iVSEC

INTELLIGENT VIDEO SECURITY

https://ivsecurity.com.au/support/manuals/

HTTP WEBHOOK API FEATURE

ABOUT THIS DOCUMENT

Some of the iVSEC camera and NVR range have the ability to use the HTTP Webhook API, so that other network devices can be interacted with when a specific alert is triggered by an event or schedule.
This could be anything from a door lock relay, a web relay, Shelly device, or even an IP controllable light.
All instruction contained within this document are using the Web Interface, though similar steps are performed using the NVR Interface.

Cameras with this feature start at the following and include the later revisions :
NC323XD, NC323ADX, NC512ADX, NC512XD, NC528XD, NC531XD, NC531ADX, NC542ADX, NC543ADX, NC544ADX, NC691XB

GETTING STARTED

To adjust these settings, you will have the following:

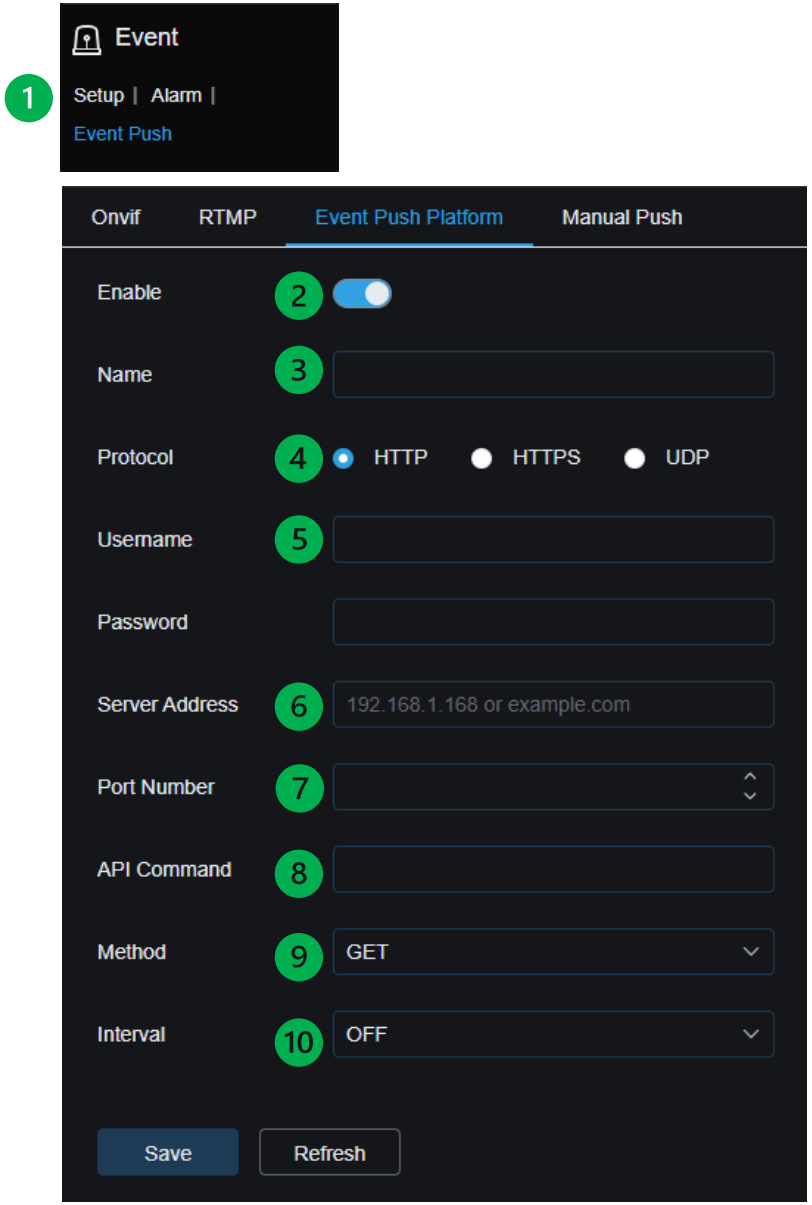
- LCD monitor and USB mouse connected to iVSEC recorder.
- Laptop (if you are logging into recorder using a web browser).
- iVSEC X mobile app installed and your iVSEC recorder added to the app.
- IP Controllable device that can accept either POST or GET requests.

