Self Hosted Push Notifications with Gotify and Home Assistant

2019.06.06

Notifications, that bane of modern existence! Most people only have to deal with getting too many. However, if you're running a smarthome or any other kind of moderately complex computer setup you need to decide how you are going to send and receive them too. Many notification systems rely on "trusted" third parties (a.k.a Apple or Google) to handle the delivery of notifications through to the current communications device of choice – the smartphone. Of course, this breaks my fully self hosted ethos and is to be avoided. Luckily it's now possible to achieve self hosted push notifications with Gotify.

The Backstory

Pretty much since starting my smarthome journey (and really even before), I've had trouble with notifications. For a long time email was the goto solution – especially with my nicely self hosted email setup. Then I tried XMPP, then Rocket.Chat (I even wrote the Rocket.Chat notification platform for Home Assistant). There were probably a few more notification systems that I tried and definitely many more that I looked into. Nothing really stuck. Most were too complex to setup and maintain for the benefit that they provided. I mean, who really wants to run a whole chat server just for sending a few notifications?

Then Home Assistant implemented HTML5 notifications. These were cool, but not without downsides. For a start the full capabilities of the platform are only really supported by Chrome on Android. The notifications also go through – you guessed it – Google. However, the notification content is end-to-end encrypted between the HASS instance and the device. You can also do cool things like including images inline and having action buttons to press (actionable notifications).

If I'm honest, HTML5 notifications never really fit into the easy to set up basket either. The set up process is quite involved and requires creation of a project via Google Cloud Services and verification that you own the domain in question. However, once I got it working they worked well for quite some time.

Why can't you get this right, Google?

After a while things started to have problems. First, I would quite often get delayed notifications. Sometimes to the point where the notification would not come through until you picked up and unlocked the phone. This negates the point of the notification entirely! Why is it that my self hosted email server can push messages in real time to k9mail running on the same phone, but Google's own system has issues? I thought GCM/FCM was made of magic that simultaneously allowed it to be more reliable than anything else and consume no battery!?! /s

More Posts

2024.05.25	Facial Recognition Update: Response from Foodstuffs
2024.04.20	Foodstuffs Facial Recognition Trial
2020.03.17	Quick Project: Splitting Docker Compose Projects
2020.03.05	Reconnecting the Web with RSS-Bridge
2020.02.26	Centralised Backups With Restic and Rsync
2020.02.19	Automating My Infrastructure with Ansible and Gitlab CI: Part 3 – SSH Keys and Dotfiles

The next nail in the coffin was Google's incessant pestering and breaking of things. First they wanted me to set up payment details, even though they weren't going to charge me (why?). Then they said they were shutting down the GCM APIs used by my (admittedly somewhat outdated) version of HASS on May 29th this year. I assume that I could have fixed that by updating HASS (which I have since done). However, by this point I'd had enough and shut down the whole thing.

I initially fell back to SMTP/email notifications from HASS, which I still had running for a few lower priority things. However, I was on the look out for a replacement. I'd already heard of Gotify via /r/selfhosted, so I decided to give it a try. Since some of my other projects are starting to pay off and my smarthome is getting smarter, having a reliable notification system is becoming more pressing for me.

Deploying the Server

My ideal notification system would just use MQTT to push notifications to an app running on my phone. This wouldn't require me to set up anything else but the app since I have everything else to support that. However, the designers of Gotify decided to use Websockets so an extra piece of server software is required.

Luckily, this software is written in Go (hence the project name). It also comes in a handy Docker container for easy deployment. Being written in Go makes it both fast and means it consumes barely any resources.

One consideration when deploying this is that you probably want it to be somewhere externally accessible, so that your phone can connect to it when not on your wifi. I installed it on an already accessible host that runs a few other dockerised services. I followed the official instructions, but came up with this to add to my docker-compose file for that server:

