

Smart Home Adapters

Usability Guide

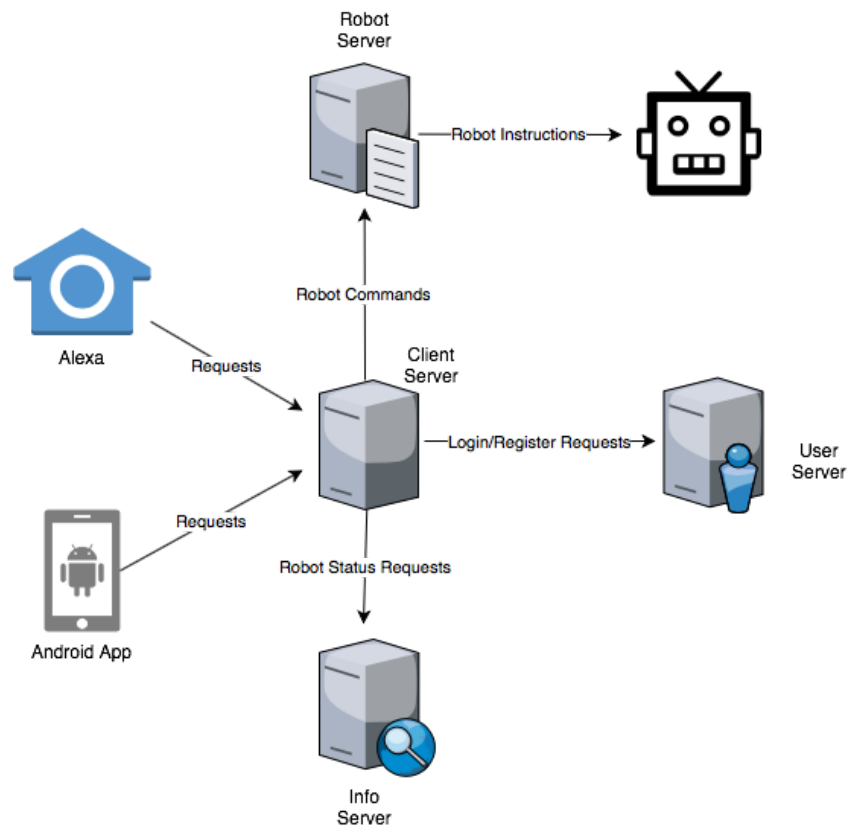
General Overview	3
Hardware	4
Hardware Overview	4
HAL (external overview)	4
HAL (internal overview)	4
Light switch PAL	5
Thermostat PAL	5
Bolt lock PAL	6
Hardware Setup	6
Installing the Switch and Bolt lock PAL	6
Installing the Thermostat PAL	7
Charging a HAL	7
Software	7
Android app installation (APK)	8
Prerequisites	8
Installation	8
Using the Android app	8
Signing up and logging in	8
Registering a new robot to your account	9
Editing an existing robot	10
Using your Smart Home Adapters	10
Alexa integration set-up	11
Using Alexa	11
Troubleshooting Guide	12
Appendix - Advanced Installation Guides	13
Android app installation (source)	13
Prerequisites	13
Installation	13
Microservice build and deployment	14
Prerequisites	14
Installation	14
Alexa lambda function deployment	15
Prerequisites	15
Installation	15
Flashing custom WiFi credentials unto the HAL	18
Prerequisites	18
Flashing	18

General Overview

HalsPals' Smart Home Adapters are a range of Internet-of-Things devices that transform your existing appliances into smart devices. Using our free Android app you can then easily control your devices remotely and accurately, thanks to our state-of-the-art microservices server architecture.

To allow for unrivaled flexibility, our adapters come in two parts: the HAL and the PAL. The HAL is a universal device which contains the servo and circuitry that makes the device function. The HAL can then be connected to any compatible PAL, which is an attachment that carries out a specific task. Our current range of prototype PALs includes:

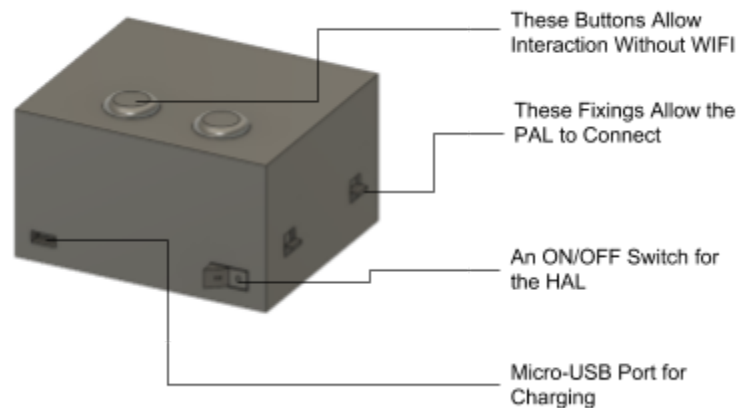
- Switch PAL - Controls a variety of switches, such as those used for lights
- Thermostat PAL - Manipulates a thermostat dial
- Bolt Lock PAL - Slides a bolt left/right to lock/unlock



Hardware

Hardware Overview

HAL (external overview)



HAL (internal overview)

Inside the HAL you will find the rechargeable battery, servo motor and PCB. The servo is what moves the PALs to perform the appropriate operation; the PCB, the design of which you can find below, controls the HAL's logic.

