

# Welcome

Welcome to openHAB. We hope that your journey with openHAB is a rewarding one. If you encounter problems or simply cannot figure something out on your own, please do not hesitate to post a question on [the forum \(opens new window\)](#). There are many fellow openHAB users ready to help out, and polite constructive feedback can help us to improve this documentation for the next new user. Most requests for help usually receive a response within a couple of hours, but please use the search on the forum to check if the topic has already been discussed.

## Why openHAB

Smart devices are appliances, such as TVs, switches, plugs, lamps, washing machines, etc. that, besides having control buttons (or remote controls), can also connect to a network, allowing app control.

Unfortunately every manufacturer has its own app making it difficult to coordinate actions among devices built by other manufacturer. If you start with a certain ecosystem, such as Ikea, you can take actions on one device based on status changes of another device, but sooner or later you will need to acquire smart devices not manufactured by Ikea and not manageable with its app.

Some of these ecosystems, such as IFTTT, Alexa, Xiaomi, TP-Link, Sonoff and others, require cloud access to work. In other words, You can't control your devices if your internet connection fails. Furthermore, some manufacturers (and their clouds) are hosted in Countries whose governments are known to spy their citizens. Do you want to end-up in that group ?

With openHAB you have your own computer, whose programming is controlled by you, to manage all your devices. In cases where you need to manage devices remotely you can use VPNs, or other secure means such as the \_\_\_ cloud to do so.

OpenHAB is developed by a group of people worried with security and ease of use. If you opted for openHAB you can be assured of having an ultimate home automation experience.

## Connecting devices

Home automation needs fast reaction. If you push a button you want the light on or off in a matter of milliseconds (seconds are not acceptable). It's not important to have a fast connection because IoT (internet of Things) devices use very short messages, what is really important is to connect IoT devices to non-crowded and without interferences connections.

Another key aspect is security. You can install firewalls in computers but not in IoT devices. IoT devices are very vulnerable to malware infections and are often used during hacker attacks, so it's strongly recommended to connect IoT devices to dedicated networks whose internet connection is protected by a firewall. It's not the purpose of this paper to discuss how this can be achieved.

Smart devices can connect to openHAB through wires or over the air (OTA). Please select smart devices that use open protocols, otherwise you risk to be tied with their manufacturer forever.

Wires are a robust way to interconnect devices. They grant performance and minimize interference with the outside world. They are also more expensive, and for existing houses their installation cost

may be prohibitive. The main open standards supported by openHAB for wired connections are ethernet and KNX.

Regarding OTA connections the main ones supported by openHAB are:

- Wifi – was not created for home automation but a lot of IoT devices use it. Such devices need more power than any other non-wired technology, so they are either continuously connected to mains or recharged frequently. They also contend with radio channels of non-IoT devices and can suffer radio interference from microwaves or other devices, from you or from your neighbours
- Zigbee – it was created with IoT in mind. So it's a way to isolate IoT devices from wifi ones. Batteries for zigbee devices can last for 1-2 years due to the low electrical requirements of its electronics. But they contend with wifi for the same radio spectrum, so can suffer from similar interferences. There are ways to minimize (but not avoid) this
- Z-Wave – like zigbee its electronics require very low power. It also operates on a lower radio frequency so the interference risk is smaller. Also its radio coverage is better (the lower the frequency the higher the coverage)
- Bluetooth – some specialized IoT devices use it

Computers where you can install openHAB usually come with ethernet and/or wifi adapters. To use other connection methods you need to connect to this computer either a USB dongle or a gateway device.

My preference goes to gateway devices. USB dongles often have compatibility issues and their radio coverage is often poorer. Gateways can usually be installed anywhere while USB dongles cannot be too far away from the openHAB computer.

## openHAB device support

openHAB supports a lot of devices, USB dongles and gateways, but often this support is not embedded in its base code. So, after installing openHAB ([link to install guide](#)), you will be surprised that openHAB does not find any of your IoT devices. You will need to install another piece of code: a binding.

