Toilet Paper Monitoring System for Public Restrooms

Today we are going to build a toilet paper monitoring system for public restrooms. The whole system consists of several toilet paper holders and a web system. The web system displays the remaining toilet paper amount of every holder connects to it. When the remaining amount of a toilet paper roll is running low, a browser notification will pop up, and a message reminding you to refill that roll will be sent to a Slack channel.

In the following tutorial, we list the tools and materials, including source code, needed for building one toilet paper holder and the web system, and the steps for you to follow to make one yourself.

What You Need -- Tools and Materials

Tools

1. 3D printer
2. Laser cutter
3. Knife
4. 502 Super Glue

Software

1. Autodesk Fusion 360 (or any 3D modeling software)
2. Adobe Illustrator (or any design software that supports drawing with vector)
3. Visual Studio Code (or any text editor)

Online Services

1. IFTTT (Webhook & Slack API) (https://ifttt.com)
2. Particle web IDE (https://build.particle.io/)

Materials

1. Particle Photon
2. Mini breadboard
3. Adafruit Round Force-Sensitive Resistor (FSR) - Interlink (Introduction and tutorial: https://learn.adafruit.com/force-sensitive-resistor-fsr?view=all)
4. 10k resistor

5. Several wires
6. 5V 1A power bank
7. Cardboard (45cm * 45cm, or any size that fits your laser cutter platform)

Code

1. Photon code
   https://github.com/Reedo0910/ToiletPaperMonitor/blob/master/photon-code/ForceMonitor.ino
2. Monitor web interface code
   https://github.com/Reedo0910/ToiletPaperMonitor

Steps

In the following tutorial, we are going to make the rod of the toilet paper holder first, then the shell. After assembling the rod and the shell, we are going to work on the Photon, including constructing the circuit and the code. At last, we will use the code provided to host a monitor website that connects to our Photon.

Step 1: Modeling the rod of toilet paper holder

The rod of our toilet paper holder consists of 3 parts, here we call them A, B, and C (as shown in Figure 1-1). We are going to use 3D printing to produce these parts, so the first step would be building the model of these 3 components.

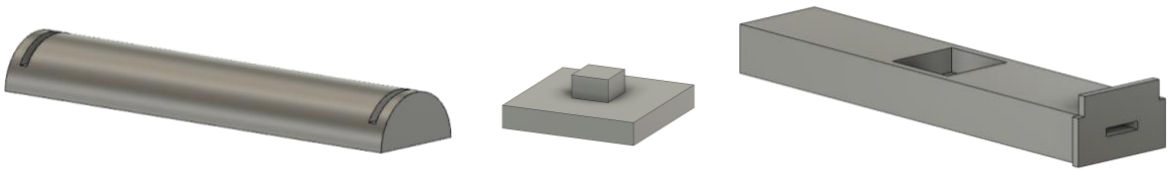

Figure 1-1. The 3 parts of the rod. From left to right is A, B, C, respectively.

1-1 Build 3D model for A, B and C

The first step is to decide the sizes of these 3 components. The width, length, and height of the components depends on the size of your toilet paper roll, so the dimension of your models will probably be different from ours.

The model of our rod can be found in the 3d-printing-model folder of our GitHub repository (https://github.com/Reedo0910/ToiletPaperMonitor/tree/master/3d-printing-model), but we strongly suggest you to alter it to fit your toilet paper roll.

The length of A depends on the size of your toilet paper roll -- it must be longer than the roll, but not too much. The width of the toilet paper roll we use is 9.7cm, so we make our A 11cm in length. The height of A should be less than half of the shaft's diameter of the toilet paper roll. Here we make our A 12mm in height.

C is the bottom part of the rod. There is a hole in the middle and a tunnel to one side of it. The hole is for putting the Adafruit Round Force-Sensitive Resistor (FSR); and the tunnel is for the wires connecting the FSR and our Photon.

B is designed to be glued to the bottom of A and will contact the FSR on its smaller side. The small side should be smaller than the FSR's area to make sure the weight of the whole toilet paper roll and A could be on the FSR and not shared by other parts of C.

1-2 Glue A and B

Use glue to stick B to the bottom of A (as shown in Figure 1-2). Make sure it's position fits C when you cap A+B on C (Figure 1-3).

Figure 1-2. B is glued to the bottom of A.

Figure 1-3. Assemble A, B, and C, and make sure B's position fits the hole on C.

