

MACBULB HOMEKIT ADVENTURES

SEBASTIAN OSIŃSKI

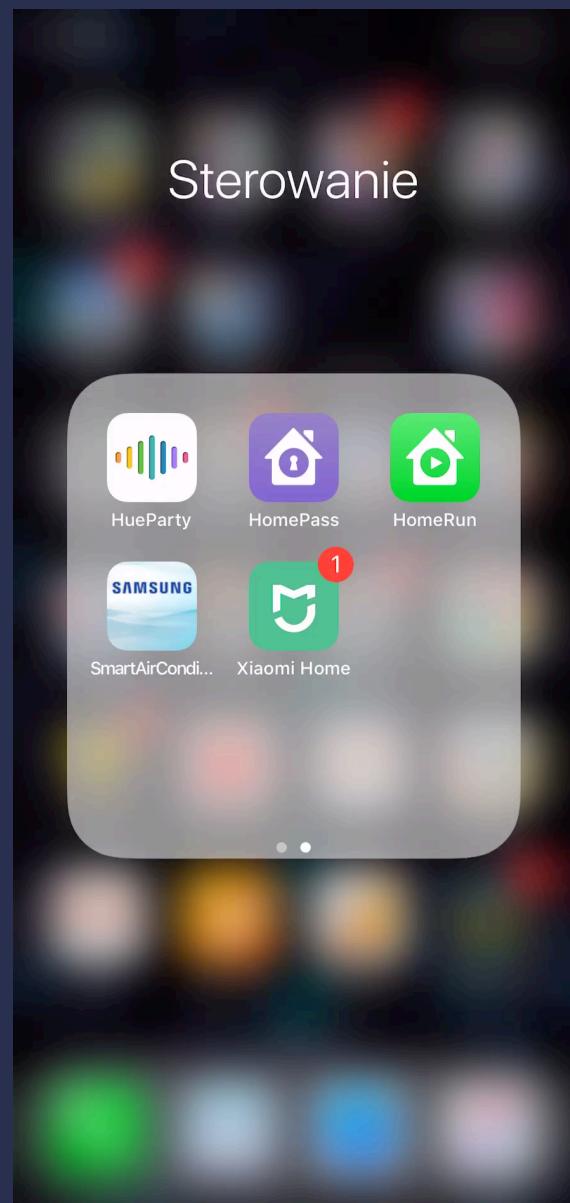
ABOUT ME

IOS DEVELOPER AT

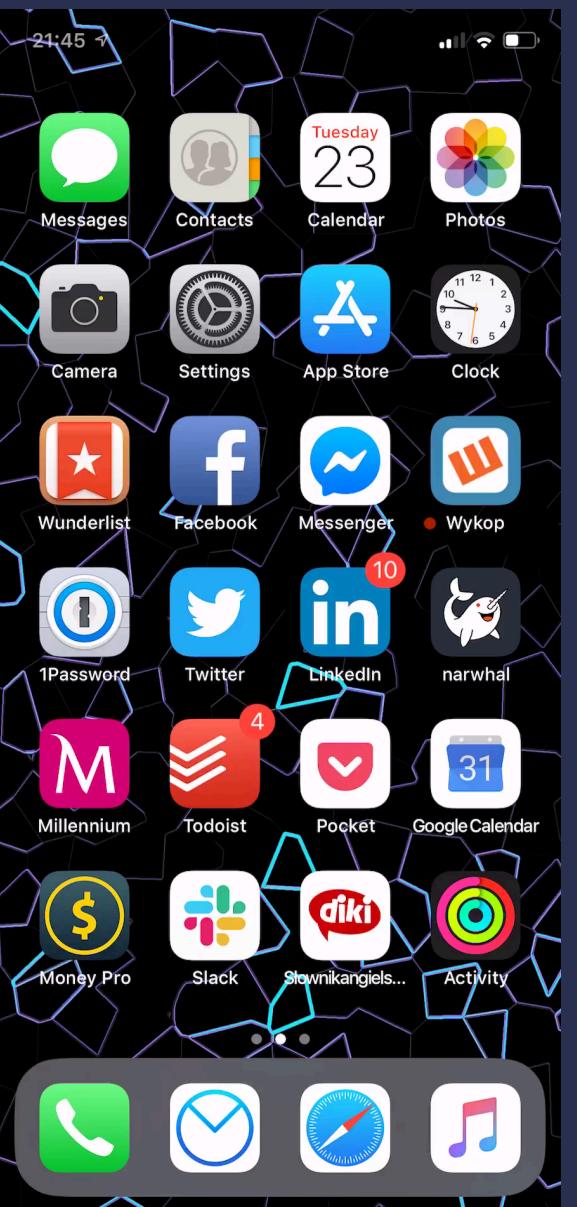


**EVERYTHING STARTED WITH
ONE ANGERED DEVELOPER...**

FROM THIS...



TO THIS!



