses**:

- "New:" new webhook. Set for newly received webhooks. The [*Create object records based on incoming webhooks*] business process ([*Webhooks to entity*] business process in Creatio 8.0.5-8.0.6) handles webhooks whose status is "New."
- "Failed:" webhook parsing failed. Set after the execution of the basic logic of the [*Create object records based on incoming webhooks*] business process.
- "Processing:" webhook is being processed. Set during the execution of the basic logic of the [*Create object records based on incoming webhooks*] business process.
- "Done:" webhook was parsed successfully. Set after the execution of the basic logic of the [*Create object records based on incoming webhooks*] business process.

Add a record based on the webhook

After Creatio retrieves a webhook, the [*Create object records based on incoming webhooks*] business process ([*Webhooks to entity*] business process in Creatio 8.0.5-8.0.6) attempts to parse the webhook to add a Creatio record. You can use basic logic in the [*Create object records based on incoming webhooks*] business process or customize it to solve custom problems. For example, this is useful if you want to achieve the following:

- Set up a notification about new webhooks.
- Create a record based on the webhook.
- Send an email that contains webhook data.

- Set up a notification about a webhook processing error.

The business process works as follows out of the box:

1. Read a collection of new webhooks (50 records whose “Status” column value is “New”) from the `Webhook` database table every minute.
2. Change the webhook status from “New” to “Processing.”
3. Change the status of webhooks whose status remains “Processing” for longer than the specified period (1 hour by default) back to “New.” It is a disaster recovery mechanism. Learn more in [Wikipedia](#).
4. Create the record.

You can deactivate or customize the business process. Specify the maximum period within which the webhook can remain in any status in the [*Webhook status*] lookup.

The [*Create entity*] user task (the [*Webhook To Entity UT*] task in the [*Configuration*] section) executes the main functionality of the [*Create object records based on incoming webhooks*] business process. The user task works as follows out of the box:

1. Define a parser for the webhook data type based on the “ContentType” column value in the `Webhook` database table.
2. Define the object to add a record based on the webhook. Requires the `EntityName` parameter in the “RequestBody” column. Learn more about configuring the parameter in a separate article: [Set up external webhook service integration](#).
3. Populate other columns of the record based on the parameters of the “RequestBody” webhook column.
4. Save the record to Creatio.

If the webhook does not contain the `ContentType` or `EntityName` parameter value in the “RequestBody” column, the process cannot create the record based on a webhook. In this case, the “Status” column of the `Webhook` database table is set to “Failed.”

If the incoming webhook data differs from the expected data, the parsing fails. The process logs the detailed error in the [*Webhook parse errors log*] lookup and changes the webhook status to “Failed.”

View the logged webhook processing errors in the table below.

Logged webhook processing errors

Error text	Error description	Error correction
An error occurred while setting the column value. ColumnName: {ColumnName}. EntityName: {EntityName}. Exception:Cannot cast {ActualType} to {ExpectedType}.	The column data type of the transferred webhook does not match the column data type in the added Creatio object. The object is added and the parsed columns are populated. The columns specified in the error are not populated. Webhook status is not changed to “Failed.”	Verify the mapping of webhook columns to Creatio object columns. Learn more in a separate article: Set up external webhook service integration .
An error occurred while processing the	Unable to find the Creatio object specified in the <code>EntityName</code> webhook parameter.	Verify the object name in the external webhook service.

<pre>webhook. Entity name: "{EntityName}" Exception: Item with name "{EntityName}" not found.</pre>		<p>Ensure the <code>EntityName</code> parameter value matches the object code. Learn more in a separate article: Set up external webhook service integration.</p>
<pre>An error occurred while processing the webhook. The parameter "{Entity name}" should be existing and not empty.</pre>	<p>The <code>EntityName</code> webhook parameter does not exist or is empty.</p>	<p>Add or fill out the object parameter according to the recommendations in a separate article: Set up external webhook service integration.</p>
<pre>An error occurred while processing the webhook. Exception: The parameter " {Content type}" should be existing and not empty.</pre>	<p>The <code>ContentType</code> webhook parameter does not exist or is empty.</p>	<p>Verify the data type in the <code>ContentType</code> webhook parameter. If the data type is populated, but the field name is different, contact Creatio support for a detailed analysis of the webhook service.</p>
<pre>An error occurred while processing the webhook. Exception: {Content type} type of webhook is unsupported.</pre>	<p>The data type listed in the <code>ContentType</code> webhook parameter is unsupported. Creatio 8.0.5 Atlas and later supports <code>application/json</code> and <code>multipart/form-data</code> data types of the webhook body.</p>	<p>Contact Creatio support.</p>
<pre>An error occurred while processing the webhook. Exception: Could not parse the webhook body: {RequestBody}.</pre>	<p>An unexpected error occurred when processing the <code>RequestBody</code> webhook parameter.</p>	<p>Contact Creatio support for a detailed analysis of the webhook.</p>

Search for a contact

Since version 8.0.7 Atlas, Creatio runs the [*Searching and creating contact*] subprocess as a part of the [*Create object records based on incoming webhooks*] business process. The [*Searching and creating contact*] subprocess searches for a contact by various data or creates a contact for the [*Submitted form*] object record that the [*Create object records based on incoming webhooks*] business process adds. Learn more about the [*Searching and creating contact*] business process in a separate article: [Identify contacts that submit web forms](#).

To search and create contact for **another object** based on a webhook:

1. Open the [*Define search options and create contact from webhook*] business process.

2. Implement custom business logic to search and create contact using a record based on a webhook. For example, [*Order*]. Custom logic settings are similar to out-of-the-box settings for the [*Submitted form*] object.

Add tracking data to the contact connected to record based on a webhook (optional)

Since version 8.0.7 Atlas, if a landing page tracks events, you can add tracking data to the contact that the [*Searching and creating contact*] subprocess found or created.

Creatio runs the [*Define contact and populate site events*] subprocess as a part of the [*Create object records based on incoming webhooks*] business process. The [*Define contact and populate site events*] business process adds tracking data to the contact.

The [*Define search options and create contact from webhook*] subprocess works as follows out-of-the-box:

- If a **landing page configured to track landing page events**, the subprocess adds tracking data using [Matomo](#) on the contact page (the [*Marketing*] tab → the [*Web sessions*] and [*Web actions*] expansion panels).
- If a **landing page is not configured to track landing page events**, the subprocess does not add tracking data on the contact page.

Delete the webhook

Creatio deletes webhooks from the `Webhook` database table automatically after a specific period. The **period values** are as follows out of the box:

- 90 days for webhooks whose status is “New”
- 90 days for webhooks whose status is “Failed”
- 30 days for webhooks whose status is “Done”

You can change the period values in the [*Webhook status*] lookup.

Note. Deletion of a webhook does not delete the created record. Deletion of a webhook whose status is “Failed” deletes the connected log.