Adding Push Notifications To MotionEyeOS (Formerly MotionPie)

With its lower power consumption and relative portability, the Raspberry Pi makes an ideal candidate for a dedicated home CCTV camera. There are a number of ways to achieve this but when I was looking to setup a CCTV system with my Raspberry Pi, I stumbled on something called **MotionPie**. This had a nice looking user interface and was simple to setup for people who didn't want to get bogged down with editing configuration files or the command line.

Very recently, MotionPie was renamed **MotionEyeOS** because the software worked on other platforms than just the Raspberry Pi (plus the author did not like the name!), so if you heard of MotionPie before but wondered why MotionEyeOS is, now you know!

One benefit of MotionEyeOS is its ability to **detect motion** and then start recording / snapping images and this is great for data storage, but you also have the option of accessing a live stream from your camera in case you wanted to check in every now and then. However, being notified of any motion detection is something that can be very useful, and MotionEyeOS has a very handy option for custom notifications.

However, I found it a little tricky getting everything working in the way I wanted. I had no idea how to mess around with API's or web hooks, but after perservering (and through trial and error!), I managed to figure it out and I decided to share this.

In this guide, **I will assume that you have already set up MotionEyeOS and have it running**. If you need help doing this, then I highly recommend following the **installation** and **configuration** guides created by Calin Crisan, the author and creator of MotionEye OS.

I used the following parts in my Raspberry Pi CCTV system:

- Raspberry Pi (I used a Model B, although it will work on others)
- Raspberry Pi CSI Camera Module (although some of this works with a USB webcam, but I won't cover this)
- Suitable PSU
- Wifi dongle (optional, but easier for portability)

Enabling Push Notifications:

MotionEyeOS has a setting for notifications located under **"Motion Notifications"** and there are various methods you can use, such as email, utilising web hooks or running a custom command. Whilst email is a straight forward way of achieving this, it can be a little long-winded and it no longer feels very modern. I tried using a separate Gmail account for this and soon grew tired of it. I much prefer push notifications and there are a couple of ways to go about this, so I'll explain two for this guide.

1 - Using Pushover's API and a Python Script (my preferred method):

Pushover is an app (for both iOS and Android) which allows push notifications in a variety of different forms. The Pushover API is pretty easy to use (I had no idea what I was doing at the start, but managed to get this working after a few tries) so I wanted to use this to notify me of any motion that was detected.

Pushover is not free to use though - I think I paid £3.99 for the full license, which is money well spent in my opinion as I use it for a number of different set ups at home. The first thing you'll need to do is to create an app over at https://pushover.net/. To do this, click on **"Register Application"** under the **"Your Applications"** heading.

Give your app a name - something like RaspiMotion - and then make sure the *type* is **"Application"**. Give your app a quick description (i.e. *"Push notifications sent by my Raspberry Pi"*) and, if feeling creative, upload a custom icon which will show in your Pushover client app whenever a notification is sent. Here is one I quickly knocked up in Photoshop:

Once you have created your application, you should have access to an **API Token/Key**. This is a

combination of numbers and letters - **please keep this a secret!** You'll also need your **user key**, which is shown once you log in to Pushover.

Okay, now you have an app and your API and user keys. You'll now need to download (or recreate if you so wish) a simple Python script to tell your Pi to work its magic once the script is called upon by MotionEyeOS.

If you wish to clone from my GitHub page, then you'll need this repository: https://github.com/raspberrycoulis/pushover

If you are doing this from the command line, you'll need to cd to the /data director first, then copy and paste the following code. I would typically use Nano to do this, so I would type the following:

```
cd /data
sudo nano pushover.py
```

Then copy and paste the following code:

You can do this using Nano on your Pi, or create it on your computer and then transfer it over to your Pi via SSH. As you will see from the script, you'll need to input your token and user keys to get this working. You can also tweak the message, URL and sound to suit if you so wish. E.g.:

- **Title**: What flashes up on your lockscreen (on your device) when the notification is received
- **Message**: What you will see when you open the notification - there are HTML codes in here (which links in the the HTML:1 section). In my example, "Front Door" will be **bold**

- **URL**: Includes a link in your message - I included the link to the MotionEyeOS UI here - feel free to customise
- **URL Title**: Defines the text that overlays the URL link
- **Sound**: This defines the sound played on notification - a list of these can be found here: https://pushover.net/api#sounds.

