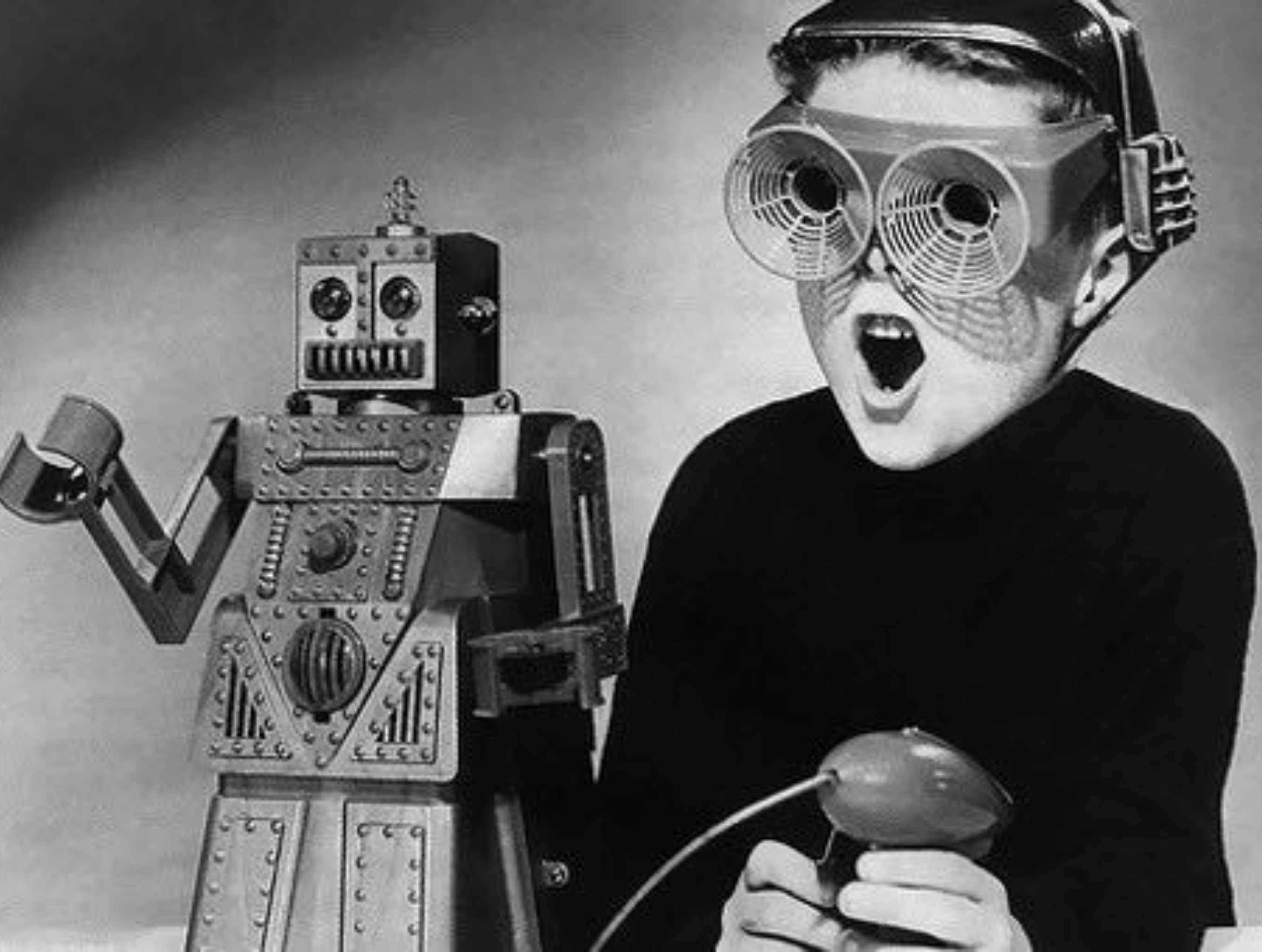




SERVERLESS FUNCTIONS AND VUE.JS



SUCCESS!







NOT FAIL...

Wanna have some fun?