You can see the binding catalog [here](#). Please search with your equipment's model to identify which bindings you need. For example, for the TP-Link HS100 smart plug you need the TP-Link binding, but for the TP-Link P100 smart plug you need the Tapo binding. So it's not always evident which bindings you need to install. So first search the catalog and then install the bindings you need. Installation is very simple, see \_\_\_\_\_. Also read the binding documentation, certain bindings, such as MQTT, OpenWeatherMap and Unifi, require initial parametrization before

Several bindings perform auto-discovery. So, after installing such bindings several of your IoT devices will be discovered. Others bindings only support manual discovery, so you have to trigger a scan in order to discover the remaining devices.

Input queue ...

After the discovery phase you will notice that openHAB created several things. You can learn more about things [here](#), but generally speaking things define how openHAB contacts devices. Things can

have several channels, for example a window contact sensor has one channel that reports if the window is closed or open, and may have additional channels to report battery status and other info. Therefore one thing has at least one channel. It can have dozens, such as things created by the Open Weather binding, where each channel contains weather forecast (humidity, pressure, nebulosity, rain, etc.) for different times of the day.

This may sound a little bit complex, but you are not obliged to use every available channel.

## MQTT

If you have devices that connect through MQTT please ignore them for now, unless you know what MQTT is and know how to use it.

Usually devices communicate with each other like a voice conversation. One speaks, the other listens, then replies, and so on. This is called a point to point communication. Nobody else listens (or should listen).

MQTT does not work like this. In the MQTT world there is a server (broker in MQTT terminology) that listens to messages published by devices and apps, and redistributes these messages to who is interested in them.

It's like a radio broadcast. You don't know who listens to it. Could be one, a million, or nobody. Each sent message has a "topic" (MQTT terminology). Devices, apps (such as openHAB, MQTT.fx and others) subscribe to these topics so that the MQTT broker knows who is interested in each topic. When a message is sent by a device or app, the broker receives it and redistributes it according to the subscriptions.

Implementing MQTT is not trivial, it requires that you install **mosquitto** (or another MQTT broker tool). For more info you may read this thread in the forum or the MQTT documentation.

## How to use devices with openHAB

In openHAB all devices are represented by things and their channels. As we have already seen, bindings assure the message flow between channels and devices so that you can read values and/or control their statuses.

There are however other basic, and very important, openHAB concepts:

- Items – they are tied (linked in openHAB terminology) to channels. Why can't we act directly on channels and need items in between ? One reason is that the info sent by devices may need some transformation to become readable. Another important reason is that you can group items to ease management tasks (we will come back to this later). You don't need to touch things and channels when you modify item groupings or transformations. Therefore you only touch things and channels when you install new devices, afterwards, if you move a device from a kitchen to a bedroom you only change associated items. This may seem a little complex in the beginning, but as time goes on you will see benefits from it
- Transformations – as said in the previous point, it is sometimes necessary to transform info sent to (or received from) devices. openHAB provides some tools to make this task easier

As said above, once devices are discovered openHAB can create things and channels automatically. But you have to create all items manually. Ideally you should use a so called “semantic model”. This allows the semi-automatic creation of fabulous management panels, such as the ones shown in this demo system.

You are not obliged to create the semantic model from the beginning. My own experience is that a semantic model evolves over time. You can create something today and modify it weeks later.

But as the minimum you should include each item in at least two groups:

- Locations – kitchen, living room, bedroom, bathroom, garden, corridors, etc. Locations can be grouped but you can create the hierarchy later on
- Equipment types – HVAC, illumination, plugs, switches, etc

openHAB comes with predefined classes for each group, just pick the one you think is closer to what you want. You can also define other groupings, such as temperatures, but I suggest to tackle this later.

## Rules

Why do you want a home automation system ? It’s not only to replace existing buttons and remote commands, it’s fundamentally to have an “intelligent house”. openHAB uses rules for this. Here are some examples of what you can do with them:

1. Adjust heating targets according to external temperature and other factors (such as being on holidays or not)
2. Perform vacuum cleaning when not at home
3. Automatically raise electric chairs during vacuum cleaning
4. Program your garden’s irrigation system according to rain, humidity, soil moisture, wind and other conditions
5. Delay dish and/or laundry washing to when your solar panel is providing enough energy
6. Control devices with your own voice

To develop a rule there are two things you need to decide:

- When should the rule be executed – this is called the trigger. It can be one event, such as a time or status change, several events, a voice command, etc
- What to do -

The forum contains miriads of examples and ideas, use them, and share your own experience.