MOST VIEWED

- Installing Zabbix for HomeLab Monitoring
- Self-Host Your Browser Data
- Repurposing a Squeezebox
- Discovering & Migrating to OPNsense from pfSense
- Zabbix Email Notifications

POST CATEGORIES

- Home Assistant (4)
- Homelab (13)
  - OPNSense (2)
- Privacy (3)
- Self-Hosting (3)
- Uncategorized (3)

FIND US

Address
123 Main Street
New York, NY 10001
Hours
Monday–Friday: 9:00AM–5:00PM
Saturday & Sunday: 11:00AM–3:00PM

Category: Self-Hosting

Free and OpenSource Photo Libraries

2022-08-25 HOMEGROWNTECHIE

Overview

In my quest to reduce my reliance upon proprietary software applications, I've begun to focus some more time in finding a good Google Photos or Apple Photos alternative. As began looking at the alternatives, I discovered that there were way more options that I had originally anticipated. Each alternative had a different feature set and I found it difficult to compare the different options. To solve this dilemma for myself (and hopefully for many others), I'm compiling a list of free and open source photo libraries that can be self-hosted or run locally without any need for cloud services.

Google/Apple Photos Alternatives

My alternative comparison list looks like the following (Be sure to visit the github repository for the most up to date comparison.

** **This page was last updated on 2023-06-05**

Free and OpenSource Photo Libraries

There are many great free and open-source alternatives to paid photo libraries. This project aims to track and compare the feature set between the many different options with a focus on 'Gratis' (free as in free beer) open source photo libraries. 'Libre' (free as in free speech) projects are also welcome, but will likely need to be submitted via a pull request since the time in testing each different project is significant.

Comparison

✅ = Feature exists in at least a limited fashion
🚧 = Feature may exist but may not be practical or officially rel
❌ = Feature does not yet exist
#️⃣ = Subjective measure of feature quality (on scale of 0-10)
Tip: Hover over icons for missing/incomplete features for more in

Feature	Damselfly	HomeGallery	Immich	Librephotos	Lych
Github Stars	1.4k	775	43k	6.8k	
Active Contributors	1	1	4	2	3
Source Language	C#	JavaScript /	Dart / TypeScript	Python	PHP

Feature	Damselfly	HomeGallery	Immich	Librephotos	Lych
		TypeScript			
License	GPL-3.0	MIT	AGPL-3.0	MIT	MIT
Demo	❌	✅ 6	✅ 6	✅ 5	✅ 4
Freeness	✅ 10	✅ 10	✅ 10	✅ 10	✅ 10
Automatic Mobile Upload	❌	❌	✅ 7	❌	❌
Web App	✅ 8	✅ 8	✅ 8	✅ 8	✅ 8
Android App	❌	❌	✅ 8	✅ 7	❌
iOS App	❌	❌	✅ 8	🚧 3	❌
Desktop App	✅ 9	✅ 8	❌	❌	❌
LivePhotos Support	❌	❌	✅ 9	❌	✅ 6
Video Support	❌	✅ 6	✅ 7	✅ 8	✅ 6
Photo Map	✅ 7	✅ 8	✅ 4	✅ 8	✅ 5
Photo Discovery	❌	❌	❌	✅ 7	✅ 6
Albums	❌	❌	✅ 8	✅ 9	✅ 8
Slideshow	❌	❌	❌	❌	❌
Timeline	✅ 5	✅ 3	✅ 8	✅ 9	❌
Photo Sharing	❌	❌	✅ 4	✅ 9	✅ 9
Photo Search	✅ 8	✅ 7	✅ 7	✅ 8	✅ 5
Duplicate Handling	❌	❌	✅ 6	❌	❌
User Defined Tags	✅ 7	✅ 7	❌	❌	✅ 5
Docker Installation	✅ 8	✅ 8	✅ 7	✅ 7	✅ 7
Object/Face Recognition	✅ 8	✅ 6	✅ 6	✅ 8	❌
Basic Editing	❌	❌	❌	❌	❌
EXIF Data	✅ 9	❌	✅ 7	❌	✅ 7
Multiple User Support	✅ 7	❌	✅ 7	✅ 8	✅ 6

Note: This list is by no means comprehensive. For links to other photo library projects, see the Awesome Self-Hosted list and the Awesome Privacy list.

