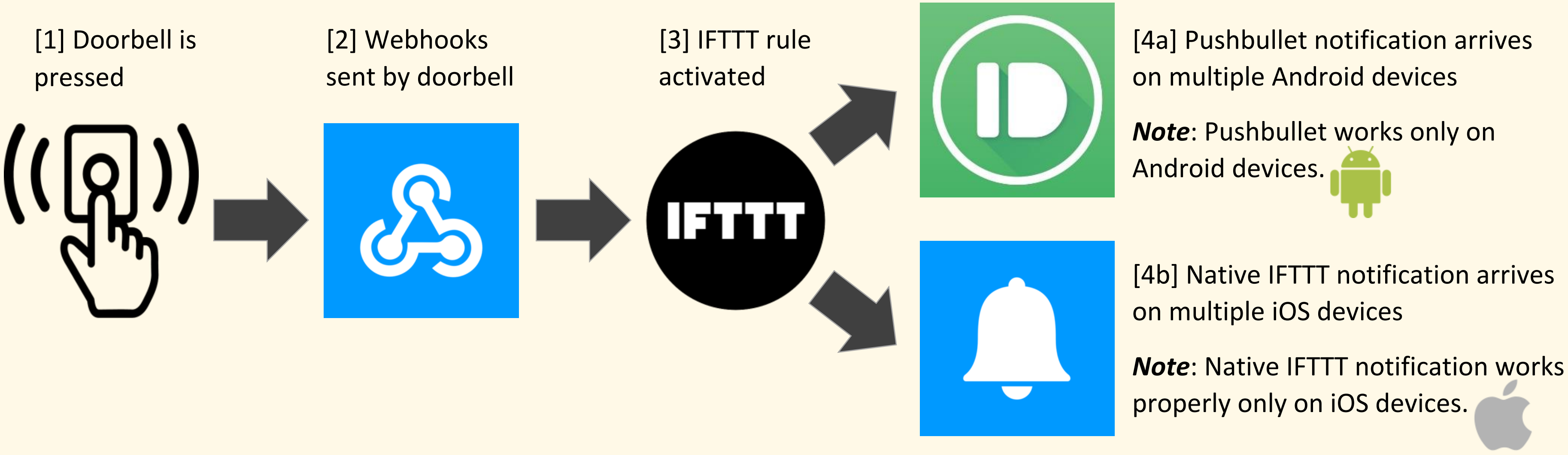
Guide to Set Up Doorbell Based on IFTTT

This Doorbell may be used to send push notifications to multiple devices. Operating principle as illustrated:



[1] Doorbell is pressed

[2] Webhooks sent by doorbell

[3] IFTTT rule activated

[4a] Pushbullet notification arrives on multiple Android devices

Note: Pushbullet works only on Android devices.

[4b] Native IFTTT notification arrives on multiple iOS devices

Note: Native IFTTT notification works properly only on iOS devices.

IFTTT & Pushbullet Account Creation and Signing In

Step 1
Create an IFTTT account, or alternatively use Apple, Google or Facebook sign in.

Step 2a **[iOS users, skip this]**
On all the Android devices for which you wish to receive the push notifications, install the Pushbullet app and sign into it with Google.

Step 2b **[iOS users only]**
On all the iOS devices for which you wish to receive the push notifications, install the IFTTT app and sign into it.

Set Up the IFTTT Rule

Step 1
In your browser, go to https://ifttt.com/explore

Step 2
Click "Create"

Step 3
Click "If This" ➲ search for "Webhooks" ➲ click "Receive a web request". Under “Event Name”, input "Doorbell_Pressed" ➲ click “Create trigger”.

Step 4a **[iOS users, skip this]**
Click "Then That" ➲ search for "Pushbullet" ➲ click “Push a note”.

If this is your first time attempting to integrate IFTTT with Pushbullet, then click “Connect” ➲ click “Approve”, otherwise skip on.

Step 4b **[iOS users only]**
Click "Then That" ➲ search for "Notifications" ➲ click “Send a notification from the IFTTT app”.

Step 5
Under "Message", type in the notification message, e.g. “Someone is at the door!” ➲ click “Create action”

Step 6
Click “Continue” ➲ click “Finish

Test the IFTTT Rule and Save the Key Offline

Step 1
In your browser, go to https://ifttt.com/explore

Step 2
Search for "Webhooks" ➲ click “Documentation”

Step 3
Under “To trigger an Event with an arbitrary JSON payload”, replace “{event}” with “Doorbell_Pressed” ➲ click “Test It”

Step 4
If you do receive the notification on your mobile devices, then the rule has been successfully set up.

Now, copy and save your webhooks key somewhere in your mobile device so that you may access it offline later.

Important Information
Your webhooks key is the string that appears immediately after “Your key is:” This will later be entered and stored in the Doorbell as the variable *webhooks_key*.