CREATING HOMEKIT ACCESSORY

TWO WAYS:

CREATING HOMEKIT ACCESSORY

TWO WAYS:

- ▶ Commercial

CREATING HOMEKIT ACCESSORY

TWO WAYS:

- ▶ Commercial
- ▶ Non-commercial

COMMERCIAL

Accessory Manufacturers

If you're interested in developing or manufacturing a HomeKit accessory that will be distributed or sold, your company must enroll in the MFi Program.



NON-COMMERCIAL

Hobbyists and Makers

Learn and explore how to create accessories for your home using the HomeKit Accessory Protocol.

-  [HomeKit Accessory Protocol Specification](#)

*Non-commercial version

-  [HomeKit Accessory FAQs](#)

-  [HomeKit Accessory Forum](#)

HOMEKIT ACCESSORY PROTOCOL SPECIFICATION

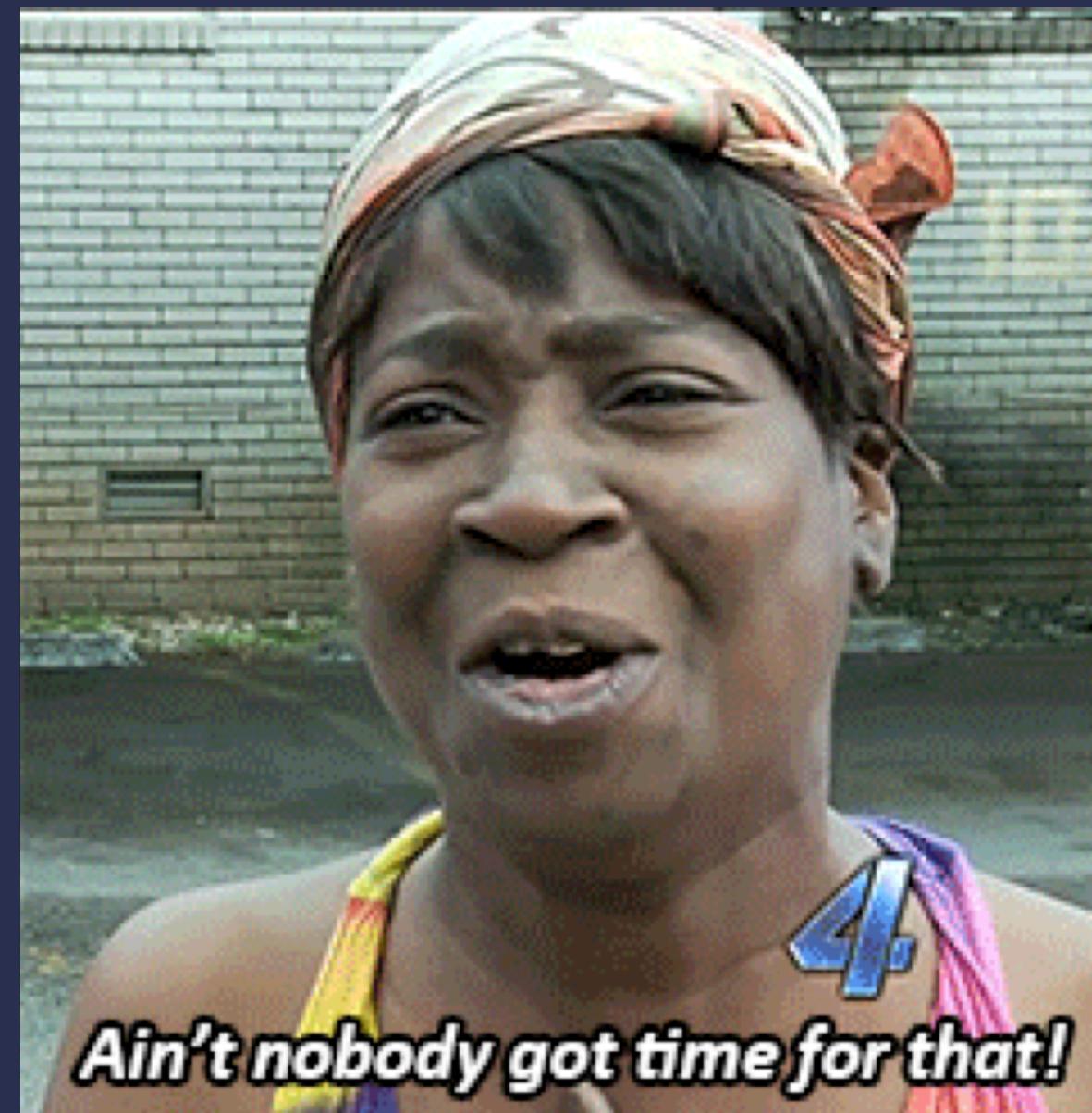
HomeKit Accessory
Protocol Specification
(Non-Commercial Version)

Release R1



Page 1 of 256

HOMEKIT ACCESSORY PROTOCOL SPECIFICATION



HOMEBRIDGE TO THE RESCUE!

