

# Webhook



Webhook is a default Switch app that subscribes to specific incoming HTTP requests and injects the received request content as a new job in the flow. The job file name is 'payload' and the file extension is automatically generated from the 'Content-Type' header.

You can add this tool to your flow to accept notifications via HTTP/HTTPS protocol.



**Note:** As of version 7, Webhook is a Node.js tool. The user preferences in the Webhooks (legacy) category do not apply anymore! Instead, the protocol (HTTP/HTTPS) and the port for the Switch Web Service (by default 51088) as set in the Web services category must be used to define the URL to which the webhook is sent. Additionally a fixed part is added to the URL; it is now mandatory to add "scripting" to the path.

## Using webhooks

In order for the Webhook element to work correctly two requirements must be fulfilled.

Firstly, the protocol and port for the Switch Web Service must be properly configured in the Web services category of the Switch preferences.

Secondly, an external system that is capable of sending notifications in the form of a webhook must be configured to send such a webhook to the Switch Server. The way in which this is done, depends of course on the external system, but the URL to which the webhook is to be sent must be built up of the following components:

- The protocol type (HTTP or HTTPS as configured in the Switch Preferences)
- The name or IP address of the Switch Server to which the outside world can connect (configured by IT)
- The port number on which Switch is listening for incoming webhooks (also configured in the Preferences)
- A fixed part: "scripting" (mandatory!)
- A path that is determined by the external system and that you must replicate in the "Path" property of the Webhook element (for example, "notify" in the example below)

This results in a URL that could look like this: <https://switchserver:51088/scripting/notify>

There are two important IT-related considerations to be taken into account. The internal IP address that you see on your Switch Server may not be the same as its public IP address. The external system must of course connect to the public IP address. The port number to be used by the external system and the port number on which Switch is listening for webhooks are not necessarily the same! The firewall can be set up to receive the webhook on a particular port (51088 in this example) and forward it with a technique called port forwarding to the one that is configured in the Switch preferences.

Note that the size of incoming webhook requests in Switch is limited to 1MB. For more information, refer to the [Web services preferences in the Switch Reference Guide](#).

### Remark

With this flow element, periodic calls to third-party systems or services (such as an MIS or CRM system) to check for changes are no longer needed. Instead, these systems will now call Switch the moment something has changed (for example, a new order that has been entered), and this will trigger Switch to do something (for example, to create a new job based on this new order).

## Keywords

If you enter one of the following keywords in the Search field at the top of the Flow Elements pane, the **Webhook** element will be shown in the list:

- Webhook
- HTTP
- HTTPS
- POST
- PUT
- DELETE
- URL
- request




## Connections

Webhook does not support any incoming connections.

Webhook requires one outgoing connection.

## Properties

Property	Description
Element type	The flow element type: Webhook.  This property is useful to identify renamed flow elements. It cannot be changed.
Name	The name of the flow element displayed in the canvas.
Description	A description of the flow element displayed in the canvas. This description is also shown in the tooltip that appears when moving your cursor over the flow element.
Protocol type	Defines the protocol.  Options are: <ul style="list-style-type: none"> <li>• HTTP (default)</li> <li>• HTTPS</li> </ul>
Request method	Defines the request method.  Options are: <ul style="list-style-type: none"> <li>• POST (default)</li> <li>• PUT</li> </ul>

Property	Description
	<ul style="list-style-type: none"> <li>DELETE</li> </ul>
Path	<p>Defines the URL path.</p> <p>This a mandatory field.</p> <p><b>Example:</b> notify</p> <hr/> <p> <b>Note:</b> The URL to which the webhook is sent should be defined as &lt;protocol&gt;://&lt; Switch IP address&gt;:&lt;port number&gt;/scripting/ followed by the value of the Path property of the webhook. For example: http://localhost:51088/scripting/<b>notify</b>.</p> <hr/>
Payload dataset name	<p>Defines the name of the dataset. The default name is Payload.</p> <hr/> <p> <b>Note:</b></p> <ul style="list-style-type: none"> <li>The dataset can be accessed by building a location path. Refer to the <a href="#">Switch Reference Guide</a> for more information.</li> <li>Note that the payload file size should not exceed 1MB.</li> </ul> <hr/>
Payload dataset model	<p>Defines the extension of the output payload file and the type of dataset model. Options are:</p> <ul style="list-style-type: none"> <li>Opaque</li> <li>JSON</li> <li>JSON from MIME (Supports MIME payloads in the Webhook request and converts them to a JSON dataset).</li> </ul> <hr/> <p> <b>Note:</b> A MIME can contain different types of information: form data (usually values from input fields) and attachments. Only the form data is added to the JSON. Attachments if any are ignored and a warning message will be generated.</p> <hr/> <ul style="list-style-type: none"> <li>XML</li> <li>JDF</li> </ul>
Request metadata	<p>Defines how to handle the request metadata like arguments, path, headers.</p> <p>Options are:</p> <ul style="list-style-type: none"> <li><b>Discard:</b> The request metadata is discarded.</li> <li><b>Inject:</b> The request metadata will be injected as a new job that is the file in JSON/XML format.</li> </ul>

Property	Description
	<ul style="list-style-type: none"><li>• <b>Attach as dataset:</b> The request metadata will be attached as dataset with the name specified in the <i>Dataset name</i> property ('Webhook' by default) to the outgoing job.</li></ul>
<i>Format</i>	<p><i>This option is only available if Request metadata is set to Inject.</i></p> <p>Defines metadata file format: JSON(default) or XML</p>
<i>Make job folder</i>	<p><i>This option is only available if Request metadata is set to Inject.</i></p> <ul style="list-style-type: none"><li>• If set to Yes, a job folder is injected</li><li>• If set to No, the files are injected as separate jobs</li></ul>
<i>Model</i>	<p><i>This option is only available if Request metadata is set to Attach as dataset.</i></p> <p>This option defines the dataset model: XML or Opaque (default). If Opaque is selected, the dataset file will be in JSON format.</p>
<i>Dataset name</i>	<p><i>This option is only available if Request metadata is set to Attach as dataset.</i></p> <p>This option allows you to specify the name of the dataset.</p>



**Note:** This tool does not perform any HTTP authentication.