The reason why we make A and B two separate components and then glue them up is that it would be hard to use a 3D printer to produce it if we make it in one model.

1-3 Put the FSR into C

First, we have to solder wires to the tabs of the FSR carefully. Remember to do it fast because this FSR is somehow fragile and you might melt the plastic and destroy the FSR if you do it slowly.

Then, put your FSR into the hole of C and let the 2 wires go through the tunnel (Figure 1-4).

Figure 1-4. The FSR is put into the hole of C and the 2 wires go through the tunnel.

Step 2: Designing and building the shell of toilet paper holder

We are using cardboard to create the shell. We will start with drawing every plane in Adobe Illustrator, and then use a laser cutter to get every plane cut. At last, we will assemble the planes and use some glue to strengthen it.

2-1 Design the planes

What is required in the shell are a place to hold the rod and a place to put our Photon and power supply, so we designed the shell to be composed with 6 planes -- there is a plane in the middle for the Photon and power supply, and there the bottom is not sealed to let the paper out (Figure 2-1).

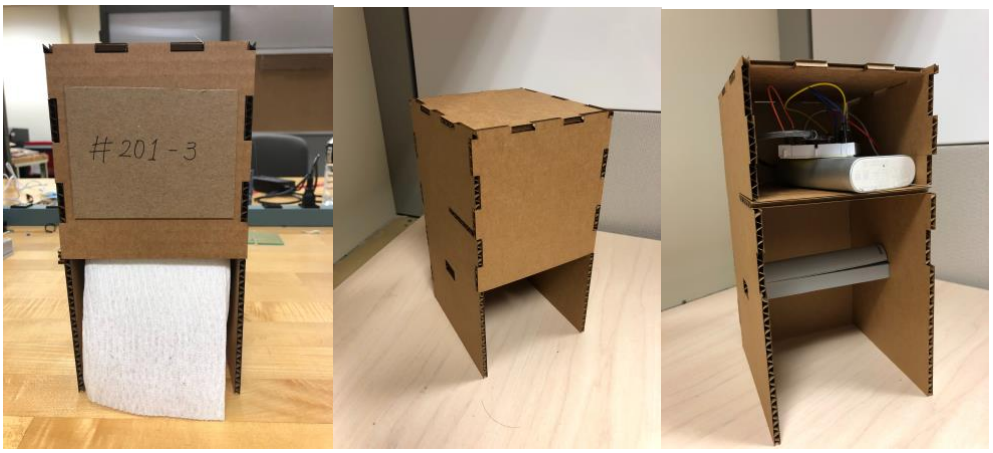

Figure 2-1. Photos of the shell of toilet paper holder.

The planes are shown in Figure 2-2. Since we are going to use a laser cutter to cut them, we set every stroke we want to cut to **red** and the thickness to **0.001pt**. Remember to check the manual of your laser cutter in case the setting is different.

Be careful that the strokes in Figure 2-2 are much thicker than they ought to be for the sake of making it easier to see.

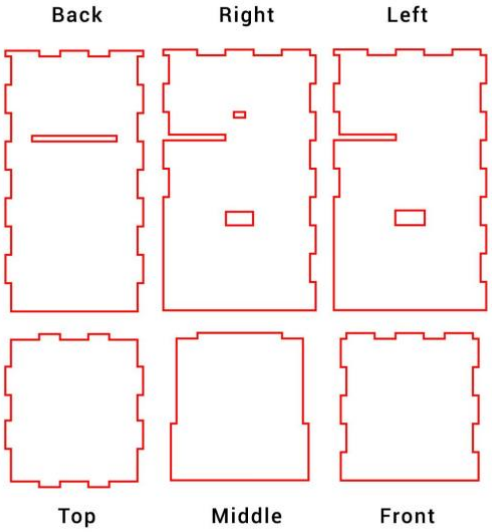

Figure 2-2. The planes of toilet paper holder's shell.

Our Adobe Illustrator file containing the planes is in the laser-cutting-file folder of our GitHub repository (https://github.com/Reedo0910/ToiletPaperMonitor/tree/master/laser-cutting-file), you can edit it to fit the size of your toilet paper roll.

2-2 Laser cut the planes

Now, let's export our design to an SVG file and send it to the laser cutting system. Then, put a cardboard into the laser cutter (Figure 2-3). The maximum adaptable size of the cardboard varies according to the platform of the laser cutter you use.