1:55 PM

No

Delivered 1:55 PM



SERVERLESS

AN ACTUALLY INTERESTING THING WITH A CLICKBAITY TITLE

FaaS

FUNCTIONS AS A SERVICE

SARAH DRASNER

@SARAH_EDO



BENEFITS

- You pay only for what you use (usually a lot less)
- Less time babysitting a server
- Same idea as functional programming, breaking things into small bits
- Principle of least power

SERVICES

WHAT YOU MANAGE

APPLICATION

DATA

RUNTIME

OS

HARDWARE

INTEL

[LINK TO CODEPEN](#)

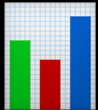
Traditional Server-Side Software

SERVERLESS
ALL THE
THINGS?

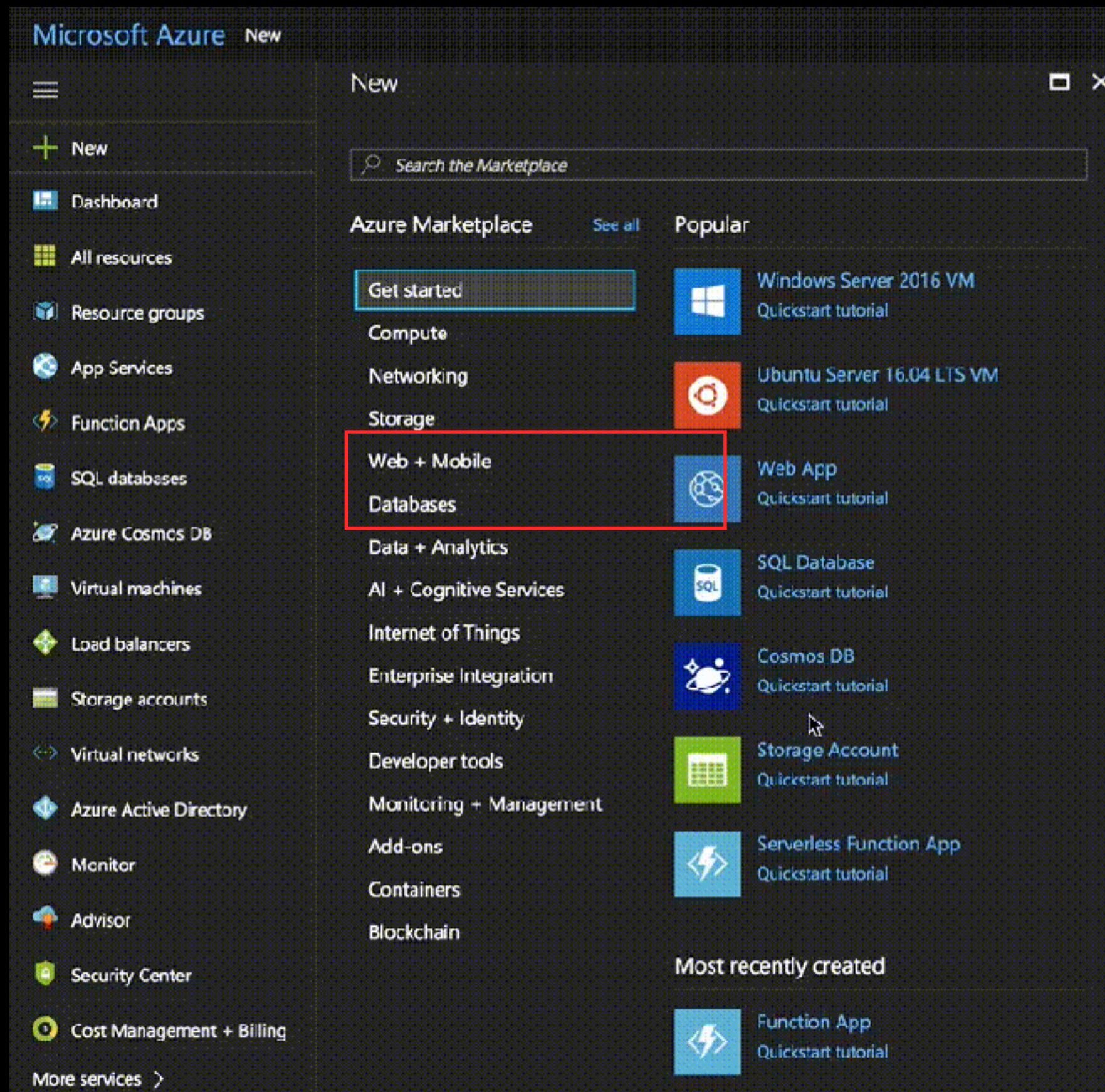
NO, NOT REALLY.