SW1 - OFF, connects GPIO16 to RST to enable the esp's deep sleep function. ON, Allows GPIO16 to be used as a IO pin.

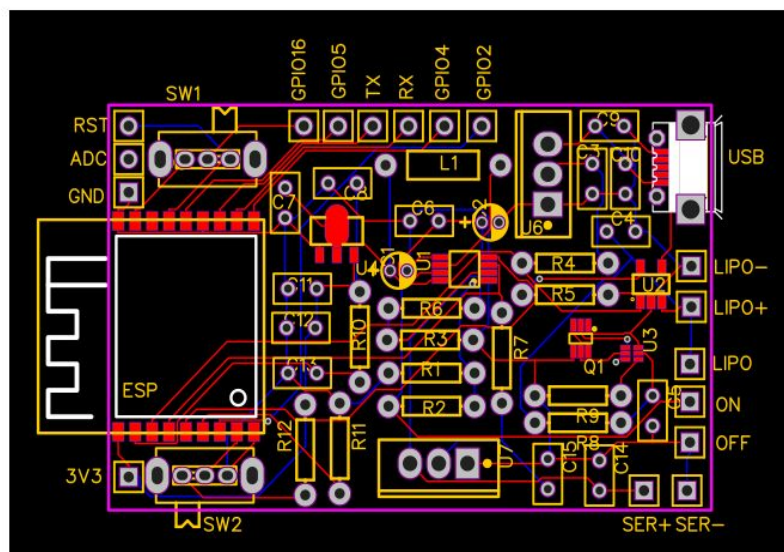
GPIO5 and GPIO4 - IO pins connected to the external buttons on the HAL.

TX and RX - Used for re-flashing the ESP.

GPIO2 - IO pin connected to servo for PAL control.

ADC - Pin available to connect to sensor.

SW2 - OFF, ESP, boots in flash mode. ON, ESP boots in function mode.



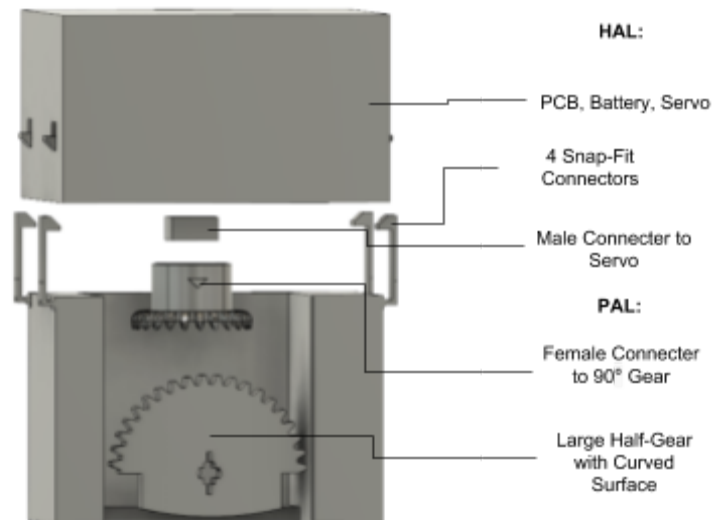
USB - Micro usb port for recharging.

LIPO- and LIPO+ : Pins to connect to re-chargable battery.

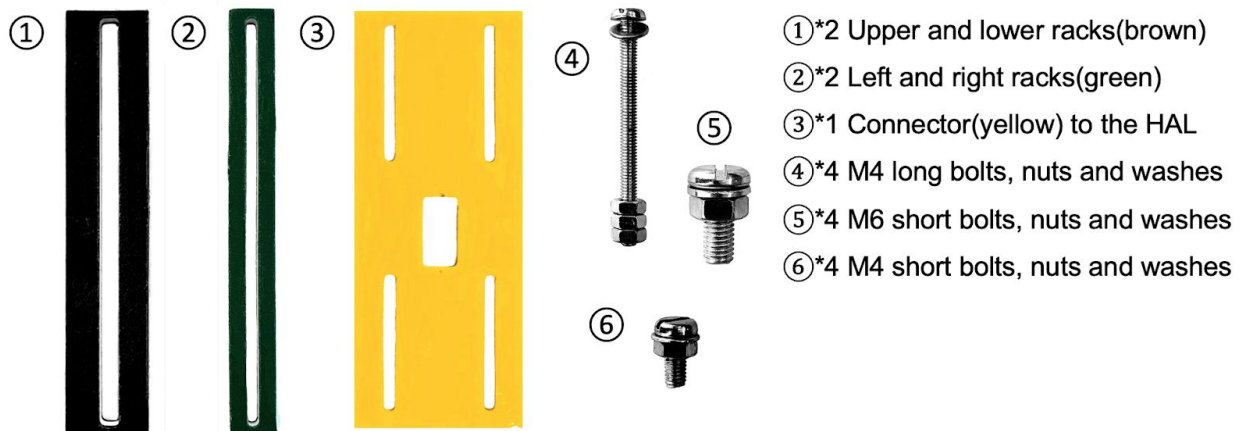
LIPO, ON and OFF - Pins connected to external switch to turn device on or off.