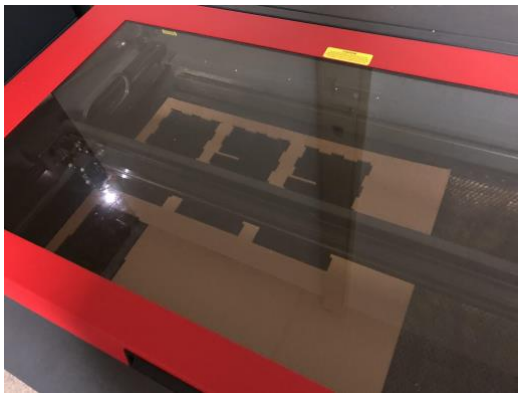

Figure 2-3. A cardboard in a laser cutter.

If some strokes of your plane are not cut through completely, you can adjust the output power of the laser cutter and cut again, or use a knife to do it manually.

2-3 Assemble the shell

Now, let's put up the planes together to form the shell. You can use some glue to help you fix the planes and stabilize the structure.

Make sure that you **DO NOT** glue the **left** side board since it is supposed to be the place for us to remove an empty toilet paper roll, refill a new one, and access our circuit and power bank in case there is anything wrong with the circuit or the battery is dead.

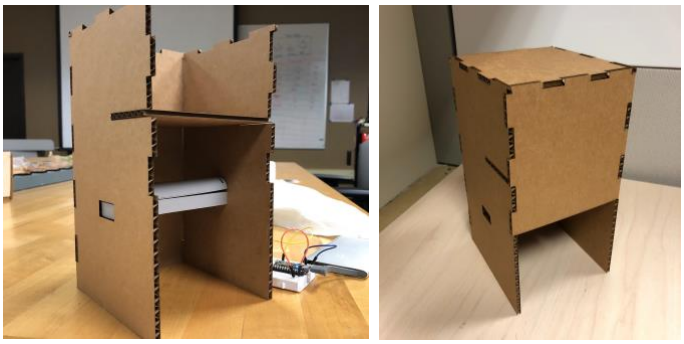

Figure 2-4. Planes are assembled to form a shell.

Step 3: Building circuit and getting FSR reading from Photon

Now, we have completed building the toilet paper holder. It is time to move on to the circuit and code!

3-1 Connect FSR to Photon

Connect one wire of your FSR to the ground, and another to 3V3 with a 10k pulldown resistor (Figure 3-1) on a mini breadboard. Since FSRs are non-polarized, you can connect either wire to one side.

Figure 3-1. The circuit of FSR connecting to Photon. (Source: https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr)

3-2 Get source code for Photon

Now we have the circuit built. The next step is to get some code! We need to get reading from the FSR and make it accessible to our monitor system. Go to Particle web IDE and create a new app. After that, go to the photon-code folder in our GitHub repository (https://github.com/Reedo0910/ToiletPaperMonitor/tree/master/photon-code) and copy all the code in the `ForceMonitor.ino` file. Then, paste it into the app you just created in the Particle web IDE (Figure 3-2).

Figure 3-2. The interface of Particle web IDE with ForceMonitor.ino's code.

Next, click Flash on the top of the left sidebar to flash the code into your Photon.

3-3 Get reading from FSR

Since we have this line of code `Particle.variable("force_level", &fvalue, INT);` working in our Photon, the reading of our FSR is accessible through the Particle service. You can check the value on Particle Console (https://console.particle.io/). Go to particle Console, click My Devices on the left side bar, then click the Photon you used in this project to access the page displaying information about your Photon. On the bottom of the right column of the information page, there is a Variables section. There should be a variable called `force_level`. You can get a reading from the FSR if you click the Get button. If there is no variable listed in the section, go back to check the circuit connection and the code.

3-4 Put it into the toilet paper holder

After we make sure the FSR is working, we are going to put the circuit into the toilet paper holder.

We can temporarily unplug the FSR's wires from the mini breadboard. Then, remove the left board of the shell, push the rod from the hole on the right side of the shell to the left side, and put the toilet paper roll on the rod. Put the left board back and make sure the shaft goes through the hole on the left side.

After that, let the two wires go through the higher little hole on the right side (Figure 3-3). Put your mini breadboard and power supply in the higher level of the shell and connect the two wires back to the breadboard.

The whole setting will be similar to what is shown in Figure 3-4 except the openable board should be the right board instead of the front one.