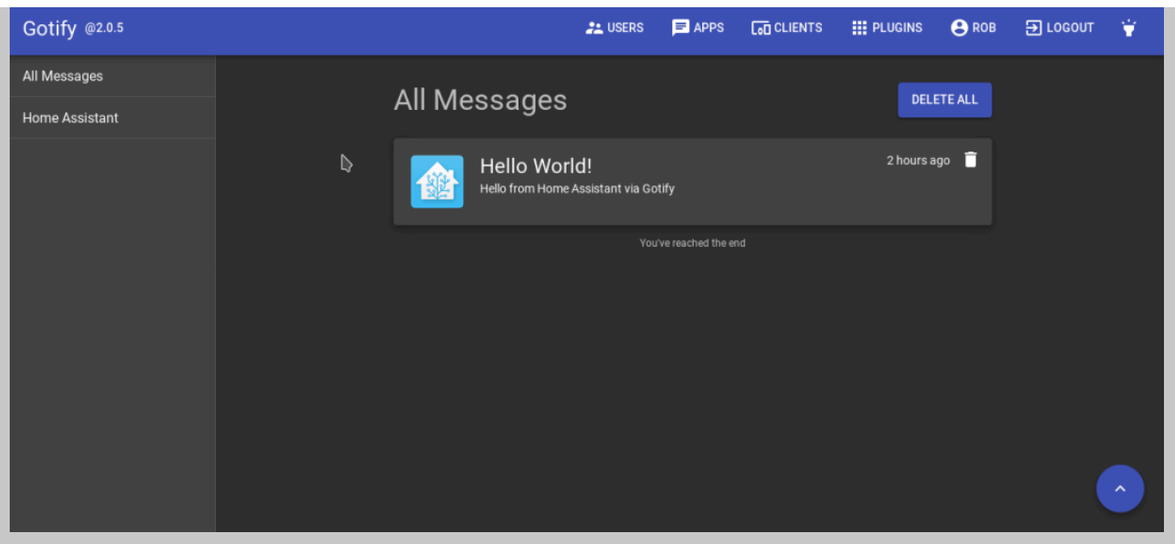
```
gotify:
  image: gotify/server
  volumes:
    - /mnt/docker-data/gotify/data:/app/data
  ports:
    - 9080:80
  restart: always
```

Well, that was easy.

Further configuration can be accomplished via config file or environment variables. However, I found the default settings to be fine for me.

Further Setup

I also needed to set up my reverse proxy to route requests through and set up TLS via Let's Encrypt. I'm not going to go through that here. There are instructions (for the Gotify part) for nginx and apache available. Also, if you've already set up TLS for HASS then you can follow the same process. The Gotify app will show large warnings if you don't use TLS. However, it will allow it so you don't need to do this if you are only doing a bit of local testing.

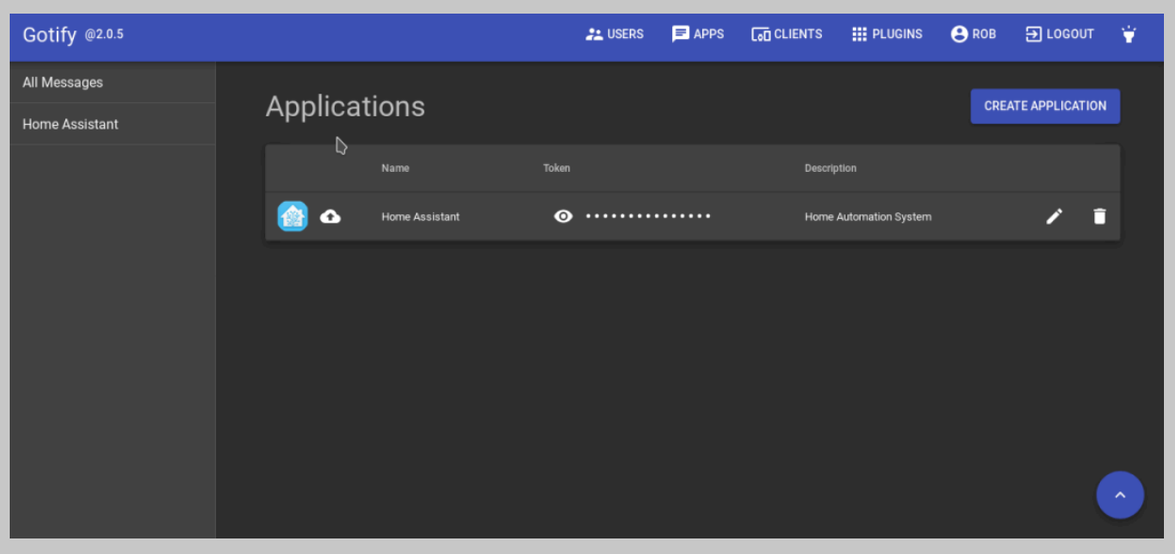


Gotify has a nice web interface, for configuration and receiving/viewing notifications on the desktop

After that I installed the app from F-Droid and added an exception in the battery optimisation page of my Android phone settings. This is different on every phone, but you need to make sure Gotify is listed as “Not optimised”. If you don’t do this Android will kill the app during sleep and you won’t receive notifications. For those that are going to bang on about how this will give you horrible battery life, I haven’t noticed a difference. Admittedly, I was already running a few apps unoptimised, such as k9mail and OwnTracks.

