

Homebridge

- [Overview](#)
- [Build Custom Homebridge Image](#)
- [Running our Container](#)
- [Our Configuration](#)
- [NGINX Integration](#)
- [Plugins](#)
 - [SonoffTasmotaHTTP Plugin](#)
 - [Sonoff-Tasmota Mqtt Plugin](#)
 - [Cmd4 Plugin](#)
- [References](#)

Overview

Homebridge is a lightweight NodeJS server you can run on your home network that emulates the iOS HomeKit API (Docker Image). It supports Plugins, which are community-contributed modules that provide a basic bridge from HomeKit to various 3rd-party APIs provided by manufacturers of "smart home" devices.

Build Custom Homebridge Image

> docker build -t jmeham/homebridge .

Dockerfile

```
FROM oznu/homebridge:latest

RUN cd /homebridge
RUN npm --prefix "/var/lib/homebridge" add homebridge-myg@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-mqttthing@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-plex-sensors@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-zp@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-weather-plus@latest
RUN npm --prefix "/var/lib/homebridge" add node-appletv-x@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-nest@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-apple-tv-remote@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-tasmota-zbbridge@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-ikea-tradfri-gateway@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-eufy-security@latest
RUN npm --prefix "/var/lib/homebridge" add homebridge-sonos@latest
```

Running our Container

```
#!/bin/bash

NAME=homebridge
IMAGE=jmehlan/homebridge

DIR=`pwd -P`

docker stop $NAME
docker rm $NAME

docker run -d \
--restart=always \
--name $NAME \
--net host \
-e HOMEBRIDGE_INSECURE=1 \
-e HOMEBRIDGE_CONFIG_UI=1 \
-e HOMEBRIDGE_CONFIG_UI_PORT=8091 \
-p 51826:51826 \
-p 51828:51828 \
-v $DIR/data:/homebridge \
$IMAGE
```

Our Configuration

```
{
  "bridge": {
    "name": "Homebridge",
    "username": "XX:XX:XX:XX:XX:XX",
    "port": 51826,
    "pin": "XXX-XX-XXX"
  },

  "description": "This is an example configuration file",

  "accessories": [
    {
      "accessory": "SonoffTasmotaHTTP",
      "name": "sonoff151",
      "hostname": "192.168.1.151",
      "user": "admin",
      "password": "XXX"
    },
    {
      "accessory": "HttpTemperature",
      "name": "Spa Temperature",
      "url": "http://192.168.1.82/info/",
      "max_temp": 200,
      "http_method": "GET",
      "field_name": "temperature",
      "update_interval": "60000",
      "units": "C"
    },
    {
      "accessory": "Sonos",
      "name": "Katie's Speaker",
      "room": "Katie's Room",
      "mute": false
    },
    {
      "accessory": "mqttthing",
      "type": "lightbulb",
      "name": "teamroom",
      "url": "http://XXXX:1883",
      "topics":
```

```

        {
            "getOn": "ikea",
            "setOn": "ikea"
        },
        "onValue": "on",
        "offValue": "off"
    }
],
"platforms": [
    {
        "platform": "Nest",
        "clientId": "XXX",
        "token": "XXXX",
        "clientSecret": "XXX",
        "code": "XXX"
    },
    {
        "platform": "Cmd4",
        "name": "Cmd4",
        "accessories": [
            {
                "type": "TemperatureSensor",
                "name": "sonoff158",
                "timeout": 3000,
                "polling": false,
                "interval": 100,
                "stateChangeResponseTime": 10,
                "state_cmd": "sh /homebridge/Cmd4Scripts/sonoff158.sh"
            }
        ]
    },
    {
        "platform": "HttpWebHooks",
        "webhook_port": "51828",
        "cache_directory": "/homebridge/.node-persist/storage",
        "sensors": [
        ]
    }
]
}

```

What does this give us?

