

Webhook messages

Description

Webhook messages is a small simple app that makes it possible to pickup messages from a webserver and present them in Switch messages.

When you use HTTP Request to send data and receive data from a web server you only get messages if Switch was able to get in contact with the server or not. If you are sending data to a database you have no information if that succeeded or not. That problem will this app solve for you.



In the web pages you are sending data to you have to add some code that sends messages back to this app in Switch over the Webhook functionality. The app itself just receives the data you send and shows it in the Switch Messages. Webhook messages are not receiving any files in the Switch flow, it just sits there and waits for messages from your web server.

Below you see some messages from different PHP-scripts that interacts with Switch.

Element	
Webhook warning	Job with ordernumber 52369-AE has been deleted from review job list.
Webhook warning	Reminder counter is set to 4 for job 52369-AE
Webhook warning	Reminder counter is set to 3 for job 52369-AE
Webhook warning	Reminder counter is set to 2 for job 52369-AE
Webhook warning	Job with ordernumber 52369-AE has been added to review job list.
Webhook warning	Preview thumb is uploaded
Webhook warning	XML-file with name ycq01svhjd.xml is uploaded
Webhook warning	Large preview of job is uploaded
Webhook warning	Preflight of job is uploaded

You can use the same app for several different web pages as long as they all call the same webhook path the app is receiving from. But you can also have the app in several flows but you must then have unique properties in the path setting to avoid any conflicts in the Webhooks.

Compatibility

Switch 18 and higher. Windows or Mac OSX.

Compatibility third-party applications

No third party applications are used by this app.

Connections

Webhook messages has no incoming or outgoing connections.

Properties detailed info

Property	Value
Name	Webhook messages
Description	
Protocol type	Any
Path	webhook_warning
Message name from server	message
Message source	Web server
Message type name from server	warningLevel
Debug JSON code	No

Property	Value
Name	Webhook messages
Description	
Protocol type	Any
Path	webhook_warning
Message name from server	message
Message source	From app
Message type	<ul style="list-style-type: none"> ✓ Info Warning Error Debug
Debug JSON code	

Flow element properties

- **Protocol type**
 - Dropdown with values "Any" (Default), "HTTP" and "HTTPS" this property must match the settings in Switch preferences. If set to "Any" it will use the settings from the preferences.
Example: *https://switchserver:51080/webhook_warning*
- **Path**
 - Defines the URL path that is called from the web page to specify which Webhook that the message is intended for. The same value is specified in the HTML as part of the whole URL-address to your Switch server.
See below for an HTML/PHP example.
- **Message name from server**
 - This property corresponds to the name of the message you send to the Switch server and that the app parses.
It comes as a JSON-string {"message": "Message"} where "message" is the value of the property "Message name from server".

- **Message source**

This property have two choices, "From app" or "Web server", default is "From app"

- **From app** give you a dropdown menu with four choices of "**Message type**", Info, Warning, Error or Debug. This will give all arriving messages the same message type.
- **Web server** let you decide which level of message type in the web page code depending on the severity of the message the web page is sending. Here you have a property, "**Message type name from server**" to set a second name for the message type you send. In this case the value of the property coming from the webserver should be as in table below.

Message type	Value
Info	1
Warning	2
Error	3
Debug	-1

- **Debug JSON**

Dropdown with "Yes" or "No", default is "No".

When setting this to "Yes" your JSON-code sent from the server will show in the Switch Messages Debug list. This will be helpful when you develop the code on your web page.

Example PHP code

On the webserver that gets the HTTP Requests from Switch you have to add some code that sends error or warnings messages back to Switch to be shown in the messages.

```
if ($mysqli->query("UPDATE review_jobs SET counter='$counter', reviewLink='$previewLink' WHERE orderNo = '$orderNo'") === TRUE){
    $errorMessage = "Reminder counter is set to " . $counter . " for job " . $orderNo;
    if($counter > "2"){
        $warningLevel = "3";
    }
    else{
        $warningLevel = "1";
    }
}
else
{
    $errorMessage = "Error updating record: " . $mysqli->error . " in database " . $db;
    $warningLevel = "3";
}

// Sending message back to Switch server
if($warningLevel != ""){
    $data="{\"message\": \" " . $errorMessage . "\", \"warningLevel\": \" " . $warningLevel . "\"}";

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "http://your_switch_server:51080/webhook_warning");
    curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
    curl_setopt($ch, CURLOPT_HEADER, false);
    curl_exec($ch);
    curl_close($ch);
}
```

In the PHP example above we have received some data from Switch to update a database record. Normally you will not see any messages in Switch that shows what happens on the web server, the only thing you will see is that the web server did receive your call but how it is handled and any results are not told. If there is any problems with the database you will not see that in an easy way. The PHP-pages that Switch is calling are usually not seen by any user so there is no easy way to display any eventual errors.

You have to define the messages that will be sent back to Switch in the HTML/PHP (or any other language) to something readable and useful.

You also see that we in this case set the variable **\$warningLevel** to different values depending on the severity of the message. In this case if everything is normal it appears in the Messages pane as an Info message, else it will show up red as a warning.

- The variable **\$data** in this example is where you construct the JSON string that is sent to Switch. Note that it has to be valid JSON, if not the Webhook will fail with a "JSON.parse error".
- Note that the message has to be in the "CURLOPT_POSTFIELDS" section to be picked up by this app.
- In the "CURLOPT_URL" section you have the complete URL to your Switch server including the property **Path** that you specify in Switch for this app. In the example above that path is "webhook_warning" and it should be a unique name for each instance of the app. You can have several web pages that call the same "Path".