An HTML version of this comparison table is here: https://meichthys.github.io/foss_photo_libraries/

Contributing

Please contribute additions and corrections! When contributing, please add links to the source of the information. (i.e. link to an issue that indicates that a feature does not exist)

~ Don’t give away your photos to the largest data collection entities in the world! Your photos document your life better than any other kinds of data. Pictures are worth more than a thousand words to advertisers!

UPDATED: 2023-06-05
CATEGORIES: SELF-HOSTING, UNCATEGORIZED TAGS: FOSS, PHOTOS, PRIVACY, SELF-HOSTING

Self-Host Your Browser Data

2021-11-29 HOMEGROWNTECHIE

Overview

For a while now I have been looking for a self-hosted, cross-platform solution that would allow me to sync my browser data (specifically bookmarks and history) between different devices. In the past I've used some of the following but have not been entirely satisfied for a number of reasons:

- iCloud
  - Not self-hosted
  - Required extension (if not using Safari)
  - Did not sync history (if syncing to windows machine)
- Xmarks
  - Not self-hosted
  - Did not sync history
  - Required extension on all browsers
  - Not mobile-friendly
- Floccus
  - ✅ Self Hosted
  - Did not sync history
  - Requires extension on all browsers
  - Not mobile-friendly
  - Great for sharing bookmarks with others (Can use Nextcloud as storage)

Firefox Sync Server

Recently I discovered **Firefox Sync Server** which is an official self-hosted implementation of Mozilla's sync service for syncing all Firefox account information. Although development on this is low priority, I have proved it to be reliable and well worth the effort to setup. Once configured, all my dreams come true:

- ✅ Self Hosted & Free!
- ✅ Cross-platform clients (requires the use of Firefox browsers – which I prefer anyway!)
- ✅ Mobile Friendly
- ✅ Syncs any or all of the following: Bookmarks, History, Tabs, Addresses, Credit Cards, Add-Ons, and Firefox Settings

Configure the Server

There are a few different ways to run the Firefox Sync Server but I found Docker-Compose to be the easiest way to get up and running quickly:

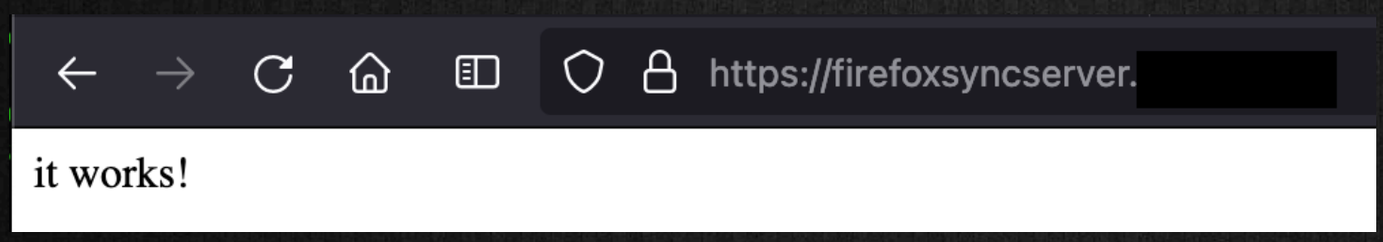
1. Setup docker (not covered in this post)
2. Create a new docker-compose (or stack):

```
version: '3.7'
services:
    syncserver:
        container_name: firefox_syncserver
        image: mozilla/syncserver:latest
        volumes:
            - data:/data
```

```
        ports:
            - 5000:5000
        environment:
            SYNCSERVER_PUBLIC_URL:
'https://firefoxsyncserver.your_domain.com'
            SYNCSERVER_SECRET: 'add_a_random_secret_text'
            SYNCSERVER_SQLURI:
'sqlite:////data/syncserver.db'
            SYNCSERVER_BATCH_UPLOAD_ENABLED: 'true'
            SYNCSERVER_FORCE_WSGI_ENVIRON: 'true'
            PORT: '5000'
        restart: always

volumes:
    data:
```

