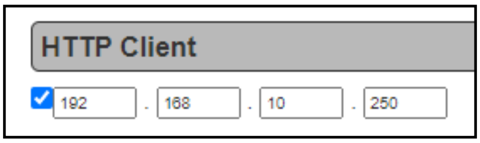
Generic HTTP Client

This device core allows the panel to send generic HTTP messages as a client to a given IP address and port.

The receiver IP is set via the standard device core configuration. Currently, we only support sending requests to an IP address - and not to hostnames.
For example:

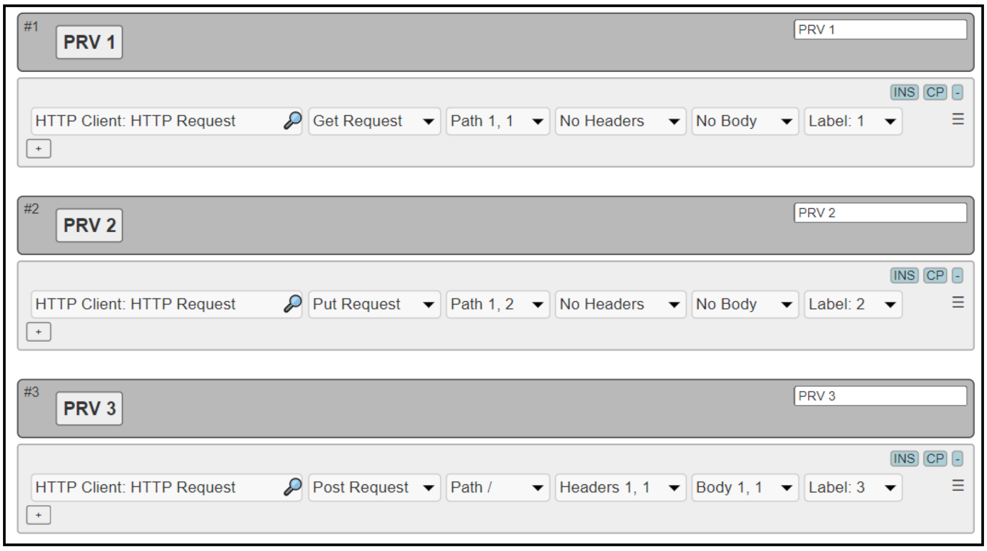


HTTP Client

☑ 192 . 168 . 10 . 250

Action Overview

The device core only has one action: “HTTP Request”. Below are three examples of configuration:



#1 PRV 1 — PRV 1

HTTP Client: HTTP Request | Get Request | Path 1, 1 | No Headers | No Body | Label: 1

#2 PRV 2 — PRV 2

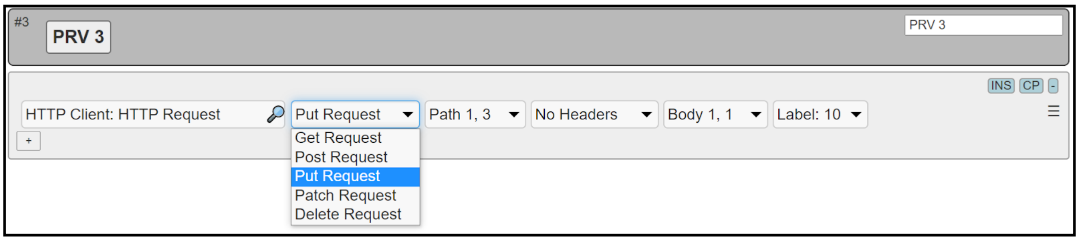
HTTP Client: HTTP Request | Put Request | Path 1, 2 | No Headers | No Body | Label: 2

#3 PRV 3 — PRV 3

HTTP Client: HTTP Request | Post Request | Path / | Headers 1, 1 | Body 1, 1 | Label: 3

In the action, you have five parameters that can be changed from left to right:

"Request Type": Selects between the different types of HTTP request that is supported in this core:



#3 PRV 3

PRV 3

HTTP Client: HTTP Request | Put Request | Path 1, 3 | No Headers | Body 1, 1 | Label: 10

Get Request	Send Simple HTTP Command
Post Request	Send Complex HTTP Command
Put Request	Send Note for HTTP Command
Patch Request	Send Note for HTTP Command
Delete Request	Send Note for HTTP Command

"Path": Choose the Path for where the request should be sent (default is "/"). The path is the command that follows the IP address.