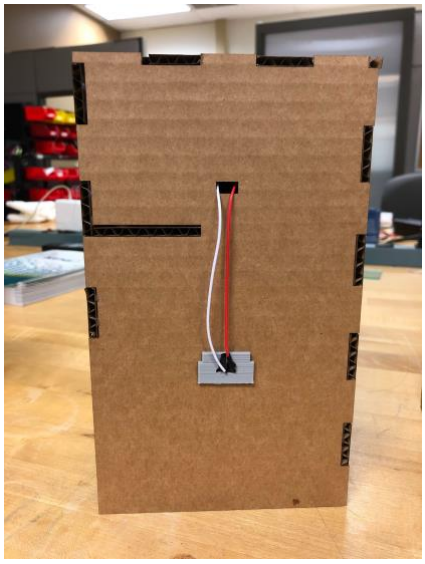

Figure 3-3. The two wires of the FSR go through the small hole on the right board to the higher level.

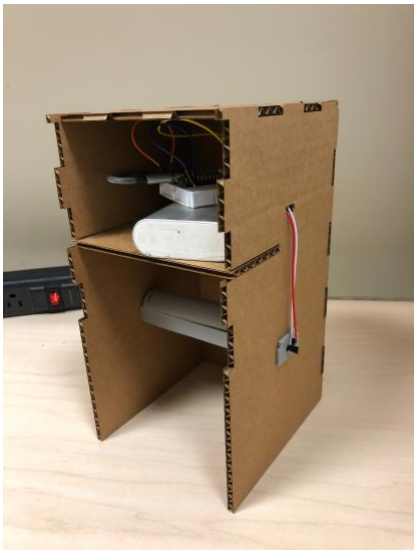

Figure 3-4 The rod and circuit are integrated into the shell.

Step 4: Registering IFTTT Service

Our monitor system sends a reminder to a Slack channel when a toilet paper roll is running low. We are going to implement this feature using IFTTT. You will need an IFTTT account to do the following steps.

4-1 Sign up or sign in IFTTT account

Go to the website of IFTTT (https://ifttt.com). If you do not have an account, click the Sign up button on the top-right corner of the page and go through the process to create an account; if you already have one, click Sign in to login you account (Figure 4-1).

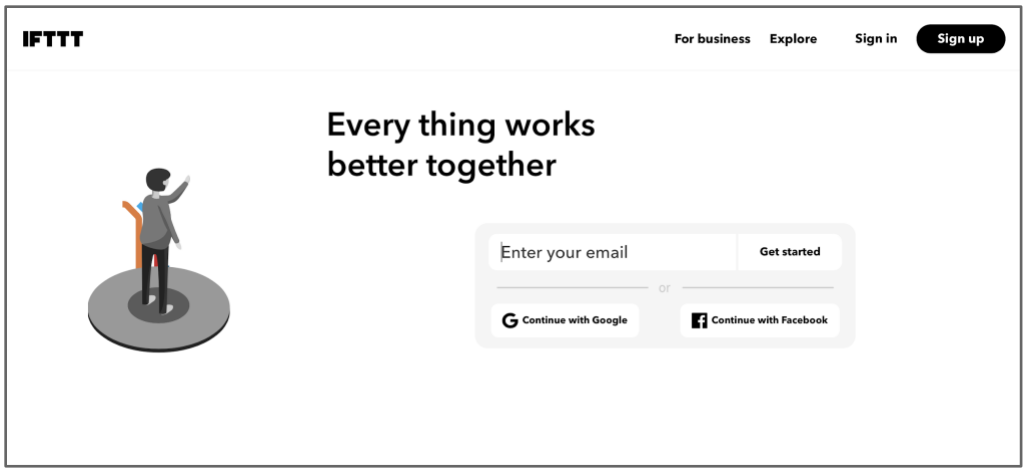

Figure 4-1. The index page of IFTTT. The Sign in and Sign up buttons are on the top-right corner.

4-2 Create a service

After signing in your IFTTT account, click on the avatar on the top-right corner, and click Create in the dropdown menu (Figure 4-2).

Figure 4-2. Click Create in the dropdown menu.

4-3 Connect to Webhooks

After clicking Create, now we are in a webpage showing “If + This Then That”. Now, let’s click “+ This” (Figure 4-3).

Figure 4-3. A web page showing “If + This Then That”.