3. Setup remote access to the service. My preferred way is to use a reverse proxy lik NGINXProxyManager.
At minimum you need:
– A static ip or an externally accessible domain (if you don’t have one, you can get one via DuckDNS)
– Port forward the desired port to your Firefox Sync Server

4. Start/deploy the docker container/stack and navigate to the SYNCSERVER_PUBLIC_URL defined in the compose file to verify that the service is running correctly:



Setup Client Browsers

In order to use your self-hosted Firefox Sync Server you will need to configure each client to use your custom sync server:

Desktop Client

Changing the sync server on Firefox desktop is easy:

1. In your address bar navigate to: about:config

2. Search for: identity.sync.tokenserver.uri and modify the value to match the SYNCSERVER_PUBLIC_URL defined in the compose with an additional path of /token/1.0/sync/1.5

3. Sign into your Firefox Account as normal – This is only to authenticate – not to store your browser data. (You can also host your own Firefox Account Server, but that is out of the scope of this post).

4. Attempt to sync. The sync should take at least a few seconds – if it completes immediately, there may be an issue. To tell if the sync properly saved your browser data to your personal server, you can navigate to about:sync-log and browse the log files to make sure your sync server is being referenced instead of the default firefox sync server.

iOS Client

Changing the sync server on Firefox iOS is also easy:

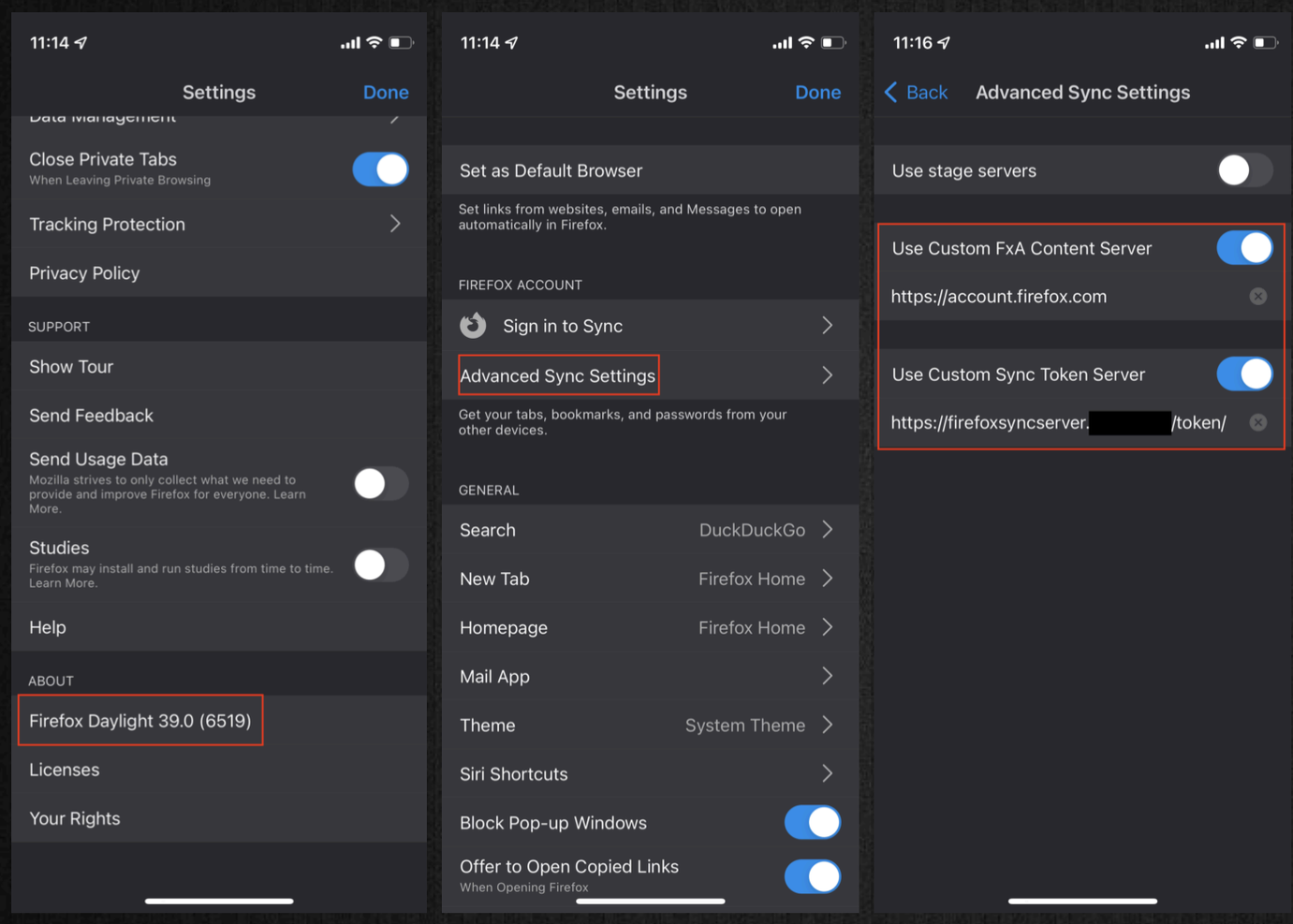
1. Open the iOS Firefox app and navigate to Settings.