TABLE OF CONTENTS

HTTP Event Push – Setup Page 03
HTTP Event Push – Example 1 – SH-SHELLYDIM Page 04
HTTP Event Push – Example 2 – 2N9137411E Page 05
HTTP Event Push – Example 3 – AX9155211C Page 06
HTTP Event Push – Alarm – Enabling Page 07
HTTP Manual Push – Setup Page 08
HTTP Manual Push – Triggering – From NVR/Camera Page 09
HTTP Manual Push – Triggering – From iVSEC X App Page 10

HTTP Event Push – SETUP

1. Open the **Remote Settings** of the Camera and navigate to the **Event** section.
   Select the **Event Push Platform** option.
2. Click the **Enable** toggle to allow the feature to work.
3. Fill out the **Name** so you know what you will be controlling.
4. Select the **Protocol** that you will be using, for the examples in this document we will be using **HTTP**.
5. If **HTTPS** is being used, then the **Username** and **Password** need to be filled out. These are the ones used to access the IP Device.
6. This is where you will need to enter the **Server Address** (Domain or IP Address) for the IP Device.
7. The **Port Number** used to talk to the Server Address needs to be entered here.
8. The **API Command** is the syntax that goes after the **Server Address**, examples will be listed later.
9. Depending on the Method that the IP Device needs, you will choose either GET or POST.
10. The **Interval** determines the time between the **Event Push** being able to be triggered. This prevents the Event Push being triggered constantly.



HTTP Event Push – Example 1 – SH-SHELLYDIM

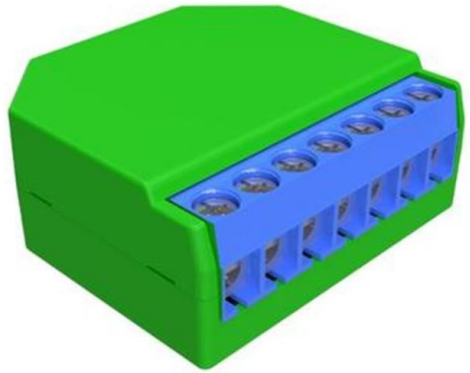
If you were to enter http://192.168.85.151/light/0?turn=toggle in a browser, then then relay 1 toggle its state from on to off, or off to on. There are other functions that the SH-SHELLYDIM can perform, though we will be using this as the example.
Visit https://shelly.guide/webhooks-https-requests/ for more information.

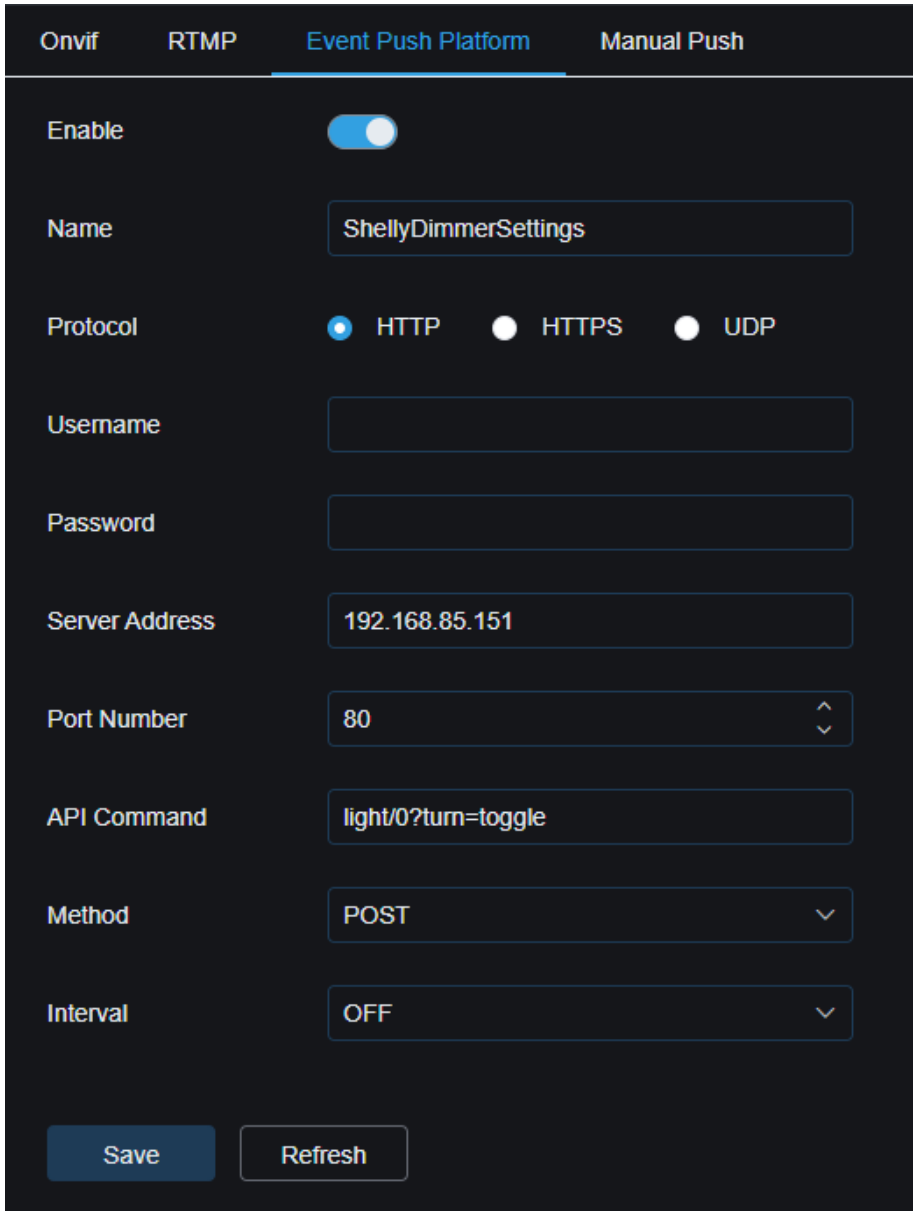
In the example we are using the **Protocol**, and as such a **Username** and **Password** are not required.