You will see a service list after clicking “+ This”. Find Webhooks (Figure 4-4) and click it, then click Connect (Figure 4-5).

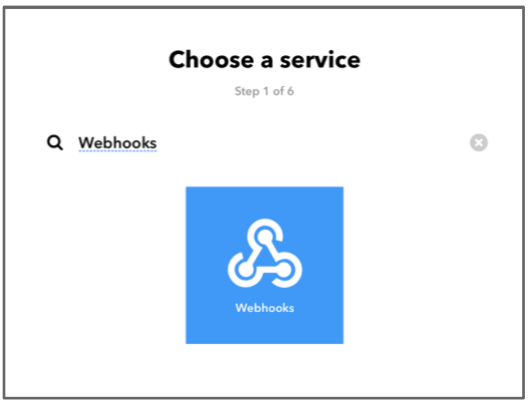

Figure 4-4. Find Webhooks in Choose a service.

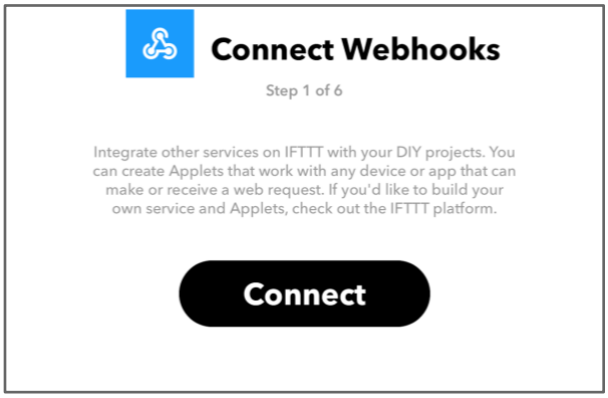

Figure 4-5. The Connect Webhooks page.

4-4 Create trigger

We have to decide what event will trigger our IFTTT service. Click Receive a web request to choose it as the trigger event (Figure 4-6).

Then in the Complete trigger fields page, enter `paper_running_low` to be the event name, and click Create trigger (Figure 4-7).

Figure 4-6. Choose trigger.

Figure 4-7. Enter “paper_running_low” as the event name.

4-5 Set action

After we create a trigger, now we see a webpage similar to Figure 4-3, but with a little difference -- “+ This” is now a webhook icon, and “That” becomes clickable (Figure 4-8). Our next step is to set the action after the trigger is activated. Click “+ That”.

Figure 4-8. A web page showing “If Webhook Then + That”.

In the action service list, choose Slack (Figure 4-9). A window will pop up, asking you which Slack workspace you want to connect. Choose one at your wish, then click Post to channel in the Slack action list (Figure 4-10).

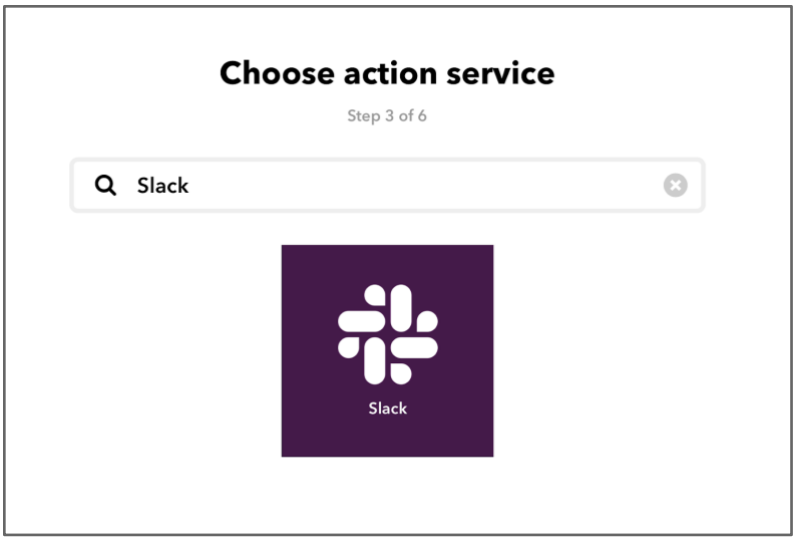

Figure 4-9. Choose Slack in Choose action service.

Figure 4-10. Choose Post to channel in Choose action.

Then in the Complete action fields page, choose a channel you want to send notifications to, and fill the Message field with
`Paper #{{Value1}} in {{Value2}} is running low!`
You can leave the other fields blank or fill in the text you prefer.