- Homebridge running on port 51826
- Homebridge UI listening on port 8089
- homebridge-http-webhooks listening on port 51828

NGINX Integration

In order to access the homebridge ui via an nginx reverse proxy, you will need a configuration like the following:

```

server {
    server_name  homebridge homebridge.jmehan.com;
    location / {
        proxy_pass          http://192.168.1.60:8089/;
        proxy_http_version  1.1;
        proxy_buffering      off;
        proxy_set_header     Host $host;
        proxy_set_header     Upgrade $http_upgrade;
        proxy_set_header     Connection "Upgrade";
        proxy_set_header     X-Real-IP $remote_addr;
        proxy_set_header     X-Forward-For $proxy_add_x_forwarded_for;
    }
}

```

Plugins

We can see what plugins are available for homebridge from the following url:

<https://www.npmjs.com/search?q=homebridge-plugin>

SonoffTasmotaHTTP Plugin

config.json

```

{
    "accessory": "SonoffTasmotaHTTP",
    "name": "sonoff97-1",
    "hostname": "192.168.1.97",
    "relay": "1",
    "user": "xxx",
    "password": "xxx"
},
{
    "accessory": "SonoffTasmotaHTTP",
    "name": "sonoff98",
    "hostname": "192.168.1.98",
    "user": "xxx",
    "password": "xxx"
},
...

```

Sonoff-Tasmota Mqtt Plugin

config.json

```

...
{
    "accessory": "mqtt-switch-tasmota",
    "name": "sonoff97-2-relay",
    "url": "mqtt://192.168.1.60",
    "username": "john",
    "password": "pass",
    "topics": {
        "statusGet": "stat/sonoff97-2/POWER",
        "statusSet": "cmd/sonoff97-2/POWER"
    }
},
...

```

Cmd4 Plugin

Here is an example that communicates to the nest developer api and pulls the humidity for a thermostat. It uses jq to parse the json returned.

sonoff158.sh

```
#Humidity

if [ "$1" = "Get" ]; then
    # $2 would be the name 'spa'
    # $3 would be the charactersistic

    if [ "$3" = "CurrentRelativeHumidity" ]; then

        humidity=`curl -v --location-trusted \
        -H "Content-Type: application/json" \
        -H "Authorization: Bearer xxx" \
        -X GET "https://developer-api.nest.com/" 2>/dev/null \
        | jq '.devices.thermostats."oFh5M-WlyNJjd1COACDoCrsH5z1eG7kM".humidity'`

        echo $humidity
        exit 0
    fi
    if [ "$3" = "StatusActive" ]; then
        echo "1"
        exit 0
    fi
fi

if [ "$1" = "Set" ]; then
    exit 1
fi

exit 0
```

References

Reference	URL
Homebridge	https://github.com/nfarina/homebridge
Homebridge Plugins	https://www.npmjs.com/search?q=homebridge-plugin
How to make Homebridge Plugins	http://blog.theodo.fr/2017/08/make-siri-perfect-home-companion-devices-not-supported-apple-homekit/
Sonoff-Tasmota Mqtt Plugin	https://github.com/MacWyznawca/homebridge-mqtt-switch-tasmota
Sonoff-Tasmota HTTP Plugin	https://www.npmjs.com/package/homebridge-sonoff-tasmota-http
Tasmota Commands	https://github.com/arendst/Sonoff-Tasmota/wiki/Commands
Tutorial	http://www.instructables.com/id/HomeKit-Enabled-Arduino-ESP8266-Self-Powered-110v-/
Docker Hub Image	https://hub.docker.com/r/oznu/homebridge/
Cmd4 npm plugin	https://www.npmjs.com/package/homebridge-cmd4
Cmd4 Git Repository	https://github.com/ztaibot2000/homebridge-cmd4
Cmd4: Sample JS script that uses all supported homekit devices.	https://github.com/ztaibot2000/homebridge-cmd4/blob/master/Extras/Cmd4Scripts/State.js
Cmd4: Sample Homebridge config file showing all supported homekit devices	https://github.com/ztaibot2000/homebridge-cmd4/blob/master/Extras/config.json