Our **Server Address** is 192.168.85.151, using the default **Port** 80 for communication.

The **URL** is going to be everything after the **Port**, though it should be noted that if Port 80 is to be used then the **Port** number may not be specified.
In this example the **API Command** will be light/0?turn=toggle .

As the SH-SHELLYDIM accepts only POST requests for this application, then POST is the required **Method**.





HTTP Event Push – Example 2 – 2N9137411E

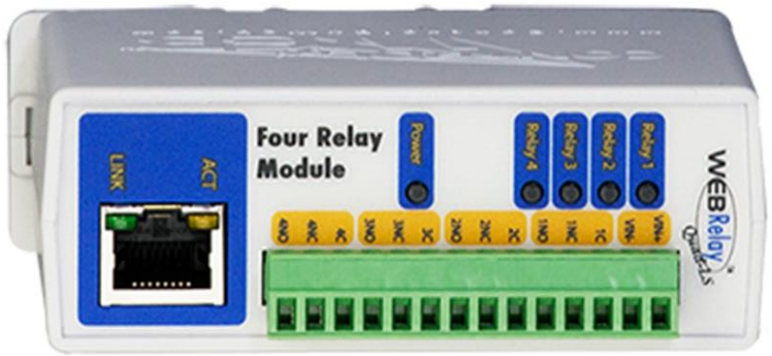
If you were to enter http://192.168.85.160:80/state.xml?relay1State=2 in a browser, then then relay 1 would turn on for 5 seconds and then off. There are other functions that the 2N9137411E can perform, though we will be using this as the example.
Visit https://wiki.2n.com/hip/inte/latest/en/6-access-control/webrelay for more information.