SER+ and SER- : Pins connected to power the servo.

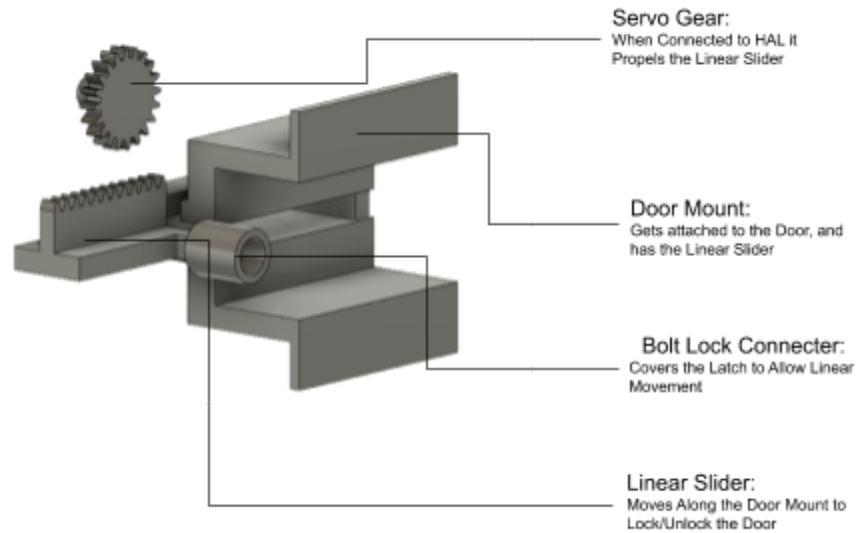
Light switch PAL



Thermostat PAL



Bolt lock PAL



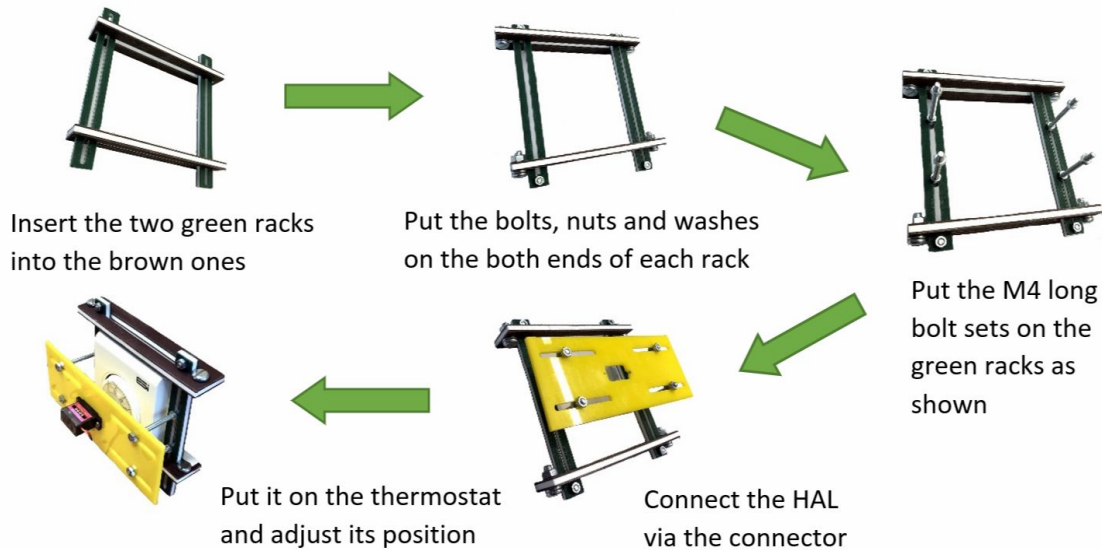
Hardware Setup

You will most likely have been given a HAL with pre-flashed WiFi credentials (e.g. for Eduroam); in this case there is no further set up needed outside of installing the PALs. If not, you will need to actually flash the WiFi credentials onto the HAL yourself, due to building regulations preventing us from setting up local hotspots. This requires some fiddling and we only recommend you doing so if you are an advanced user; the instructions can be found at the end of the appendix. If you do not feel comfortable doing this, please contact us regarding giving you a pre-flashed HAL instead.