SO WHAT IS IT GOOD FOR?

- Clean data on a cron job 
- Take data and use it to create data visualizations 
- Crop images uploaded by the user and create a user profile 

Free account signup



Consumption Plan

index.js

Save

Run

</> Get function URL

```
1 module.exports = function (context, req) {  
2   context.log('yoyoyoyo');  
3  
4   context.res = {  
5     body: "Hello " + req.body.name  
6   };  
7  
8   context.done();  
9 };
```

Logs

Pause Clear Copy logs Expand

2018-02-02T17:56:22 Welcome, you are now connected to log-streaming service.

View files

Test

HTTP method

POST

Query

There are no query parameters

+ Add parameter

Headers

There are no headers

+ Add header

Request body

```
1 {  
2   "name": "delicious tacos"  
3 }
```

Output

Run

FIRST USE CASE

VUE.JS, STRIPE, AND SERVERLESS

VISIT THE SITE

SAMPLE VUE SHOP



Women's

Men's

Sale



Shoppity

Highest Price: \$300



Only Show Sale Items



Special Sale!

Shop now because half our items are greatly reduced

Contact Us

Questions? Call us at 1-888-555-SHOP, we're happy to be of service.

Khaki Suede Polish Work Boots



SALE

\$149.99

ADD ITEM

Camo Fang Backpack Jungle



\$39.99

ADD ITEM

Parka and Quilted Liner Jacket



SALE

\$49.99

ADD ITEM

Cotton Black Cap



SALE

Knit Sweater with Zips

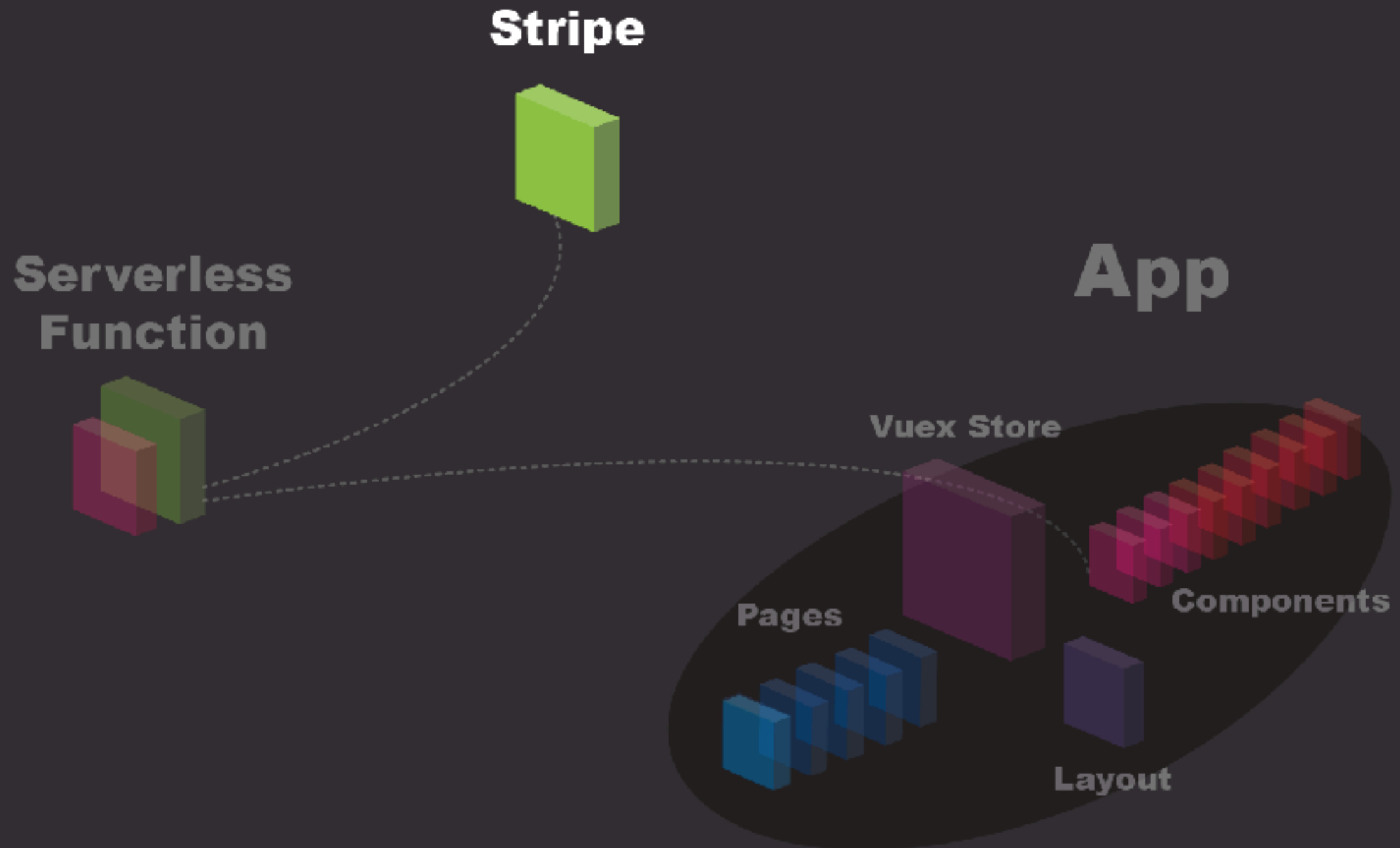


Long Linen-blend Shirt



VISIT THE REPO

THE APPLICATION



UNDER API

Unnamed Account

Search...

Home

Activate your account

Payments

Payouts

Customers

Radar

Subscriptions

Connect

Products

API

Events & logs

Viewing test data

Beta settings

Business settings

API Webhooks

Version

You're using API version 2017-08-15 LATEST

TEST DATA

API keys

Viewing test API keys, toggle to view live keys

Learn more about API authentication

Viewing test data

Standard API keys

NAME	TOKEN	LAST USED	CREATED
Publishable key		—	Nov 25, 2017