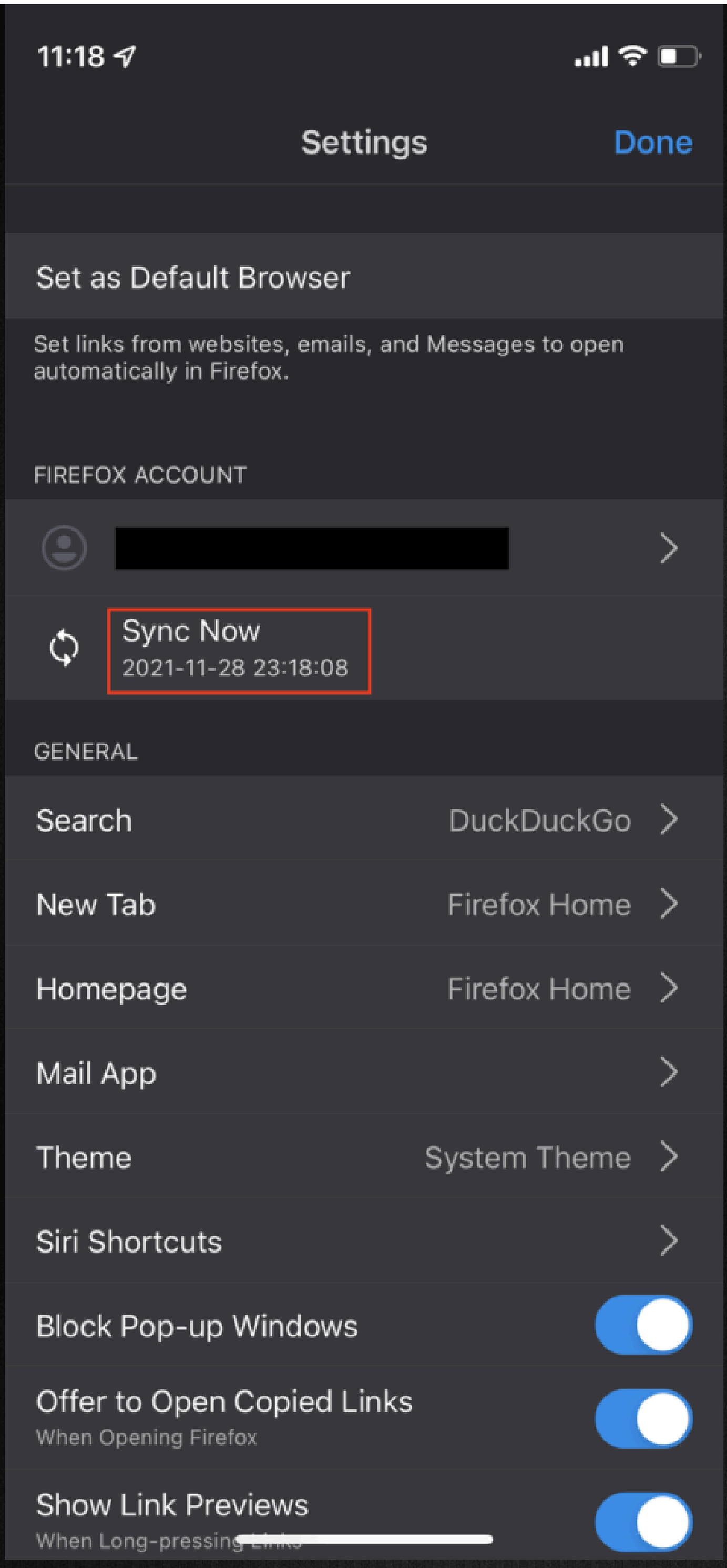
2. Scroll to the bottom of the settings pane and tap on “Firefox Daylight” five times quickly (This will enable the advanced/debug menu):