"Header": Choose what Header should be sent with the request (default is none). The header is additional metadata that can define the body.

"Body": Choose what Body should be sent with the request (default is none). The body is additional specified data.

Note: For simple HTTP commands, you only need to use Path.

"Label": will select a custom label for the button if it has a display. Label 0 will keep the default for the action (see Action Feedback), while starting at Label 1, will reference the String of the same number on the Manage Media Page. (see the UniSketch Systems Actions Manual for more information on labels).

Action Feedback: When the action is attached to a screen it will give you the basic info about what type of request and if it's **"Available"** or **"Processing"**. In order to give these a name that means more to you, please use the label field to rename the buttons to something more specific for that request.

Device Core Options

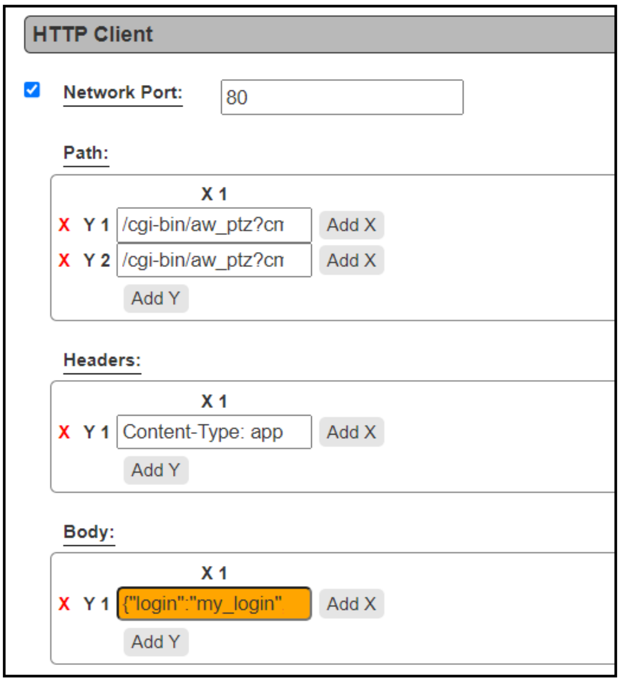
Setting up your Path, Header and Body variables:

For all three types, you will find a UI element within device options where you have an X:Y grid for each variable type. Fill out these in the same manner as you would do with the grid for labels, formatting the numbers as X then Y.

For the path variable please start it with "/" as this will be placed directly after the port number and will most likely fail without the slash.

The header and body will be sent out "as is", so no encoding is done inside the SKAARHOJ controller, therefore please keep this in mind when setting up your requests. If you need to escape a character please use the backslash "****" character. Some specific characters might need URI encoding, please do this when you type it into the text fields.

Here is an example of those filled out Path, Header, and Body grids:



HTTP Client

☑ Network Port: 80

Path:

	X 1
Y 1	/cgi-bin/aw_ptz?cn
Y 2	/cgi-bin/aw_ptz?cn

Headers:

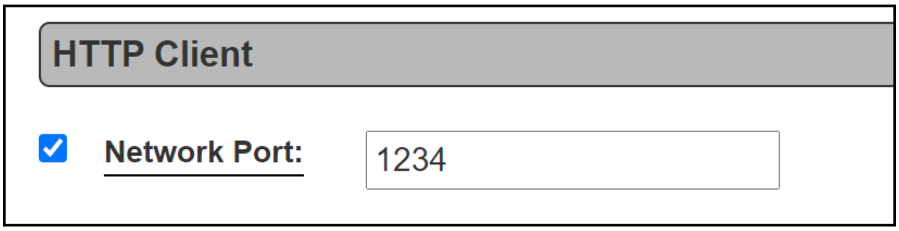
	X 1
Y 1	Content-Type: app

Body:

	X 1
Y 1	{"login":"my_login"

Note the yellow background is because the current website isn't super happy with using a " in a text field. It still works as intended, but will give you a small warning. In order to type a " please copy and paste it in.