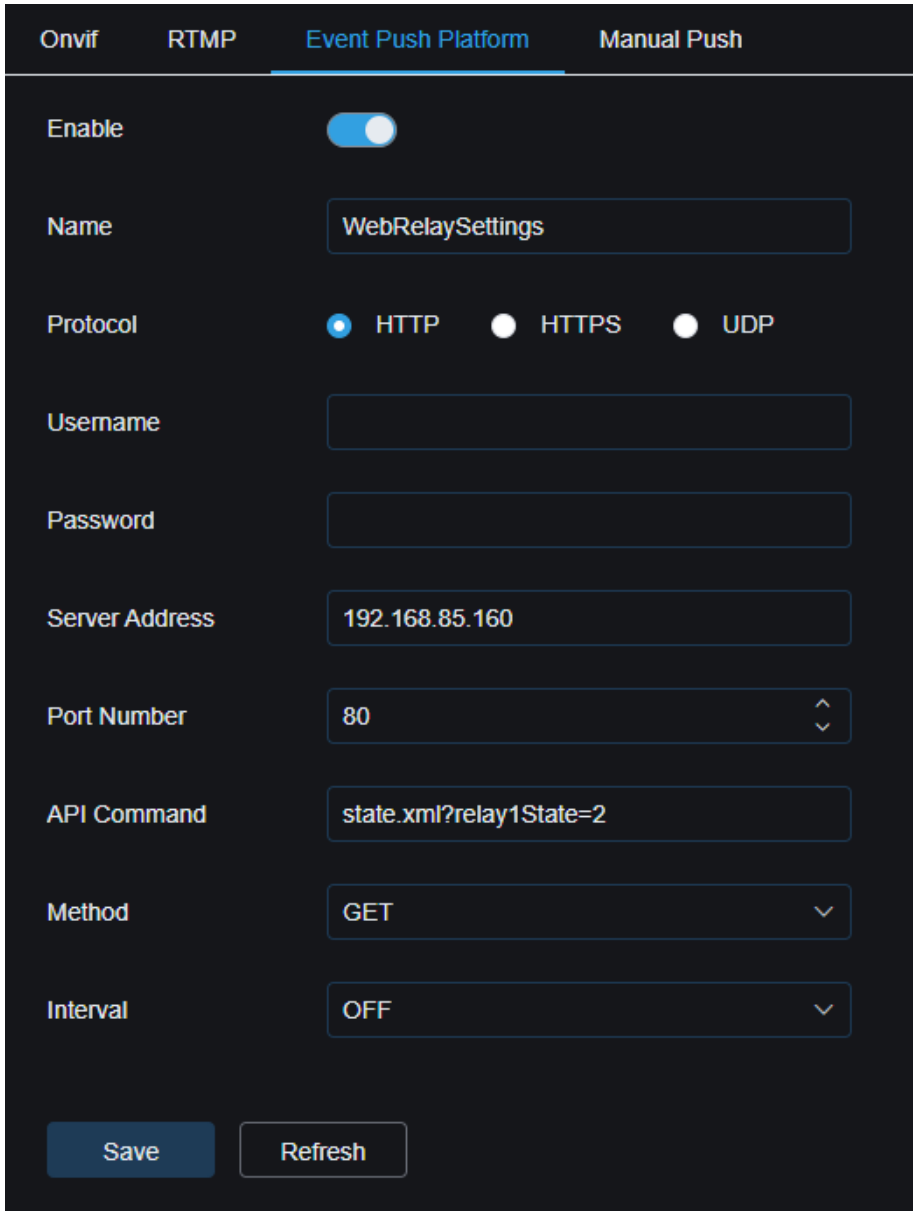
In the example we are using the **Protocol**, and as such a **Username** and **Password** are not required.

Our **Server Address** is 192.168.85.160, using the default **Port** 80 for communication.

The **URL** is going to be everything after the **Port**, though it should be noted that if Port 80 is to be used then the **Port** number may not be specified.
In this example the **API Command** will be state.xml?relay1State=2 .

As the 2N9137411E accepts only GET requests for this application, then GET is the required **Method**.





