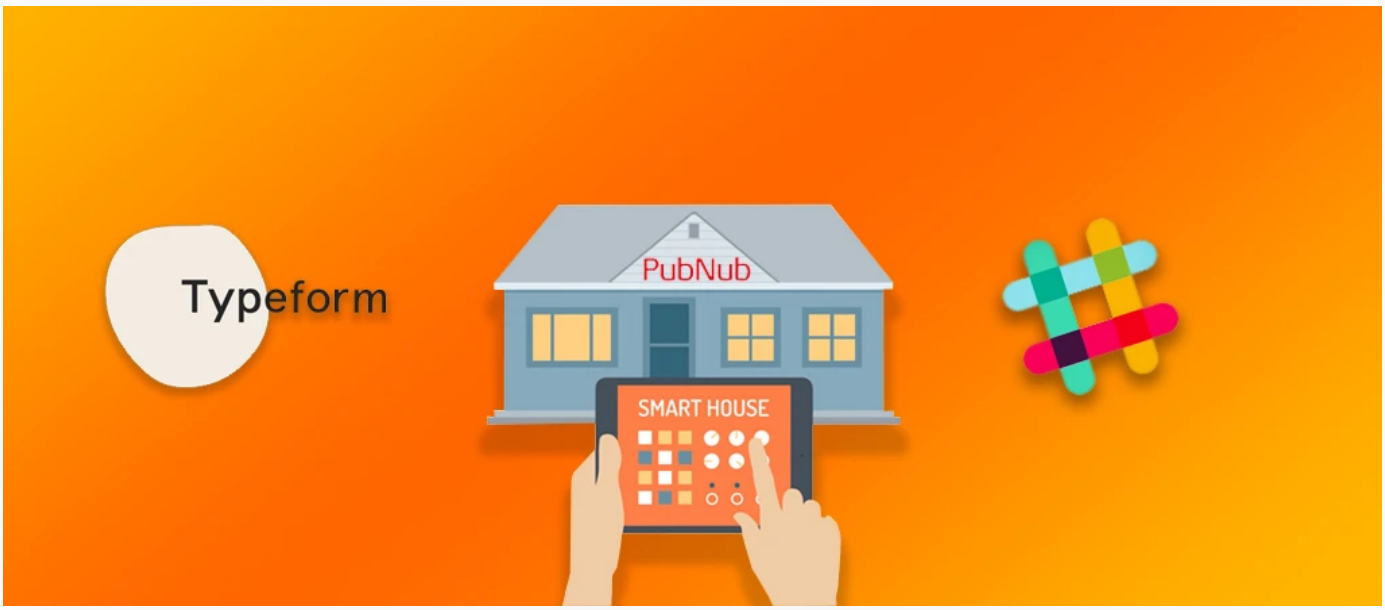
Securely Unlocking Your Front Door with Typeform, PubNub Functions and Slack

How to build an intelligent remote door lock that allows you to grant and revoke access to the door with Typeform, Slack and PubNub Functions.