Installing the Switch and Bolt lock PAL

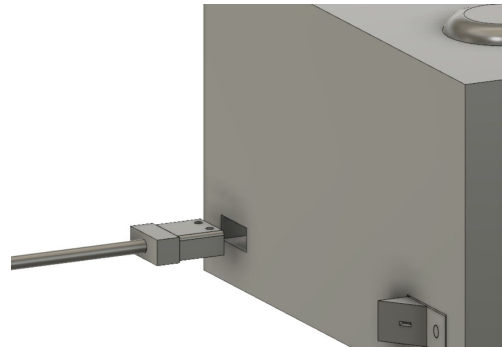
1. Peel the strip of the sticky tape and affix the robot to the device
2. Push down for 30 seconds
3. Proceed to the app and follow the prompts to calibrate the attachment
4. After calibration you will be prompted to connect the HAL to the PAL

Installing the Thermostat PAL



Charging a HAL

To charge a HAL simply connect the provided charger as shown. The HAL will notify you via the app once it is fully charged and ready to be used.



Software

The Smart Home Adapter software is comprised of two parts: a collection of microservices (servers), and an Android app. Here we will describe how to install the app from a pre-compiled *Android Application Packet*, which will then use the microservices which we host on our website. This is the typical way you as a customer will use our software, and also makes the process very simple; if you are an advanced customer who wants to modify the software or who wants to host a local version of the microservices yourself, please refer to the more advanced details in the appendix.

Android app installation (APK)

Prerequisites

- Android OS version Marshmallow or later (this may also be called *version 6.0* or *API level 23*)

Installation

1. Go to <https://halspals.co.uk/download/android> and download the latest *smarthomeadapters_android.apk*.
2. Navigate to the *Downloads* folder on your phone and tap the file.
3. *If* you are asked to confirm that you want to allow this APK to be installed, press yes.
4. Done; see below for how to use the app once installed.

Using the Android app

Signing up and logging in

The first time you launch the app you can press *Sign in or register* to be taken to an authorization website in the app. Here you can securely enter your credentials and either create a new account or log in to an existing one; after having successfully validated your inputs the app will sign you in and take you to the main screen. The *Smart Home Adapters* app supports *Single-Sign-On* functionality by default, so unless you manually sign out of the app by navigating to *Settings > Sign Out* you will not be asked to sign in again.



Registering a new robot to your account

To register a new robot to your account, open up the app's main screen and press the button which looks like a big plus. Doing so will launch the *Robot Registration Wizard*, which will guide you through the following steps:

1. *Scanning or entering the HAL's unique ID and giving it a nickname.*

The nickname entered here will help you keep track of what each of your robots does and does not need to be unique. The HAL's ID is guaranteed to be unique and by registering it to your account no one else will be able to take control of your robot; you can enter this either by scanning the QR code attached to the HAL or by manually entering the code found directly under the QR code.

Scan QR

Name Your Device

Camera

Find the QR code on your smart home adaptor device and align it with the camera

Device name
Fantasti

MANUAL ENTRY

CONTINUE

CANCEL SETUP

CANCEL SETUP

2. *Choosing an attachment for the device.*

Here you should indicate which PAL attachment you wish to use with this HAL.

Select Attachment

Thermostat
Control the thermostat temperature

Switch
Turn a switch on and off

Bolt lock
Unlock and lock a bolt lock

3. *Configuring the device.*

The exact process involved depends on the attachment you've chosen:

- a. *Switch:* You need to set the angle which the servo should move to when turning the light on (in the range 0-90°) and the angle it should move to when turning it off (in the range 90-180°).
- b. *Thermostat:* You need to set the unit of temperature (Kelvin, Celsius, or Fahrenheit), the minimum and maximum temperatures supported by your thermostat, and the angles which the servo should move to in order to reach these extreme temperatures; a linear relationship between servo angle and temperature is then assumed.
- c. *Bolt lock:* You need to set the angle which the servo should move to when opening the lock (in the range 0-90°) and the angle it should move to when closing it (in the range 90-180°).