HTTP Event Push – Example 3 – AX9155211C

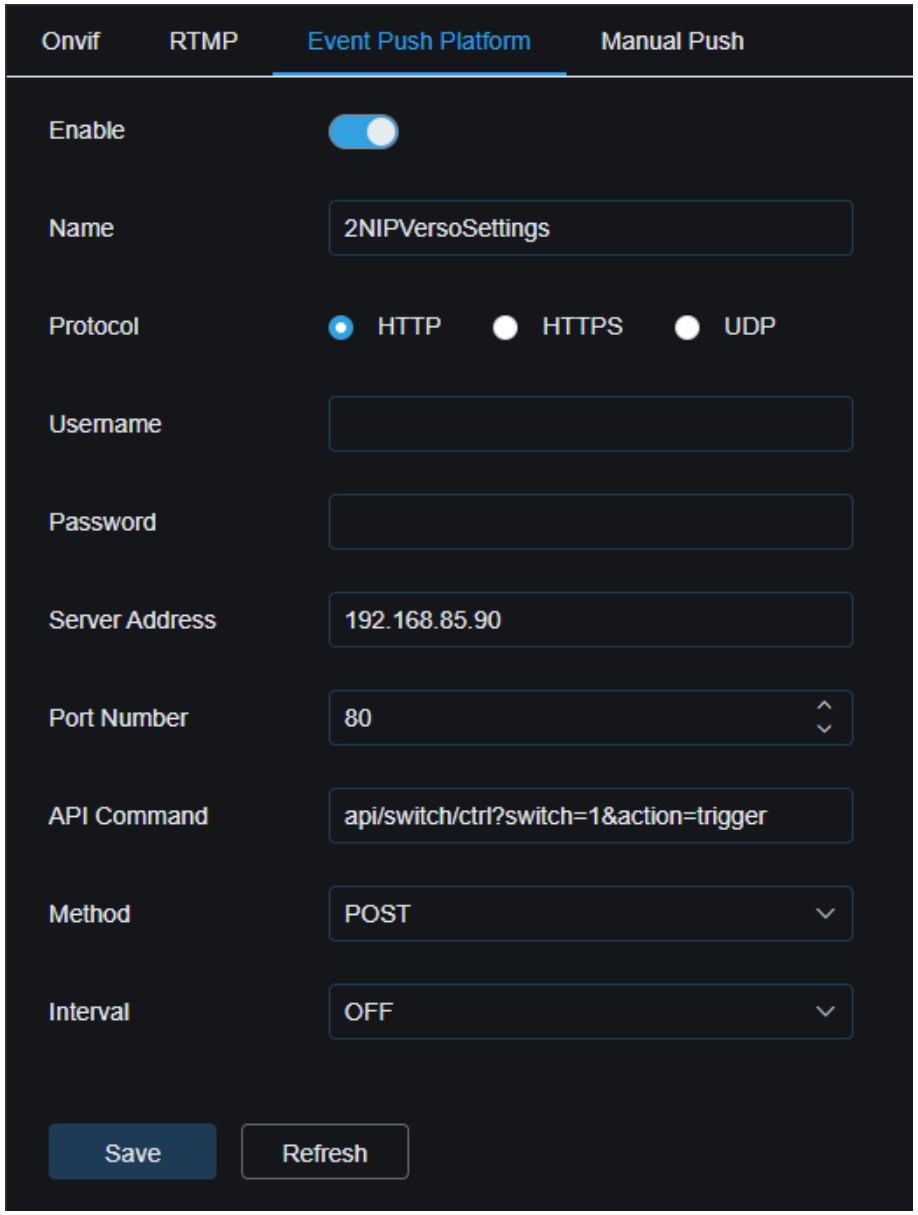
If you were to set the HTTP API Services on the AX9155211C to have a connection type to Unsecure (TCP) and then proceed to enter http://192.168.85.190/api/switch/ctrl?switch=1&action=trigger in a browser, then then switch 1 will be triggered for a predetermined amount of time (5 seconds). There are other functions that the AX9155211C can perform, though we will be using this as the example.

In the example we are using the **Protocol**, and as such a **Username** and **Password** are not required.

Our **Server Address** is 192.168.85.190, using the default **Port** 80 for communication.

The **URL** is going to be everything after the **Port**, though it should be noted that if Port 80 is to be used then the **Port** number may not be specified.
In this example the **API Command** will be api/switch/ctrl?switch=1&action=trigger.

As the AX9155211C can accepts only POST requests when HTTP API Services are set to Unsecure, then POST is the required **Method**.