Setting up an Application

Before we can send notifications we need to create an application on the Gotify server. Applications map to individual streams of notifications on the recipient devices. One minor issue is that (as of the time of writing) applications are user specific, there is no way to share an application between users. This isn’t such an issue for us since we will need to set up individual notification platforms in HASS for each user anyway.



Our Application Screen

I set up my application as “Home Assistant” (surprise, surprise). I also uploaded the HASS logo which will be displayed in the notifications. Once the application is configured you will be given a secret token/key that can be used to send notifications via the REST API. You’ll need to copy this for use later.

Configuring Home Assistant

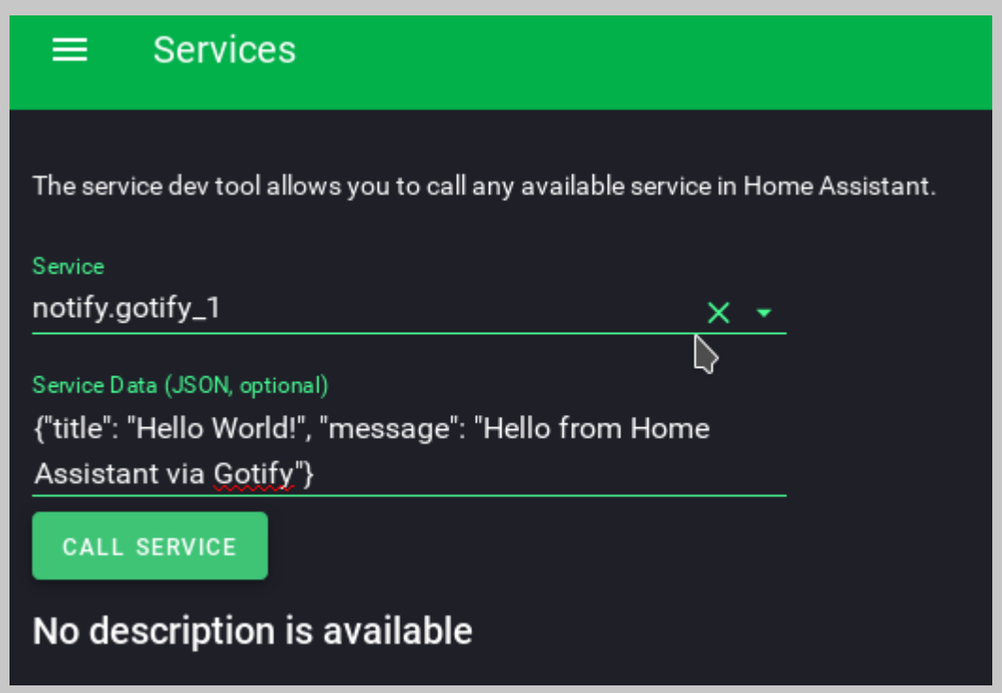
I played around for quite a while sending notifications with cURL as per the documentation and also some more complex messages via the RESTED Firfox addon. However, I’m going to skip straight to how to integrate this will Home Assistant, since that’s probably why you’re here!

Gotify has a simple REST api for sending notifications. Therefore we can use the REST notification platform in HASS to integrate it without a custom component:

```
- name: gotify_1
  platform: rest
  resource: https://my.gotify.server/message
  method: POST_JSON
  headers:
    X-Gotify-Key: !secret gotify_key
  message_param_name: message
  title_param_name: title
```

This goes wherever you have your other notification platforms set up, for me this is in my notify.yaml file. After restarting HASS you should have the `notify.gotify_1` service available. The reason for numbering it is that we will need more notification services to extend this to other users. You'll need to update the `resource` key to the URL of your Gotify server and the `X-Gotify-Key` header value to the key you generated for your application earlier (which I recommend keeping in your secrets.yaml file, as I'm doing).