3. Setup the advanced settings according to the screenshots below (be sure to only include the /token/ path for the token server url – you do not need to additional `/1.0/sync/1.5` path that is needed for Firefox desktop. ALSO instead of the url in the screenshot, use accountS.firefox.org for the FxA server (note the S).

4. Sign into your Firefox Account (again this is only used to authenticate – not to store your browser data).

5. Sync your browser data, and confirm that you can see the changes on your other clients that are synced to your same Firefox Sync Server.





Decloud

Once you feel satisfied that your sync server is working correctly and that you have proper backups in place to prevent data loss, go ahead and remove your other sync solutions like (iCloud, Xmarks, etc) and delete the data stored on any of those cloud services. You're in control of your data now!

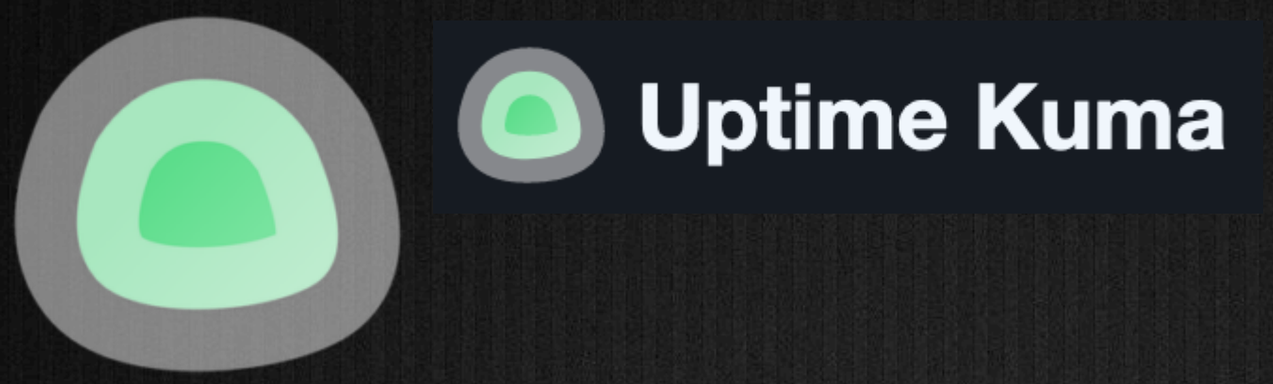
~ Don't litter! – That includes your personal data on the internet!

UPDATED: 2022-09-04

CATEGORIES: PRIVACY, SELF-HOSTING TAGS: BOOKMARKS, BROWSER, BROWSING HISTORY, DOCKER, FIREFOX, FOSS, SYNC

Uptime Kuma

2021-11-09 HOMEGROWNTECHIE LEAVE A COMMENT



Overview

Uptime Kuma is a ‘fancy’ self-hosted monitoring service that can be used to create your very own status page of any services you would like to monitor. Initial configuration and setting up monitors is very easy.