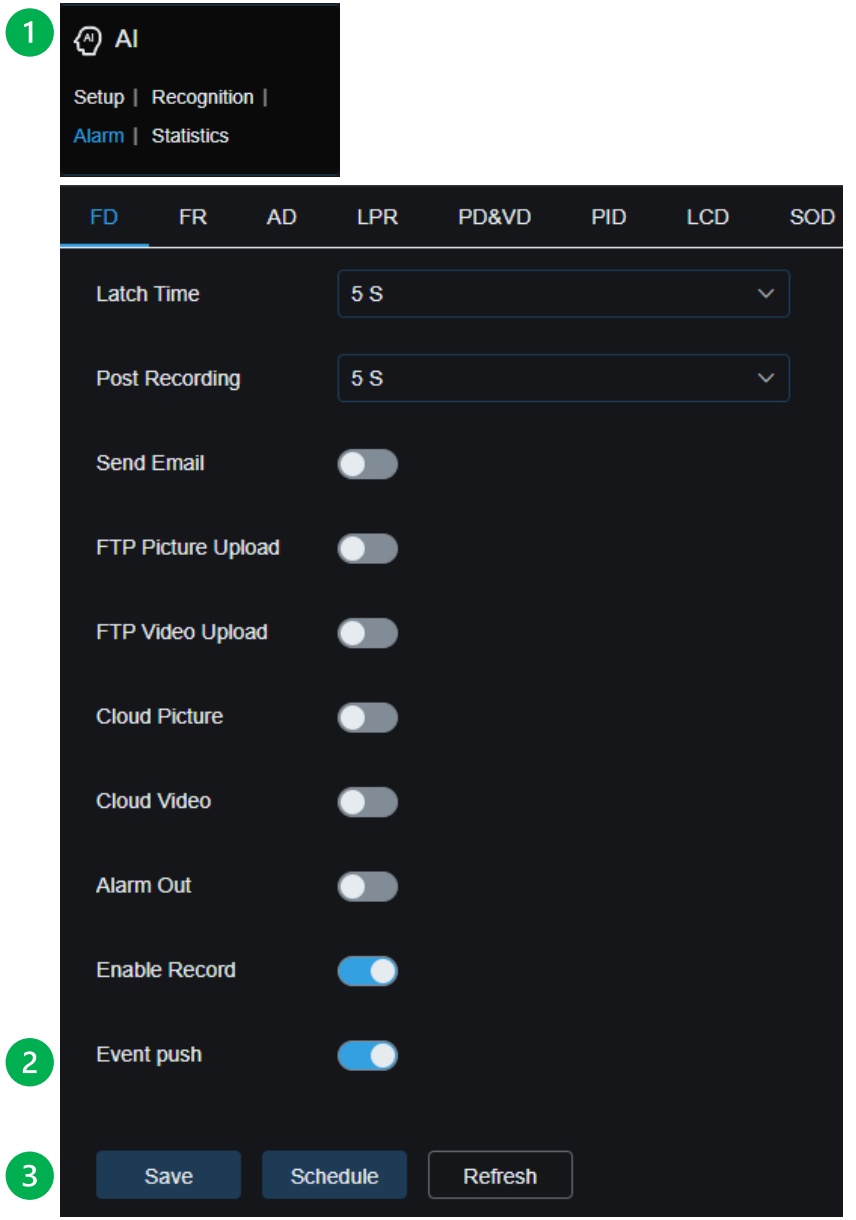


HTTP Event Push – ALARM – Enabling

1. Navigate to the **AI** section.
   Select the **Alarm** option.

2. Select the **FD (Face Detection)** tab and down at the very bottle toggle the **Enable** feature to on for **Event Push**.

3. Once set, click the **Save** button and the camera/device will send the Event Push that you have set up every time the camera/NVR detects a face.



HTTP Manual Push - Setup

When using the iVSEC X App, or viewing the NVR or camera directly via the web or NVR interface, you are able to trigger one of two Manual Push Events.

This is filled out the **EXACT** same way as the previous examples for the Event Push Platforms.

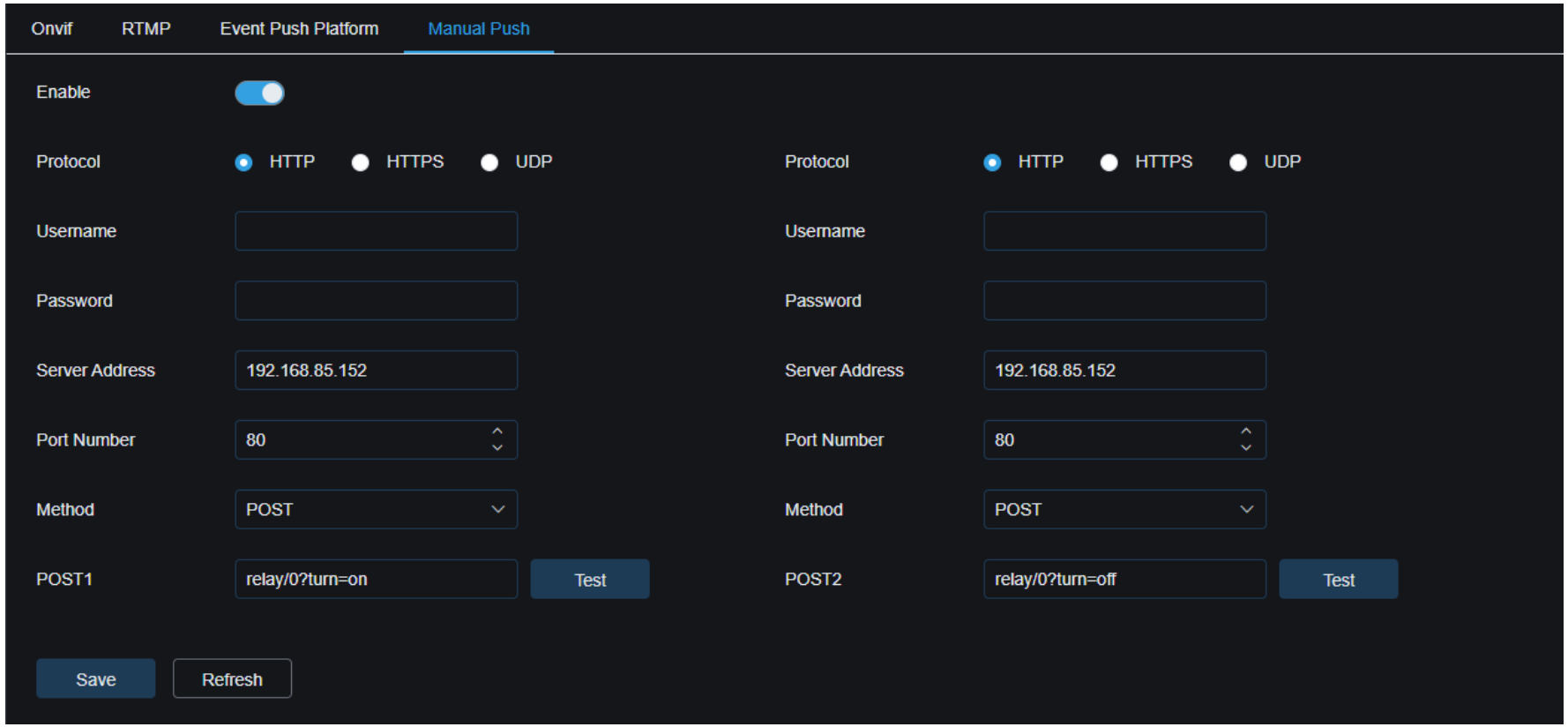
There is an extra feature here which is the **TEST** function, for POST 1 or POST 2.