Sending Notifications

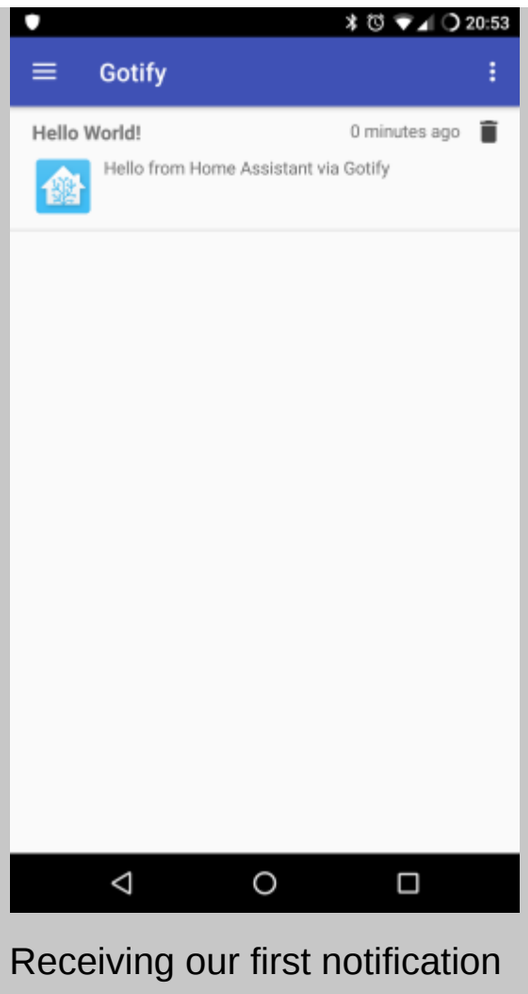


Sending a notification from Home Assistant

We should now be able to send notifications via the services developer tool in Home Assistant. It should be noted that you need to encode the data to send in JSON here for it to work. The data should contain 'title' and 'message' fields, exactly the same as any other notification platform. For example:

```
{
    "title": "Hello World!",
    "message": "Hello from Home Assistant via
Gotify"
}
```

Once you hit the 'CALL SERVICE' button, you should immediately see a notification on your phone from Gotify:



Receiving our first notification

Again, that was easy (wasn't it?).

Advanced Notifications

So far, we can send simple text notifications from Home Assistant via Gotify. However, Gotify also supports sending markdown formatted notifications, which opens up many more options.

To configure this, we edit our REST notification platform to the following:

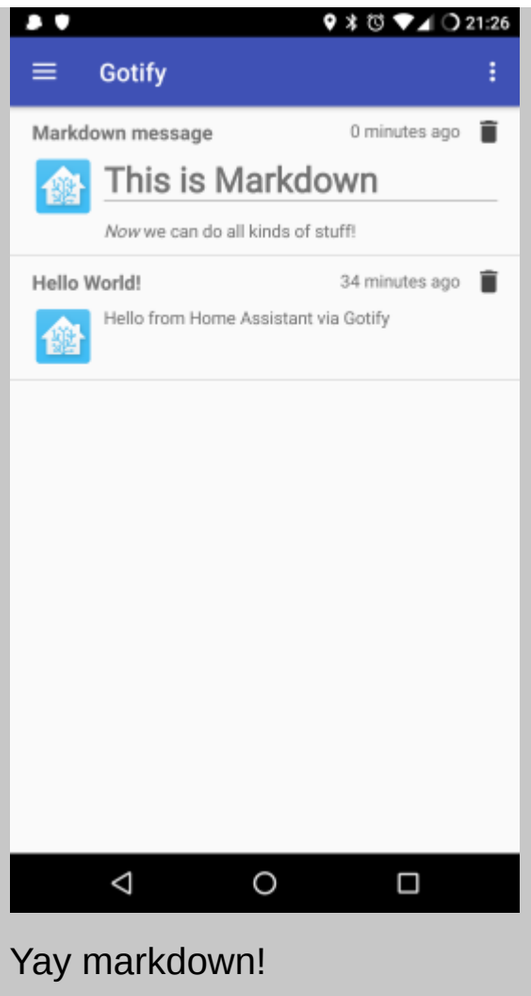
```
- name: gotify_1
  platform: rest
  resource: https://my.gotify.server/message
  method: POST_JSON
  headers:
    X-Gotify-Key: !secret gotify_key
  message_param_name: message
  title_param_name: title
  data:
    extras:
      client::display:
        contentType: "text/markdown"
```

Don't forget to update the `resource` and `X-Gotify-Key` values as before. This updated configuration adds some extra data as per the Gotify documentation to indicate that the message payload should be rendered as markdown.