After you finish filling the fields, click the “Create action“ button at the bottom of the page (Figure 4-11).

Figure 4-11. Choose a channel and fill in the message.

4-6 Review service settings

Check if the setting is correct. If things look right, click Finish (Figure 4-12).

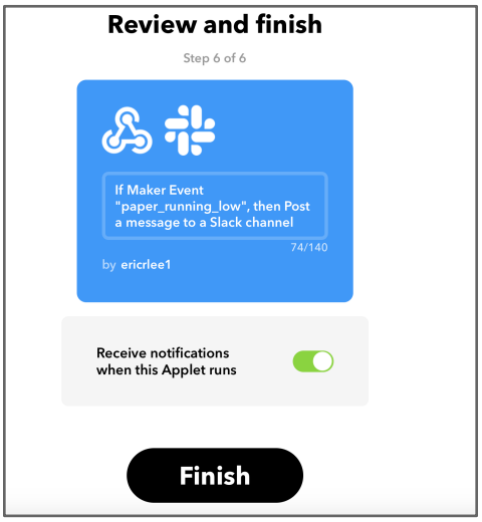

Figure 4-12. Review and finish service settings.

4-7 Get API url

We have set up an “If this then that” rule. If the webhook is triggered, then a message will be sent to the designated channel. But how do we trigger the webhook? Now we are going to find the trigger.

Click the Webhook icon, (Figure 4-13) then click on the “Setting” button on the top-right corner (Figure 4-14).

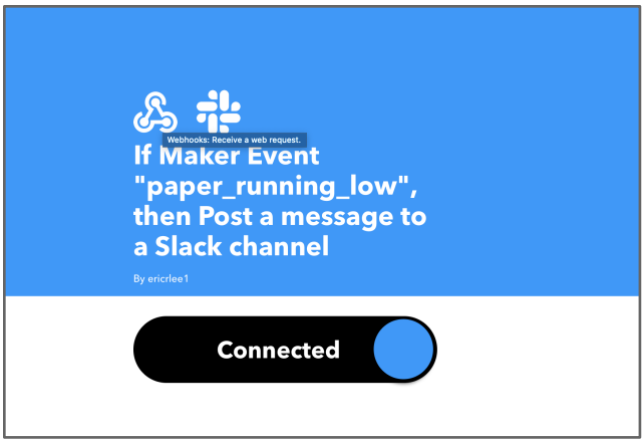

Figure 4-13. Click the Webhook icon.

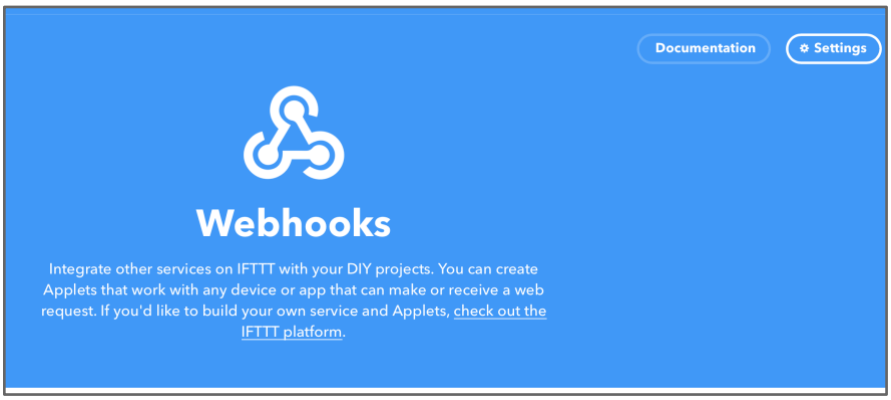

Figure 4-14. Click the Settings button on the top-right corner of the webpage.

Now you see the Webhooks Settings page. In the Account Info section, there is a url (Figure 4-15). Leave this webpage open or copy this url and record it somewhere. We will need this url later.