JORDAN SCHUETZ
@jordanschuetz
Developer Advocate, MuleSoft

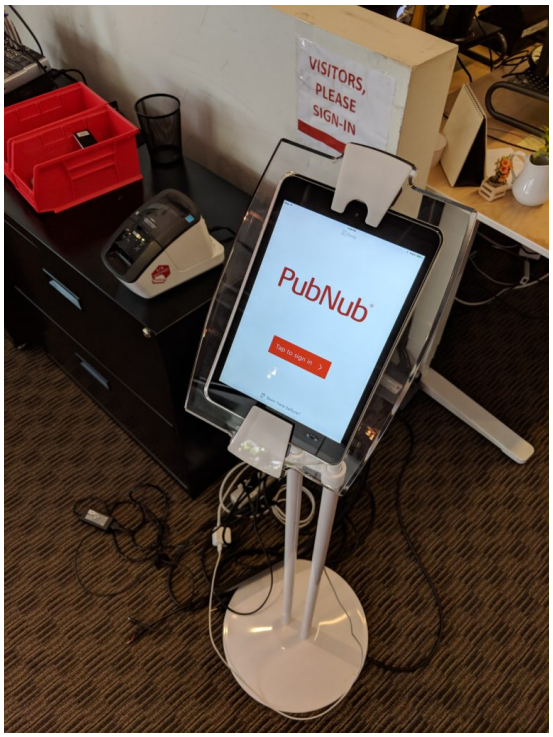


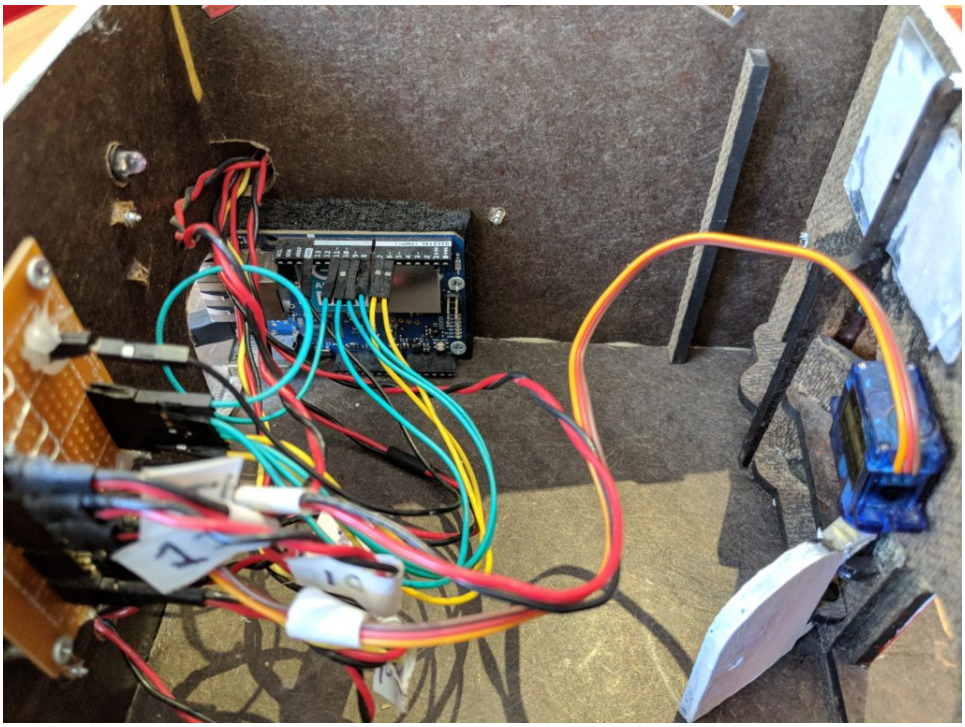
The days of carrying around house keys are numbered. Rejoice! In this tutorial, we're going to walk you through how to build a connected home lock with a couple really cool APIs – Typeform API and PubNub Functions. By the end, we'll have an Envoy-style login form that houseguests can use to check into a business or residence.

We'll also explain how our sample model house was built and how we used an Arduino Yun to power servo motors that open and close the front door. Lastly, we will discuss how we used the Slack API for authentication on whether the door should be open or closed.

Typeform

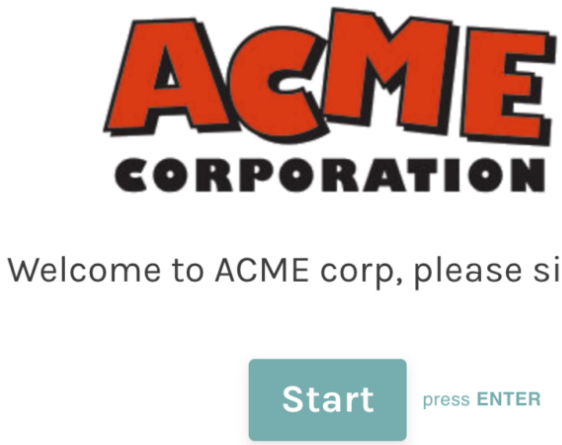
Have you ever been to a business and seen one of these kiosks asking you to sign in?