So let's send a markdown formatted message. In the services dev tool again, select your notification service and use the following data:

```
{
    "title": "Markdown message",
    "message": "#This is Markdown\n*Now* we
can do all kinds of stuff!"
}
```

And you should get:



Neat.

Using that for Something Useful

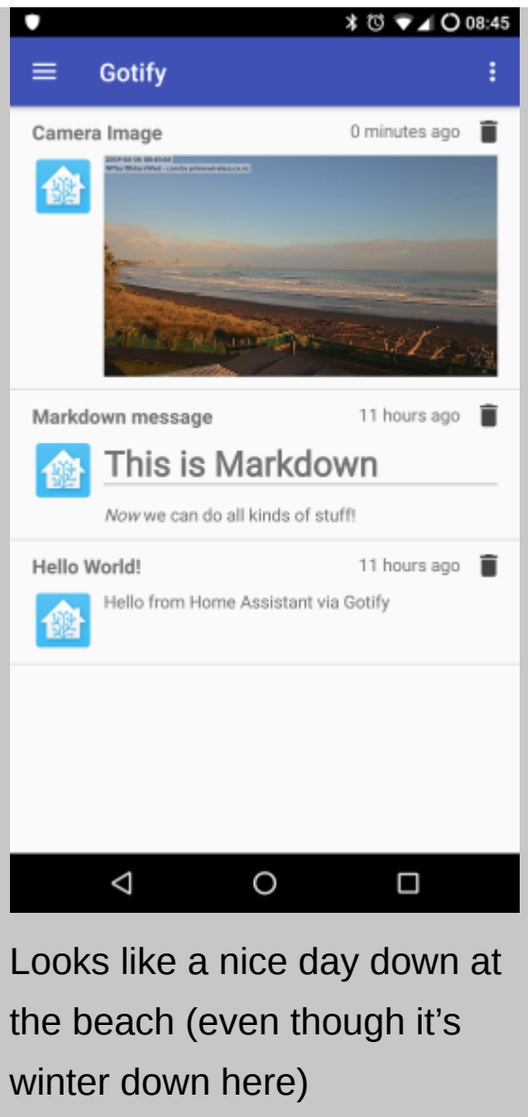
Markdown formatting is all well and good, but not all that useful just for making silly (but well formatted) messages. We'd like to actually put it to good use.

I've thought about including links in various notifications which could be used to trigger a Home Assistant webhook. This could then perform some action, but I'm still not convinced on the usability of it and haven't had a chance to try it out.

One very useful option is including an image in the notification. This is particularly interesting if this image could come from a camera in Home Assistant. As it turns out this is relatively easy thanks to some minor templating:

```
{
    "title": "Camera Image",
    "message": "![Camera Image]
(https://my.hass.server{{
states.camera.beach_webcam.attributes.entity_p
icture }})"
}
```

Here I'm including the latest image from a local beach webcam that I have set up in HASS as a generic IP camera. All we need to do is use the `entity_picture` attribute to get the path of the image on the HASS server and join it to the base URL to build our image source. The resulting message is shown below:



Looks like a nice day down at the beach (even though it’s winter down here)

Obviously, this could be very useful for security alerts, etc.

Conclusion

Overall, I’m pretty impressed with Gotify. Although the project is still young, it works as advertised and I haven’t had any functional issues. There are a few rough edges, but no showstoppers. I’m looking forward to seeing the feature set improve over time. I’d particularly like to see actionable notifications, which would set it up as a full alternative to HTML5 notifications in HASS.

I’ve been able to integrate Gotify into HASS up to the level of it’s current feature set. This is thanks to the well thought out REST API and the flexibility of the REST notification platform in HASS. So far I haven’t needed a custom/official component. As the Gotify API becomes more featureful, it’s likely that a component will be needed in order to unlock it’s full potential. However, as it stands the REST notification platform works just fine.

I’ve had no problems so far with delayed or missed notifications – which is better than Google can do! That in itself is an achievement the Gotify developers should be proud of.

3 responses to “Self Hosted Push Notifications with Gotify and Home Assistant”

sabamimi
2020.01.30

Brilliant, just what I hoping for !
Thanks for sharing.

Niko
2021.11.20

Hey, Thank you for sharing. It’s working great for me. Do you know a way to change the Priority of the notification in Homme assistant? My guess is with the Data field but i can’t get it working.

CJ
2023.07.13

sorry just cannot get the HA service to post anything even though my gotify works fine from nodered

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Post Comment

Search...

Designed with WordPress

Bad Behavior has blocked **290** access attempts in the last 7 days.