You can even use the **TEST** to confirm the API command being sent works prior to adding it to the Event Push Platform.

You will see the message **Success !** if the device triggers properly and sends back a HTTP Status Code indicating that everything executed correctly.

You will see the message **Error !** in three different instances :

1. If the device is unreachable because of network issues or wrong network address.
2. If the API command being sent is invalid or has a syntax error.
3. If the device triggers properly and does **not** send back any HTTP Status Code.



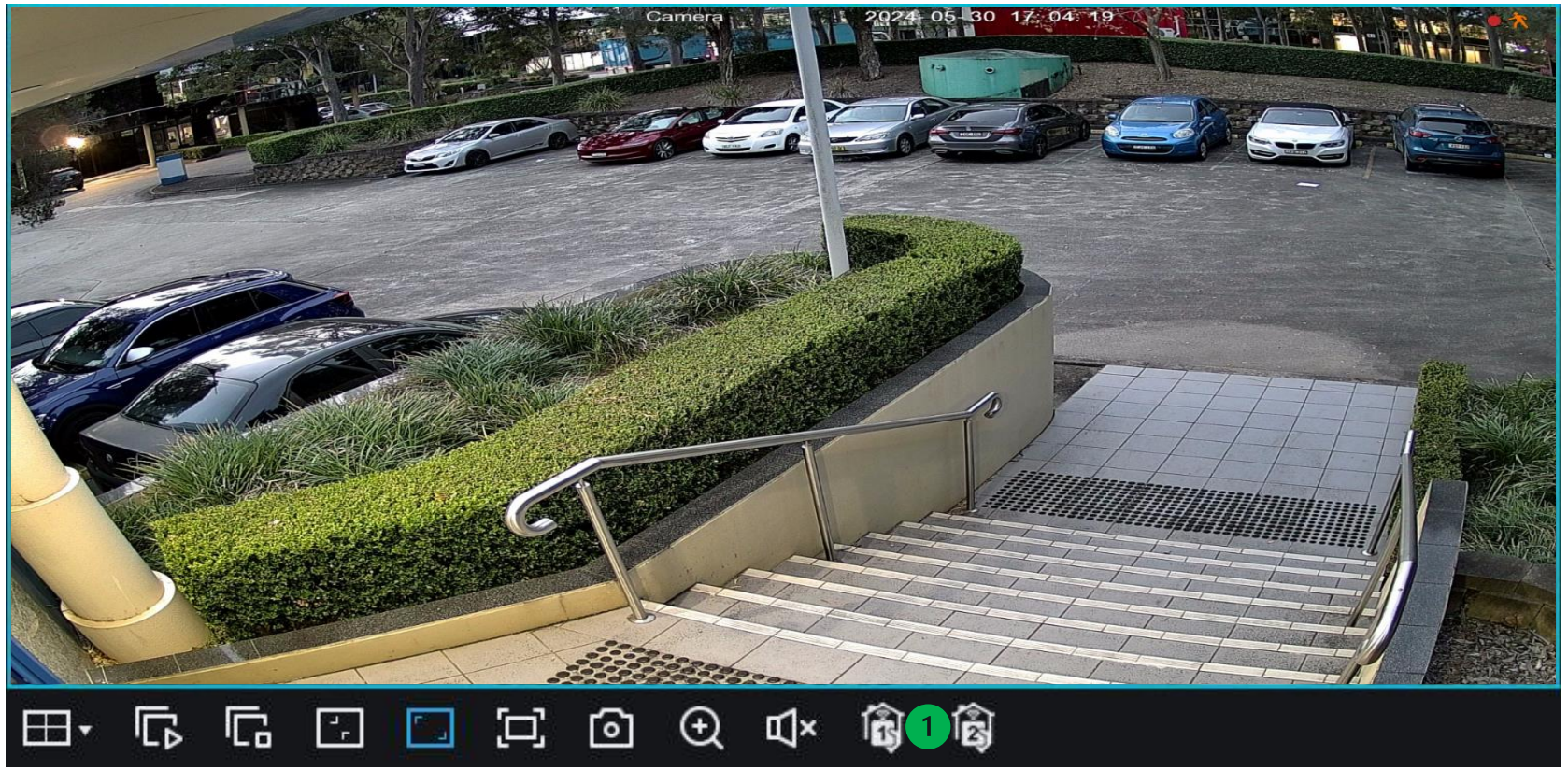
HTTP Manual Push – Triggering – From NVR/Camera