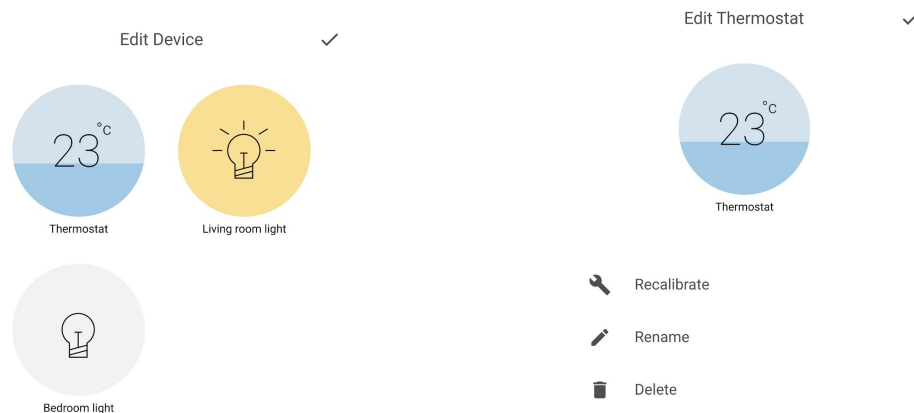
CONTINUE

CANCEL SETUP

Editing an existing robot

Sometimes it may be necessary to recalibrate a robot which is already registered to your account. For example, you might want to attach a different PAL to a HAL you've already got set up, or perhaps your thermostat always ends up being just one degree off. You may also want to give a robot a new nickname if you find that Alexa struggles to recognise your robot's current name. In the extreme case, you may wish to delete a robot from your account all together.

Luckily, such editing is very easy: simply navigate to *Settings* in the app and select the robot you wish to edit. Choosing *Recalibrate* will start the same Robot Registration Wizard as you used to register your robot in the first place, but will skip straight to step number 2 so that you do not need to enter the HAL's details again. This way, you can change the attachment registered to the HAL or simply adjust the parameters.

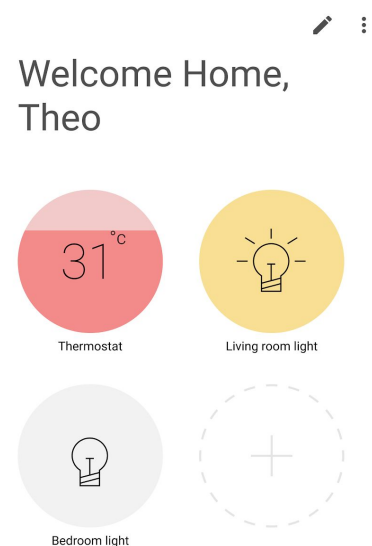


Using your Smart Home Adapters

Interacting with your robots is super easy, and is done directly from the app's main screen:

- **Switch PAL & Bolt lock PAL:** Simply click on the desired robot in the UI to change its state between on/off or open/closed.
- **Thermostat PAL:** To increase or decrease the temperature, click and drag on the robot in the UI.

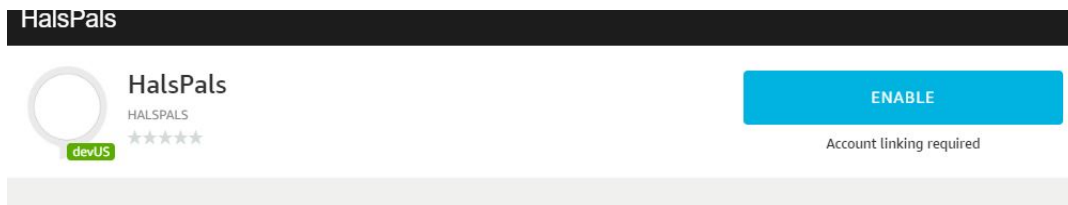
In either case the desired command is then sent over to the web server, which will translate the command and direct appropriate instructions to the robot wirelessly.



Alexa integration set-up

1. Navigate to the *Skills & Games* if you are using the Alexa mobile app, or the *Skills* tab if you're using the Alexa web app.

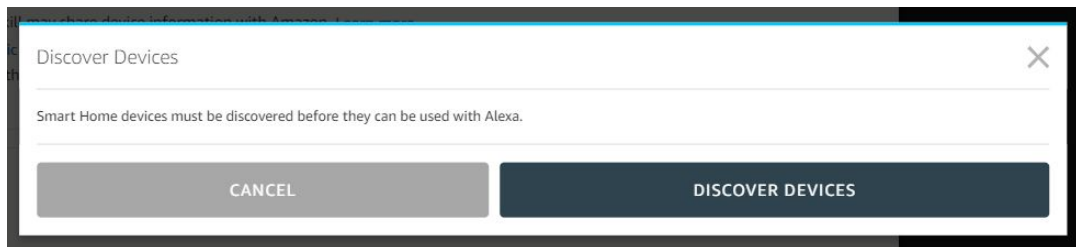
2. In the skills tab, search for *HalsPals*. Enable this skill.



3. You will now be taken to the HalsPals login page; sign in using your credentials, and if prompted consent to our use of your login details.

HalsPals

4. Once signed in, the *Discover Devices* tab should open automatically in your Alexa console. Click *Discover Device*; this will automatically add all the robots which you have registered to your account to your Alexa.