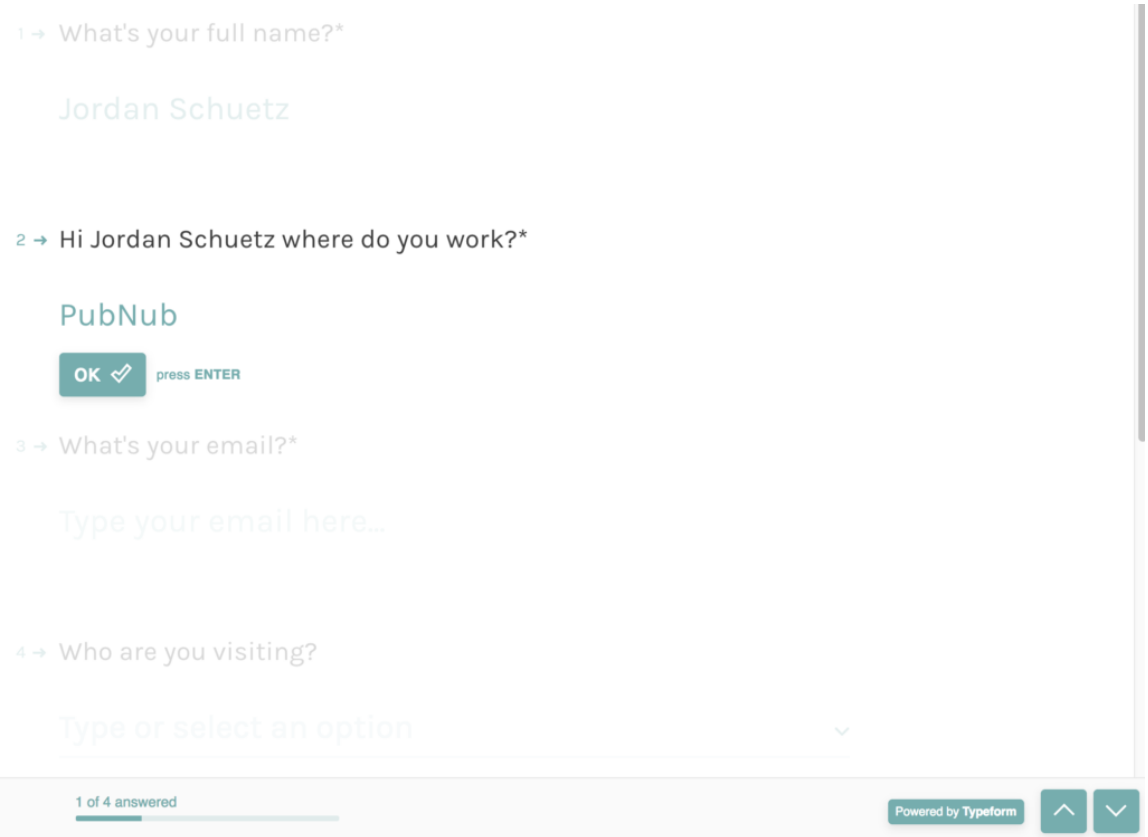


The goal of this tutorial is to showcase how you can use both Typeform and PubNub Functions to create a simple IoT door lock. To see the process on how we created the model house, check out this blog post.

Typeform is free to try to out, and allows you to create beautiful looking form interfaces that you can customize for your application. Below is what the user would see on the kiosk on the front door of a business or home.



Once the user clicks the Start button, they have to fill out a variety of information that we setup in the Typeform dashboard.



When the user fills out the entire form, and click the submit button, a REST API call is made to a PubNub Function which is coded below. The PubNub Function is using the OnRequestevent type to listen for an event from Typeform.

Slack

Once the form has been submitted, the PubNub Function sends the information filled out in the form, and sends it over to Slack's API. We use a web-hook to ping Slack and send a message to the users in the Slack Channel that someone is at the front door.