You can also specify individual devices using the API too (if you have more than one device), but check out the info on this over at https://pushover.net/api.

Once you have customised the script to include your token and user keys, you will need to transfer

it to your Pi (if not created on there from the command line). I usually use **WinSCP** and drag and drop the file to the correct folder on the Pi.

You need to ensure that your script is stored in the "/data" directory otherwise it won't work.

You will also need to ensure that your new **Python script is executable.** Again, I usually do this through WinSCP by selecting the script, pressing F9 and then changing the permissions to 0755. Alternatively, run the following command over SSH in the data folder:

```
chmod+x scriptname.py
```

Now back over at the MotionEyeOS UI, you need to tell the system to run your new script once it detects motion. To do this, go to the "**Motion Notifications"** menu and then select **"Run A Command"** from the drop down menu. You then need to specify which command to run, which will be the Python script you just created, but sure you use the correct location - i.e. /data/scriptname.py.

It should look something like this:

Now, when your Pi detects motion it should take photos, record videos but also send you a push notification alerting you to this. Quite handy!

2 - Using IFTTT to send you push notifications:

Now if you do not have Pushover or you are not confident enough to experiment with API's and such, there is another method you can try using the very handy **IFTTT.com** (which stands for If This Then That) to do something similar. If you haven't used IFTTT before, you will need to sign up for a **free account** and then activate certain channels to get started.

For notifications on your Pi, these are the channels you'll need to activate:

- Maker (required)

And one of the following push notification apps so that you can actually receive notifications out-and-about:

- Boxcar 2 (recommended if not using Pushover)
- Pushbullet
- Instapush
- IF Notifications (through the IFTTT app)

- Pushalot (Windows only)
- Pushover (if you want to try this route instead of the API route)

You'll need to create free accounts with the above to get things up and running, but this is straight forward so I won't go into that here. **Maker** is a relatively new channel that enables you to create web hooks that can used for various actions. For this project, we are going to use Maker to generate a URL to trigger an event (which will notify you of any detection).

When you activate your Maker channel, you will be given a user key - we will need this.

Now you will need to **create a recipe** on IFTTT. For the **"this"** part, select **Maker** as the **trigger channel**. Then choose **"Recieved a web request"** and you'll be asked for an event name. This can be anything you like, but to keep it simple let's stick with **"motion_detected"**. Finish off by clicking **"Create Trigger"**.

You now need to select an action for the **"that"** part of the recipe, which will be our push notification provider. Each of the providers has a slightly different **action**, but it will be self explanatory - something like **"Send a notification"** or **"Push a note"**. Once you have selected this, you will be able to customise the message that will be sent via push to your client on your smart device (don't forget you'll need to install the client app from the relevant app store!).

Your recipe is effectively saying: "If the Maker channel receives a web request using the event name "motion_detected", then use Boxcar 2 to send a notification."

Once your recipe has been created, we'll need to get your unique URL from the Maker channel that includes your user key and event. You can access this from the Maker channel on IFTTT under **"How to Trigger Events"** and it will look something like this:

https://maker.ifttt.com/trigger/{event}/with/key/youruserkeywillbealonglistoftextlocatedhere

We need to change a few things to get this working, **specifically the {event} part**... Remember the event name you created when making the recipe (motion_detected)? So...

https://maker.ifttt.com/trigger/**motion_detected**
/with/key/youruserkeywillbealonglistoftextlocatedhere

You can test this out on IFTTT through the "How to Trigger Events" page and by typing your event name into the part that says {event} - if you have set up your recipe first (i.e. the Maker channel

and the push action) it should trigger a push notification on your smart device.

Now this link is what we need to use over at MotionEyeOS. Under the "**Motion Notifications**" section, we need to choose **"Web Hook Notifications"** and choose **GET** as the method and then paste your unique link into the box below. All being well, when your Pi detects motion you will get a notification through IFTTT to your chosen push notification app! It should look somthing like this once done:

Once you have adding your Maker URL to MotionEyeOS, apply the changes and you should now get mobile notifications whenever motion is detected!

Next Steps...

My current CCTV system is actually housed inside a dummy CCTV camera. I had intended to include how I achieved this in this guide, but then it would have been far too long! Instead, I will be adding this guide separately so keep an eye out for this to follow soon!

I hope this helps and please experiment with the API on Pushover and also the recipes you can create on IFTTT!