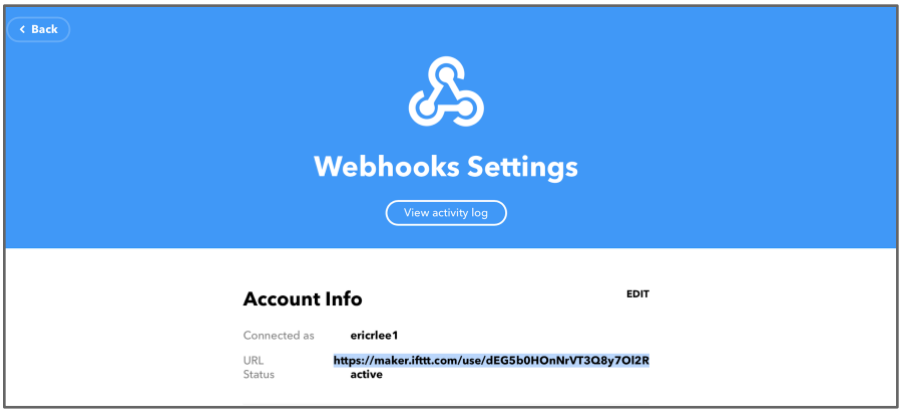

Figure 4-15. There is a url on the Webhooks Settings page.

Step 5: Setting up the monitor web system

Almost done! Now we are going to set up the monitor web system. We will set up the environment, download the code, do some edit, and then run the system.

5-1 Install Yarn on your computer

Install Yarn on your computer following the guide on Yarn's website: https://yarnpkg.com/en/docs/install

5-2 Download the project code from the repository

Go to our project repository on GitHub (https://github.com/Reedo0910/ToiletPaperMonitor), click the green button Clone or download, then choose Download ZIP (Figure 5-1).

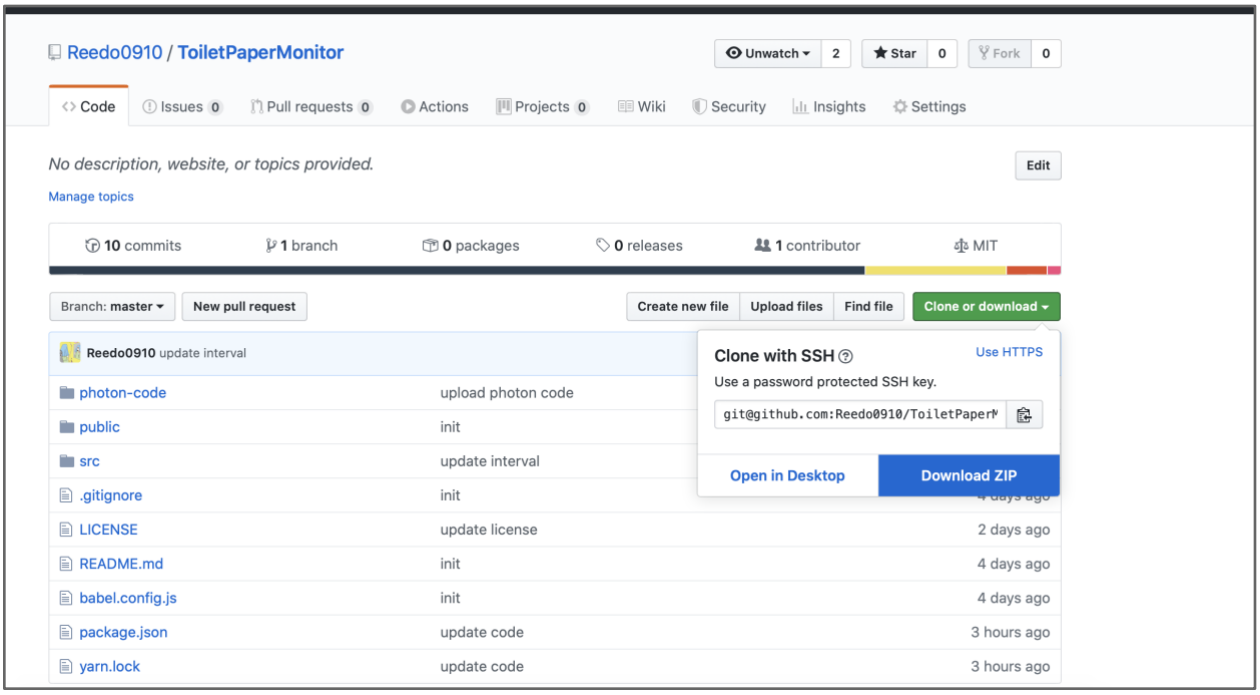

Figure 5-1. The GitHub repository for this project.