5. Done!

Using Alexa

Once set up, using Alexa to control your devices is easy. Alexa can recognise the friendly name you set for the adapters and you can therefore control them using short, descriptive sentences.

Here are some example commands:

- "Alexa, discover the device"

- “Alexa, turn on the kitchen switch”
- “Alexa, turn off the kitchen switch”
- “Alexa, set the smart thermostat to 25 degrees”
- “Alexa, open the front door”
- “Alexa, lock the front door”

Troubleshooting Guide

Problem	Solution
My phone doesn't allow me to install the APK	Enable the installation of untrusted apps. The exact steps involved will depend on your specific version of the Android OS; refer to your phone's user manual if you're unsure
My Switch / Thermostat / Bolt lock PAL is inaccurate or fails to carry out its task	Recalibrate the PAL. This can be done by selecting the desired PAL in the app interface, pressing <i>Recalibrate</i> , and then following the on-screen instructions
My Switch / Bolt lock PAL doesn't stick to the wall	Clean the surface that you are applying the PAL to and apply another sticky pad
My Thermostat PAL doesn't stick to the wall	Adjust the racks and tighten the screws
The app won't start	Update to Android Marshmallow or later and try again
The app crashes	See above
I can't login to my account	Check your credentials and try again
I get a network error when trying to control my robots	Make sure you are connected to the internet so that your commands can reach the server
The QR code doesn't scan	Click the enter the robot ID manually button and input the ID that is below the QR code
My phone doesn't have a camera	See above
The app says the robot is disconnected	Move the device closer to your WiFi router and try again
The server accepts the command but nothing happens	Charge the HAL which isn't working

Alexa cannot find the device	Navigate to the device page in your Alexa console click the <i>Discovery</i> button . If it is still not working, try to disable the skill and enable it again
Alexa cannot recognise the device you want to control	It is important to choose a simple and easy name for your device so that Alexa can recognise it easily

Appendix - Advanced Installation Guides

Android app installation (source)

Note: this guide is for advanced users. If you are not planning on modifying the software, chances are you'll have an easier time following the steps in *Android app installation (APK)* instead.

Prerequisites

- Android OS version Marshmallow or later (this may also be called *version 6.0* or *API level 23*).
- USB Debugging enabled on the target phone.
- Android Studio version 3.0 or later installed on the computer you wish to build the app on.

Installation

1. Download the latest stable version of the source code from the **master branch** of <https://github.com/mrbenshef/SmartHomeAdapters/>. If you're not familiar with Git, the easiest way of doing this is by clicking *Clone or download* > *Download ZIP* and then unzipping the downloaded file.
2. Open the source code's *android* subdirectory in Android Studio.
3. Wait for the Gradle sync process to finish; this usually takes a few minutes. In the meantime:
 - a. Plug your phone into the computer. You may be asked to confirm that you trust the computer; you must do this in order to be able to install the app.
4. Build and run the app on your phone by pressing *Run* > *Run* and selecting your target phone.
 - a. If you can't see your phone in the target list, make sure your phone is properly connected to the computer and that you have successfully enabled USB debugging.
5. Android Studio will now download and install all the dependencies for the project before building and installing the app onto your phone; this may take a couple of minutes!

If the app now automatically starts on your phone you have successfully built the app from source, congratulations! You can now safely close the app and use your phone for other purposes; the app will remain installed on your phone for future use.

Microservice build and deployment

Follow this guide if you want to run a local instance of our microservices (instead of the hosted services at halspals.co.uk).

Prerequisites

- Knowledge of docker, docker-compose and nginx
- A domain name
- DNS configured to point to the server you will be deploying on
- A Linux server (for this guide we will assume Ubuntu)

Installation

1. Follow the instructions at <https://docs.docker.com/install/linux/docker-ce/ubuntu/> to install Docker and docker-compose
2. Install dependencies: `sudo apt install nginx certbot`
3. Configure HTTPS
 - a. Run certbot: `sudo certbot --nginx certonly -d YOUR_DOMAIN_NAME`
 - b. Follow the prompts to register the domain
4. Run the bash script located at TODO, giving it your domain name as an argument. It will automatically set up an nginx configuration.
5. Build the docker images
 - a. Clone the repository: `git clone https://github.com/mrbenshef/SmartHomeAdapters.git`
 - b. Enter the directory: `cd SmartHomeAdapters`
 - c. Build the images: `docker-compose build`
 - d. Copy the docker-compose file to somewhere safe: `cp docker-compose.yml SAFE_LOCATION`
 - e. Run a bash script to set up your nginx config: `cd scripts && chmod +x nginx_config.sh && ./nginx_config.sh`
 - f. Remove the source code: `cd ../../ && rm -rf SmartHomeAdapters`
6. Restart nginx: `sudo systemctl restart nginx`
7. Go to the docker-compose file and start up all the containers: `cd SAFE_LOCATION && docker-compose up -d`
8. Create the OAuth clients
 - a. Create android OAuth client: `docker-compose exec hydra clients create --id 'YOUR_CLIENT_ID' --scope "openid" --response-types code --grant-types authorization_code`

- ```
--callbacks https://callback.DOMAIN
--token-endpoint-auth-method none
```
- b. (you have to create the alexa skill first to collect callback URL) Create the alexa OAuth client: `docker-compose exec hydra clients create --id 'alexa' --scope "openid" --response-types code --grant-types authorization_code --callbacks URL1,URL2,URL3 --token-endpoint-auth-method none`
  - c. Follow the alexa/android build instructions to add the OAuth information