Change of port: The Network Port information can generally be found in the manual or web interface of the device you are trying to send commands to. For cameras, the most common network port is 80.



HTTP Client

☑ Network Port: 1234

Want to send JSON?

If you want to send JSON or other special formats, please define the content type in the header field.

JSON use: Content-Type: application/json

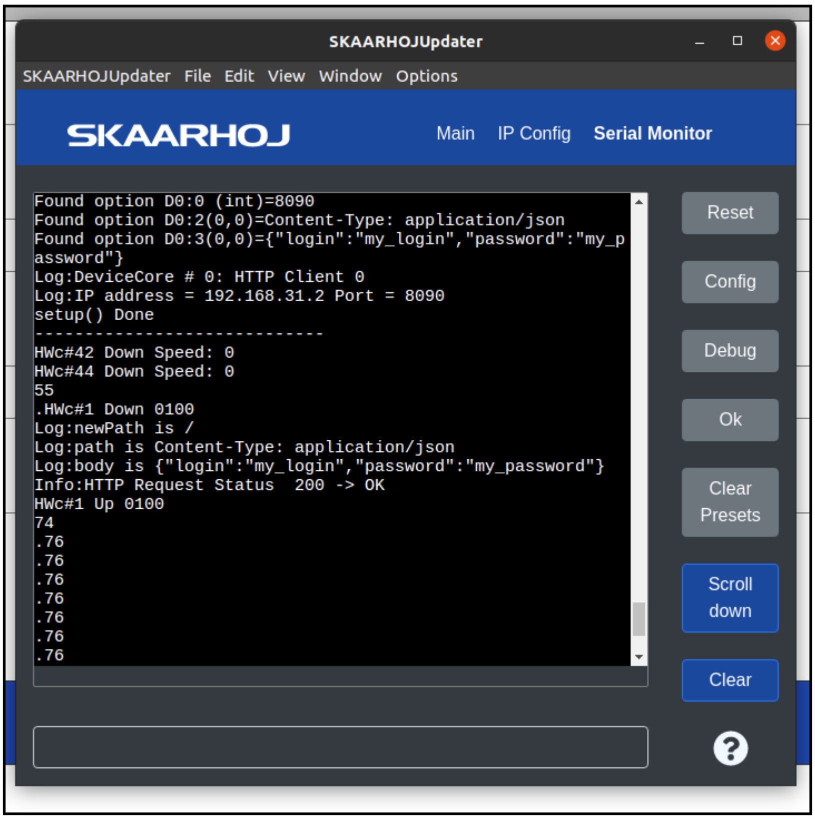
HTML use: Content-Type: text/html

Plain text use: Content-Type: text/plain

And in the body fill out the content that you want to have sent out. So for JSON, the body would look something like this:

{"login":"my_login","password":"my_password"}

Verify a request went successfully: When you press a button that sends a request out, the SKAARHOJ panel will show a message in the serial port. This can be used to check up on your setup while testing, and as a place to check when debugging. After a message has been send and it was successful it should show something close to this, if not please reboot the device into debug mode by clicking the “debug” button on the right side of the serial monitor:



```
Found option D0:0 (int)=8090
Found option D0:2(0,0)=Content-Type: application/json
Found option D0:3(0,0)={"login":"my_login","password":"my_p
assword"}
Log:DeviceCore # 0: HTTP Client 0
Log:IP address = 192.168.31.2 Port = 8090
setup() Done
-----------------------------
HWc#42 Down Speed: 0
HWc#44 Down Speed: 0
55
.HWc#1 Down 0100
Log:newPath is /
Log:path is Content-Type: application/json
Log:body is {"login":"my_login","password":"my_password"}
Info:HTTP Request Status  200 -> OK
HWc#1 Up 0100
74
.76
.76
.76
.76
.76
.76
.76
```