If you would like to trigger one of the two Manual Push Events, log into the NVR or camera via the web interface of NVR interface.

In Live View, click on the camera you are interested in triggering the event for and a light blue box will surround the camera.

1. Two icons will appear at the bottom of the screen, with the numbers 1 & 2 on them.

Clicking these will trigger the corresponding Manual Push Event that has been set up previously.



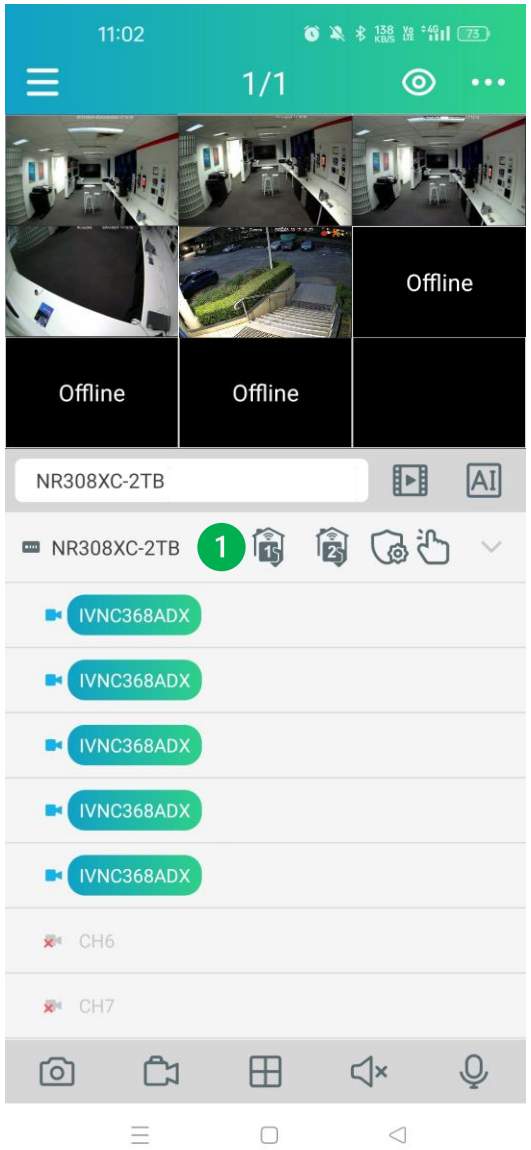
HTTP Manual Push – Triggering – From iVSEC X App

If you would like to trigger one of the two Manual Push Events, open up the latest version of the iVSEC X App.

In Live View, click on the NVR you are interested in triggering the event for and it will expand showing all the cameras.

1. Two icons will appear at the side of the screen, with the numbers 1 & 2 on them.

Clicking these will trigger the corresponding Manual Push Event that has been set up previously.



iVSEC

INTELLIGENT VIDEO SECURITY

Ver 8.2.4