To learn more about how to create your own Slack Incoming webhook, check out the documentation here. In order for the front door to open, a user in the Slack channel has to click the Yes button.

```
export default (request, response) => {
    const pubnub = require('pubnub');
    const xhr = require('xhr');
    const db = require("kvstore");

    console.log('request',request);
    //extract from typeform
    const tf_data = JSON.parse(request.body);
    console.log('tf_data', tf_data)
    const { answers } = tf_data.form_response;

    console.log('answers', answers);

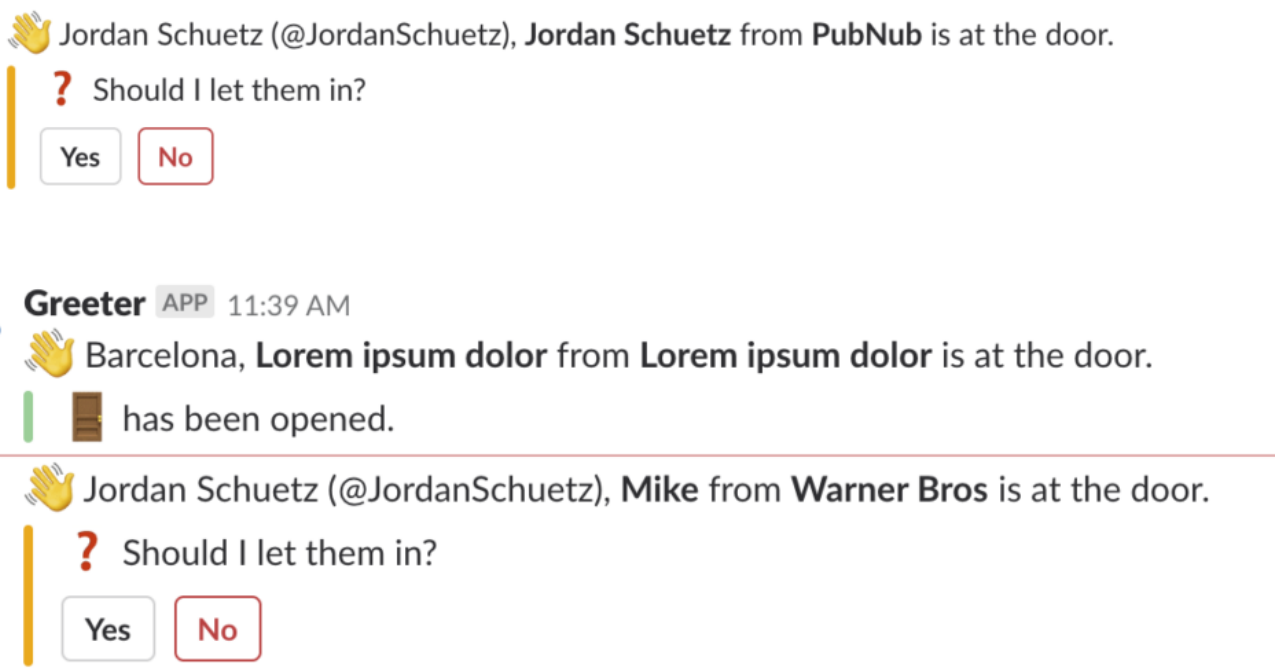
    const full_name = answers.filter(a => a.field.id==='shiw8XAduZWu')[0].text;
    const company = answers.filter(a => a.field.id==='Tx8ZY4soSdrk')[0].text;
    const email = answers.filter(a => a.field.id==='OlvdHHQ58mN2')[0].text;
    let host_name = answers.filter(a => a.field.id==='VkkU6SMbEQmT')[0].choice.label;
    host_name = host_name.replace(/\(\@/,'(<@').replace(/\)/,'>)'); //format to slack

    //format slack message
    let message = `:wave: ${host_name}, *${full_name}* from *${company}* is at the door.`;

    //send to slack
    const http_options = {
        "headers": {
            "Content-Type": "application/json"
         },
        "method": "POST", // or PUT
        "body": JSON.stringify({
            "text": message,
            "link_names":1,
            "parse":"full",
            "attachments": [{
            "text": ":question: Should I let them in?",
            "fallback": ":question: Should I let them in?",
            "callback_id": "wopr_game",
            "color": "#FFA500",
            "attachment_type": "default",
            "actions": [
                {
                    "name": "open_door",
                    "text": "Yes",
                    "type": "button",
```

```
                    "value": true
                },
                {
                    "name": "open_door",
                    "text": "No",
                    "style": "danger",
                    "type": "button",
                    "value": false,
                    "confirm": {
                        "title": "Are you sure?",
                        "text": "You wouldn't want them to come steal your candies :candy:, would you?",
                        "ok_text": "Yes",
                        "dismiss_text": "No"
                    }
                }
            ]
        }
    ]
        })
    };

    const url = "SLACK-WEBHOOK-URL-HERE";
    return xhr.fetch(url, http_options).then((x) => {
        //console.log(body);
        return response.send();
    });
};
```



Once a user in the Slack channel clicks YES, Slack will send a message to an addition PubNub function. If the action.name of the request is equal to open_door, the PubNub Function will send a message to the pubnub_iot_house channel telling the IoT house to open

the front door. Additionally, the code will send a response to Slack to display to the user that the door has been opened.

```
export default (request, response) => {
    const kvstore = require('kvstore');
    const xhr = require('xhr');
    const PubNub = require("pubnub");
    let pubnub = new PubNub({
        publishKey: 'demo',
        subscribeKey: 'demo'
      });

    let bodyString = request.body;
    console.log('request',request); // Log the request envelope passed
    var p = request.body
    var c = request.body.replace(/payload\=/g, '')
    console.log('payload', c)
    const payload = JSON.parse(decodeURIComponent(c))

    console.log('interactive', payload)
    const action = payload.actions[0]
    var msg, type = ''

    if(action.name === "open_door"){
        if(action.value){ //open_door
            pubnub.publish({
                "channel": "pubnub_iot_house",
                "message": {
                    name: 'garage',
                    value: 1
                  }
            })
            msg = `:door: has been opened.`
        }else{
            msg = `:no_entry_sign: will remain closed.`
        }
    }
    // // Set the status code - by default it would return 200
    response.status = 200;
    return response.send({
        text: payload.original_message.text.replace(/\+/g,' '),
        fallback: payload.original_message.text.replace(/\+/g,' '),
        link_names:1,
        parse:"full",
        attachments:[{
              "fallback": msg,
              "text": msg,
              "color": "#7CD197"
          }],
        replace_original: true,
        response_type: 'in_channel'
      })
};
```



The front door is now open on the IoT house once Yes was clicked on Slack!



That’s It!

It’s really that easy to setup a fully working IoT demo with Typeform, PubNub Functions and Slack. Make sure to go register for a Typeform account so you can start implementing custom forms in your application. If you have any questions about this demo, contact schuetz@pubnub.com.