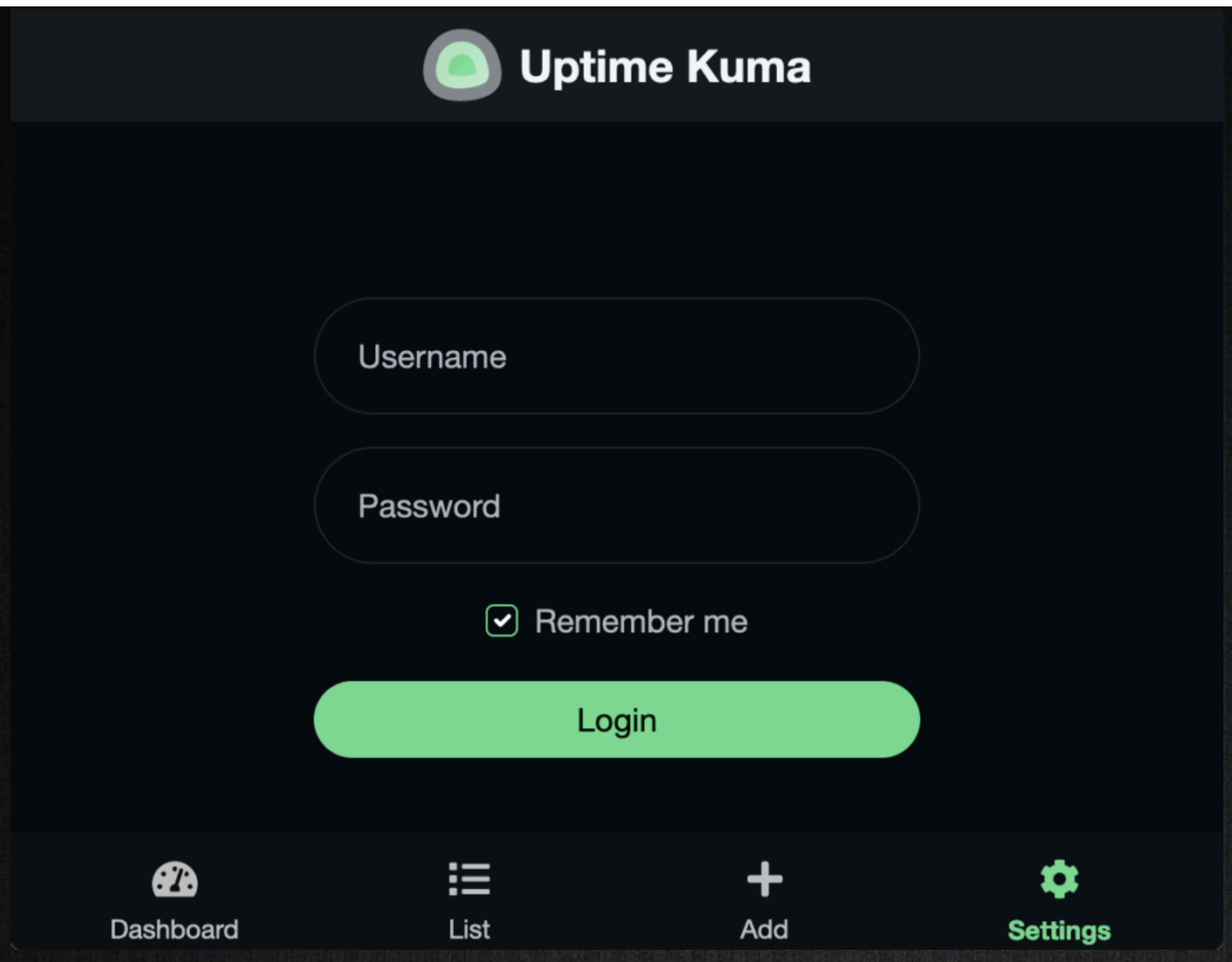
Setting Up Uptime Kuma

The preferred method for setting up Uptime Kuma is Docker. And to make the setup in docker even easier, I like to use Portainer:

Create a new stack in Portainer:

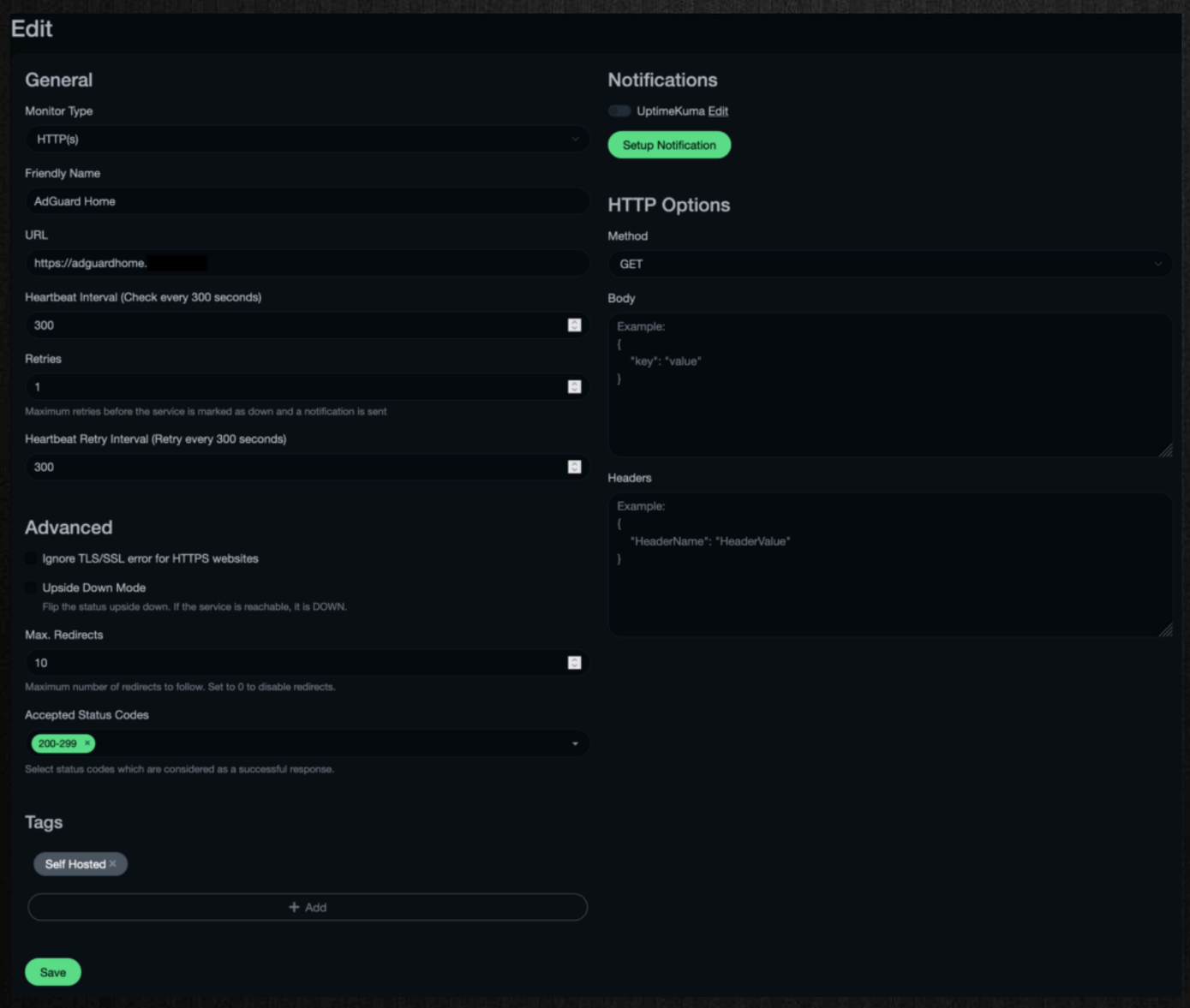
```
version: '3.3'

services:
  uptime-kuma:
    image: louislam/uptime-kuma
    container_name: uptimekuma
    restart: always
    volumes:
      - data:/app/data
    ports:
      - 3001:3001

volumes:
  data:
```