5-3 Install dependencies

Before we run the web system, there are some dependencies we must install. Open the terminal and move to the root of the project folder (If you are unfamiliar with terminal commands, here is a cheat sheet: https://github.com/0nn0/terminal-mac-cheatsheet).

Then, enter `yarn install` to install the project's dependencies.

5-4 Set APIs

Create a `.env` file at the root of the project folder.

Then, edit the file following the sample code below. Fill in the { } blank using the corresponding API URLs from the previous steps. Your Photon access API url will be in this format: `https://api.particle.io/v1/devices/{YOUR DEVICE ID}/force_level?access_token={YOUR ACCESS TOKEN}`. The device ID can easily be found on the My Devices page of your Particle console; the access token can be found in the Particle web IDE. Your IFTTT access API url is the one you got in step 4-7.

```
VUE_APP_PHOTON_ACCESS_API_1 = {YOUR PHOTON 1 ACCESS API URL }

VUE_APP_PHOTON_ACCESS_API_2 = {YOUR PHOTON 2 ACCESS API URL }

VUE_APP_IFTTT_ACCESS_API = {YOUR IFTTT ACCESS API URL }
```

Figure 5-2. The device ID of your Photon is displayed in the My Devices page of the Particle console..

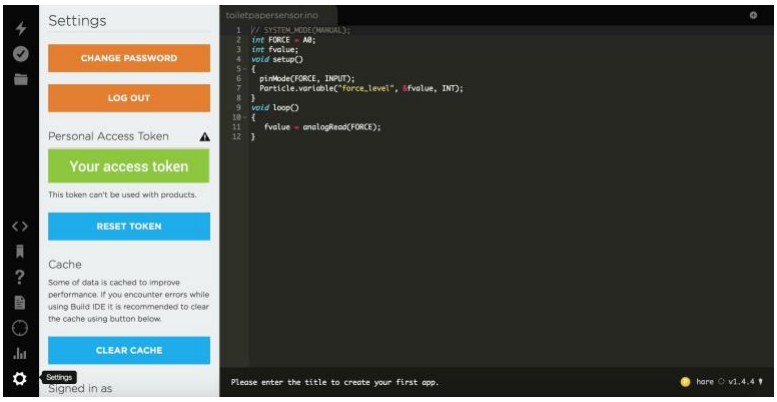

Figure 5-3. The access token can be found in the Settings section of the Particle web IDE.

5-5 Running the web app

Run the project by entering the command `yarn serve` in the terminal at the root of the project folder.

Then, you can see the web app (Figure 5-4) is running on `http://localhost:8080`!

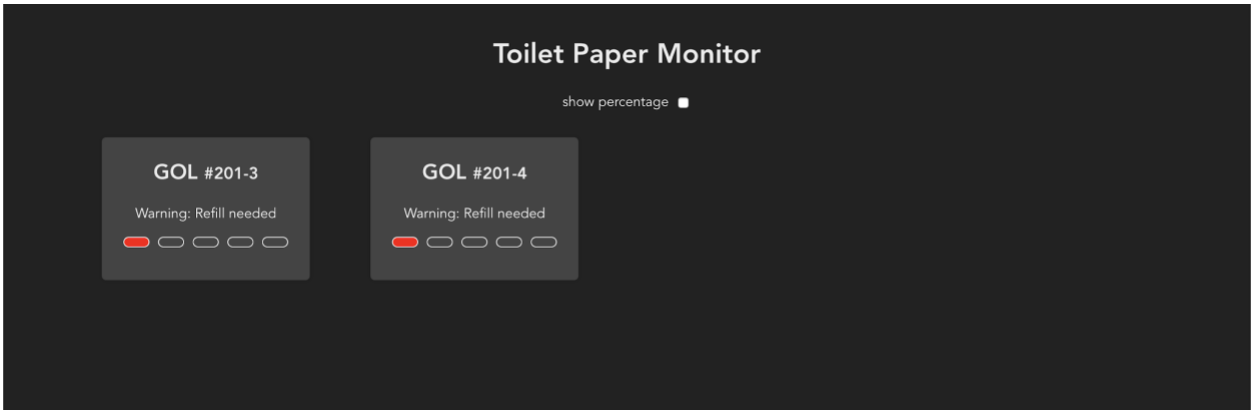

Figure 5-4. The toilet paper monitor web system.

You can see the remaining amount of your toilet paper roll in the holder

There are two toilet paper roll status in Figure 5-4 because we produced 2 toilet paper holders and connected them to the system.