## Alexa lambda function deployment

### Prerequisites

- An Amazon Developer account. This is required to create and configure Alexa skills.
- An Amazon Web Services (AWS) account. This is required to host the skill logic in an AWS Lambda function.

You also need Git and Python to work with lambda. And you have to follow the previous step to set up the server first.

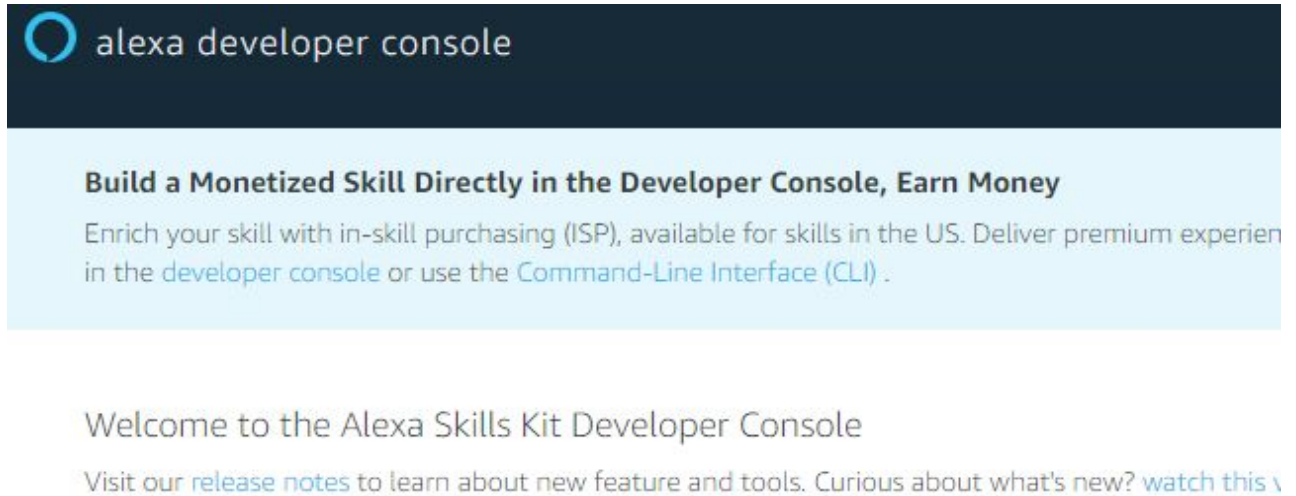
### Installation

#### Get Source Code

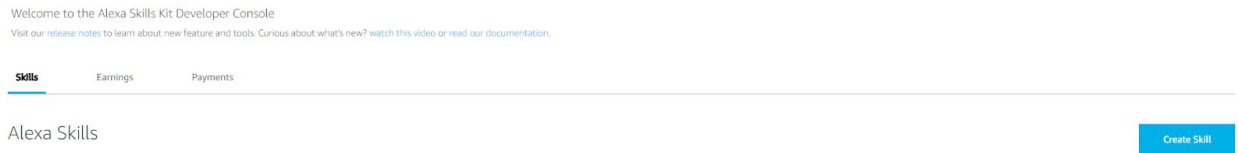
Get the Source Code from our GitHub repo `lambda` folder.