Start the Stack and Log In



Login Page

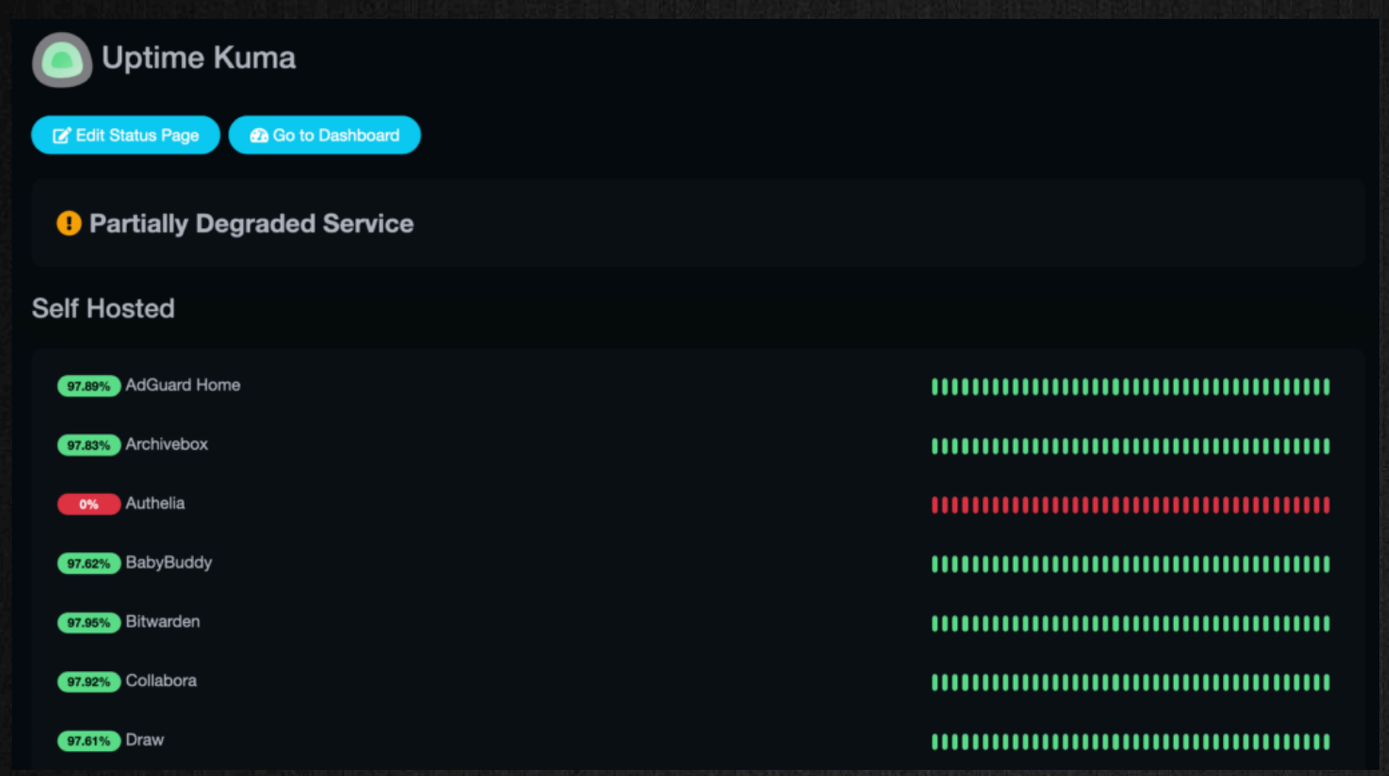
Add Monitors



Sample HTTP Monitor

Bonus

You can use a reverse proxy like **NGINXProxyManager** to fetch an SSL cert and expose the Uptime Kuma service publicly:



CATEGORIES: HOMELAB, SELF-HOSTING TAGS: DOCKER, PORTAINER, SELF-HOSTING, UPTIME KUMA