HOMEBRIDGE

HOMEBRIDGE

- ▶ Allows to 'HomeKitify' every device which has some kind of interface to interact with

HOMEBRIDGE

- ▶ Allows to 'HomeKitify' every device which has some kind of interface to interact with
- ▶ Most of the time installed on Raspberry Pi

HOMEBRIDGE

- ▶ Allows to 'HomeKitify' every device which has some kind of interface to interact with
 - ▶ Most of the time installed on Raspberry Pi
- ▶ Based on HAP-NodeJS - HomeKit Accessory Protocol implementation

HOMEBRIDGE

- ▶ Allows to 'HomeKitify' every device which has some kind of interface to interact with
 - ▶ Most of the time installed on Raspberry Pi
- ▶ Based on HAP-NodeJS - HomeKit Accessory Protocol implementation
- ▶ Based on system of plugins - adding support for some device is done by writing plugin - NPM module

HOMEKIT BASICS

HOME

Place with accessories, just that ;)

ROOM

Contains group of accessories in given Home. One accessory can belong to only one Room.

ACCESSORY

Represents single device.

SERVICE

Single functionality of an accessory. One accessory can have multiple services, e.g. be a bulb and light sensor simultaneously.

CHARACTERISTIC

Single property of a service. E.g. bulb has on/off characteristic and, among many others, brightness characteristic.

- ▶ Readable
- ▶ Writable
- ▶ Observable

BRIDGE

Special kind of accessory which enables access to other accessories which are not compatible with HomeKit directly.

PLUGIN DEVELOPMENT STEPS

PLUGIN DEVELOPMENT STEPS

- ▶ Create NPM module which name starts with "homebridge"

PLUGIN DEVELOPMENT STEPS

- ▶ Create NPM module which name starts with "homebridge"
- ▶ Find appropriate service(s) for your device

PLUGIN DEVELOPMENT STEPS

- ▶ Create NPM module which name starts with "homebridge"
- ▶ Find appropriate service(s) for your device
 - ▶ Code!

EXAMPLE

MACBULB

Simple MacOS app with REST API:

MACBULB

Simple MacOs app with REST API:

- ▶ GET/PUT power

MACBULB

Simple MacOs app with REST API:

- ▶ GET/PUT power
- ▶ GET/PUT hue

MACBULB

Simple MacOS app with REST API:

- ▶ GET/PUT power
- ▶ GET/PUT hue
- ▶ GET/PUT saturation

MACBULB

Simple MacOS app with REST API:

- ▶ GET/PUT power
- ▶ GET/PUT hue
- ▶ GET/PUT saturation
- ▶ GET/PUT brightness

CREATING PLUGIN

```
mkdir homebridge-macbulb; cd homebridge-macbulb
```

```
npm init
```

```
npm link
```

ADJUSTING PACKAGE.JSON

```
{  
  "name": "homebridge-macbulb",  
  "version": "1.0.0",  
  "description": "",  
  "keywords": [  
    "homebridge-plugin"  
,  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
,  
  "author": "Sebastian Osiński",  
  "license": "ISC",  
  "engines": {  
    "node": ">=0.8",  
    "homebridge": ">=0.1"  
,  
  "dependencies": {  
    "request": "^2.88.0"  
,  
  }  
}
```

PLUGIN ENTRYPPOINT

```
module.exports = function(homebridge) {
  Service = homebridge.hap.Service;
  Characteristic = homebridge.hap.Characteristic;
  homebridge.registerAccessory("homebridge-macbulb", "MacBulb", MacBulb);
};

function MacBulb(log, config) {
  this.log = log;
  this.name = config.name;
}
```

PLUGIN'S "HEART"

```
MacBulb.prototype = {
  getServices() {
    const bulbService = new Service.Lightbulb(this.name);

    bulbService
      .getCharacteristic(Characteristic.On)
      .on('get', this.getOn.bind(this))
      .on('set', this.setOn.bind(this));

    bulbService
      .getCharacteristic(Characteristic.Hue)
      .on('get', this.getHue)
      .on('set', this.setHue.bind(this));

    ... // Saturation and Brightness characteristics

    return [bulbService];
  }

  ...
}
```

```
getHue(callback) {
  request
    .get(baseUrl + "/hue")
    .on('data', function(data) {
      json = JSON.parse(data)
      callback(null, json["value"])
    })
    .on('error', callback);
} ,

setHue(level, callback) {
  request
    .put(baseUrl + "/hue")
    .form({value: level})
    .on('response', function(response) {
      callback(null);
    })
    .on('error', callback);
}
```

DEMO!

THANKS!

THIS PRESENTATION CAN BE FOUND HERE:

<https://osinski.dev/talks>

SOURCES

- ▶ <https://developer.apple.com/homekit>
- ▶ <https://homebridge.io>
- ▶ <https://github.com/KhaosT/HAP-NodeJS>

QUESTIONS?