#### Create a Skill via the Alexa Skills Kit Developer Console

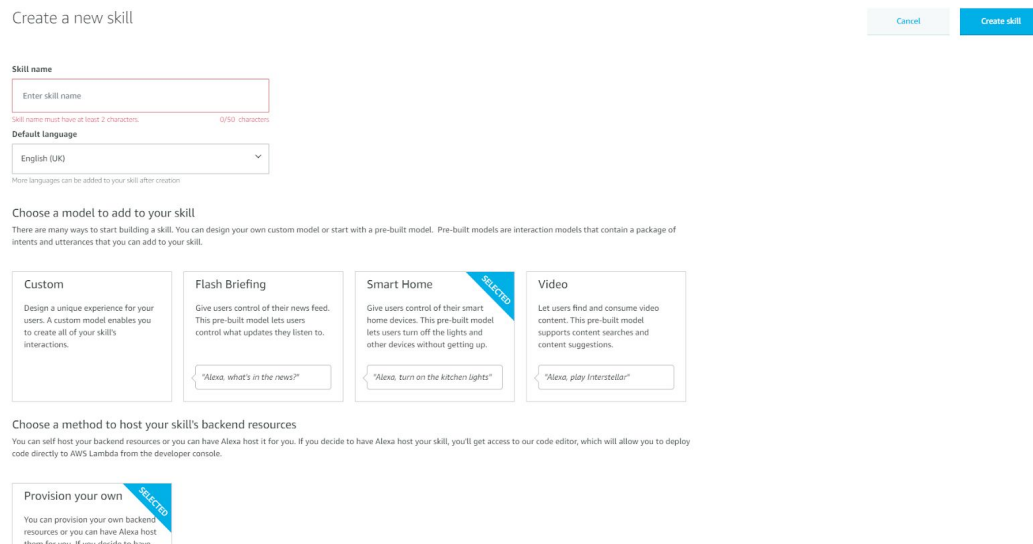
1. In a web browser to the Alexa Skills Kit Developer Console at <https://developer.amazon.com/alexa/console/ask>. If not already authenticated, you may have to Sign In with your Amazon Developer Account.



2. Click the Create Skill button.



3. For the Skill name, enter your-skill-name.
4. Leave the Default language as English (UK).
5. Under Choose a model to add to your skill select the Smart Home model.



6. Click the Create a skill button. When completed, you should be at the configuration page for a newly created your-skill-name page.



Be careful here, you have to remember your skill ID for later Lambda function set up.

### Create the Lambda function

1. Navigate to “/sample\_lambda/python”
2. Zip all contents of that directory to python.zip. NOTE: Take care to ensure that the files from /sample\_lambda/python appear in the root of your .zip file, not in a nested subfolder.

Now let's set up the Lambda function:

1. Go to <https://console.aws.amazon.com/console/home> and sign in
2. Go to Services > Compute > Lambda
3. Click on Create Function
4. Step 1: Click on Author from scratch
5. Step 2: Configure your Lambda function
  - a. Name = SampleLambdaFunction (or whatever you want)
  - b. Role = Create a Custom Role which will launch a new tab. Click Allow to create a new role named lambda\_basic\_execution and automatically insert this role into the Lambda basic information dialogue.
  - c. Click Create Function
6. Step 3: Click Triggers -> Add Trigger and select Alexa Smart Home
  - a. Application Id = skill ID of your test skill that you noted above
  - b. Enable trigger = checked
  - c. Click Submit
7. Step 4: Click Configuration
  - a. Runtime = Python 3.6
  - b. Code entry type = Upload a .ZIP file
  - c. Click on Upload and find the python.zip you created earlier
  - d. Handler = lambda.lambda\_handler
  - e. Click Next, then click Save
  - f. On the top right corner, note the Lambda ARN

### Configure Skill

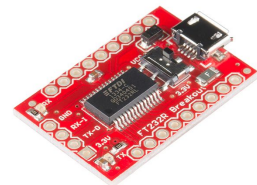
1. Go back to <https://developer.amazon.com/home.html> and sign in as needed
2. Go to Alexa > Alexa Skills Kit > the test skill you created earlier
3. In the Configuration tab:
  - Lambda ARN default = enter your Lambda ARN noted from the previous step
  - Authorization URI = <http://oauth.halspals.co.uk/oauth2/auth>
  - Client ID = alexa
  - Scope: openid
  - Access Token URI: <http://oauth.halspals.co.uk/oauth2/token>
  - Client Secret: After finish the Microservice build and deployment 8.b, you will have the secret, copy that secret to here.
  - Client Authentication Scheme: HTTP Basic (Recommended)

- Click Save
- 4. Provide redirect URLs to your OAuth 2.0 server:
  - There are three redirect URLs at the bottom, copy these URLs to your OAuth client while you are doing Microservice build and deployment 8.b, replace them with URL1,URL2,URL3

## Flashing custom WiFi credentials unto the HAL

### Prerequisites

- A compatible USB-to-Serial board for reflashing the ESP wifi module. We suggest the *SparkFun USB to Serial Breakout - FT232RL*.
- USB to Micro-USB capable of passing serial information.
- Jumper cables.
- A computer with *Arduino Studio* installed.
- The HAL source code (found on our GitHub repository).



### Flashing

1. Connect the RX pin of the USB-to-Serial board to the TX pin on the PCB via jumper cables, and also the TX pin of the USB-to-Serial board to the RX pin on the PCB.
2. Connect the USB-to-Serial board to your computer.
3. Open the Arduino IDE and import our HAL source code.
4. Add your WiFi credentials to the source code and flash it onto the PCB.
5. Once complete open up the serial monitor to check for a successful connection as shown below.