Similarly, the event name “Doorbell_Pressed” will later be entered and stored as the variable *maker_event*.

Enable Pushbullet Notifications even when Locked on Android Devices

Why this is Necessary

Android devices tend to sleep on Pushbullet notifications, especially when the device has been locked and idle for some time.

Step 1 – Unrestrict Battery Usage

Settings ➲ Apps ➲ Pushbullet ➲ Battery ➲ Set to "Unrestricted"

Step 2 - Enable pop-up notification

Settings ➲ Apps ➲ Pushbullet ➲ Notifications ➲ Notifications categories ➲ Tap on "Follower" ➲ Set to "Alert", then set enable "Show as pop-up" and also set to "Vibrate"

Step 3 - Customise Ring Tone

Do all the steps in *Step 2* above, except instead of "Alert" at the last step, ➲ Sound ➲ Select the sound you like

Step 4 - Add to Ignore Do Not Disturb

Settings ➲ Notifications ➲ Do not disturb ➲ App notifications ➲ Add the Pushbullet app

Guide to Store WiFi & Webhooks Credentials

The Doorbell boots up in Access Point (AP) mode by default if there are no user credentials saved in its flash memory. In this mode, the status LED of the Doorbell is unlit, and the user may connect to the AP via WiFi to access a web-based configuration page. The following user credentials may then be entered, and the values will survive successive reboots of the Doorbell:

1) WiFi SSID
2) WiFi password
3) A 64-character array which is stored as *webhooks_key*
4) A 32-character array which is stored as *maker_event*

If in doubt about (3) & (4), look up ***Important Information*** for tips.

While the Doorbell has been programmed to send webhooks to IFTTT using the saved variable *webhooks_key* as the key and the saved variable *maker_event* as the name of the event, a coder may re-purpose these two variables for any other usage.

Once the Doorbell successfully connects to a WiFi hotspot, it exits AP mode and operates in Station (STN) mode for all subsequent reboots unless it fails to connect to the WiFi hotspot at boot; in which case it enters AP mode for 180 seconds before returning to STN mode.

In the STN mode, the status LED of the Doorbell flashes slowly. It listens for a trigger of the Doorbell, and lights up solid for 5.5 seconds when that happens.

Entering User Credentials for the First Time

Before You Begin

As you will lose internet access during this process, consider saving your user credentials offline before you begin so that you may copy and paste them as strings into the input fields later.

Step 1

Power up the Doorbell. It will enter AP mode.

Step 2

Use your device to connect to the WiFi SSID “AutoConnectAP”. If your device refuses to connect to it, try to hit the “Forget” option and re-attempt.

Step 3

Once connected, your browser may launch a set-up page automatically. If it does not, tap on the WiFi icon or pop-up prompt which states “*Sign in to the network.*”, or manually launch the page “192.168.4.1” in your browser.

Step 4

Click on “Configure WiFi”, and enter all user credentials in the next page that loads. Click “Save”. The Doorbell will be reset and it will enter STN mode if it successfully connects to the WiFi hotspot.

Amending User Credentials

There are two methods to put the Doorbell back in AP mode.

Method 1: Reflash the entire NodeMCU

Step 1
Launch Arduino IDE

Step 2
Open the Doorbell.ino code provided along with your order.

Step 3
Select Tools ➲ Erase Flash ➲ All Flash Contents

Step 4
Select Sketch ➲ Upload. Once completed, the Doorbell will be reset and it will enter AP mode without any previously saved user credentials.

Method 2: Turn off WiFi Hotspot

Step 1
Switch off your WiFi hotspot

Step 2
Power cycle the Doorbell once (this means disconnect power, and then reconnect power). This will put the Doorbell in AP mode for 180 seconds but does not erase the previously saved user credentials.

You will have a three-minute window to complete Steps 3, 4 & 5, otherwise the Doorbell will return to STN mode and you will have to redo Step 2.

Step 3
Use your mobile or desktop device to connect to the WiFi hotspot with the SSID “AutoConnectAP”. If your device refuses to connect to this SSID, try to hit the “Forget” option and re-attempt.

Step 4
Once connected, your browser may launch a set-up page automatically. If it does not, tap on the WiFi icon or pop-up prompt which states “*Sign in to the network.*”, or manually launch the page “192.168.4.1” in your browser.

Step 5
To erase WiFi credentials, select Info ➲ Erase WiFi Config. Once completed, the Doorbell will be reset and it will enter AP mode without any previously saved user credentials.

Step 6
Switch your WiFi hotspot back on.

Step 7
Re-connect your device to the “AutoConnectAP” and re-load the page “192.168.4.1”. If your browser shows the previous WiFi credentials upon reloading the page, clear your browser cache and reload the page.

Step 8
Click on “Configure WiFi”, and enter all user credentials in the next page that loads. Click “Save”. The Doorbell will be reset and it will enter STN mode if it successfully connects to the WiFi hotspot.