Secret key	Reveal test key token	Nov 25, 2017	Nov 25, 2017

Restricted API keys

Create restricted key

NAME	TOKEN	LAST USED	CREATED
No restricted keys			

What Stripe expects: Express

```
app.get('/', (req, res) => res.render('index.pug', { keyPublishable }));

app.post('/charge', (req, res) => {
  let amount = 500;

  stripe.customers
    .create({
      email: req.body.stripeEmail,
      source: req.body.stripeToken
    })
    .then(customer =>
      stripe.charges.create({
        amount,
        description: 'Sample Charge',
        currency: 'usd',
        customer: customer.id
      })
    )
    .then(charge => res.render('charge.pug'));
});

app.listen(4567);
```

Where we're going we don't need Express!



Kick things off!

```
var stripe = require('stripe')('sk_test_XXXXXXXXXXXXXXXXXXXX');  
// ^ this is a stripe testing key  
  
module.exports = function(context, req) {  
  context.log('starting to get down');  
}
```

If we have what we need,
create the customer and then the charge

```
if (
  req.body &&
  req.body.stripeEmail &&
  req.body.stripeToken &&
  req.body.stripeAmt
){
  stripe.customers
    .create({
      email: req.body.stripeEmail,
      source: req.body.stripeToken
    })
    .then(customer => {
      context.log('starting the stripe charges');
      stripe.charges.create({
        amount: req.body.stripeAmt,
        description: 'Sample Charge',
        currency: 'usd',
        customer: customer.id
      });
    })
  ...
}
```

Then finish, otherwise error logs

```
...  
.then(charge => {  
  context.log('finished the stripe charges');  
  context.res = {  
    // status: 200  
    body: 'This has been completed'  
  };  
  context.done();  
})  
.catch(err => {  
  context.log(err);  
  context.done();  
});  
} else {  
  context.log(req.body);  
  context.res = {  
    status: 400,  
    body: "We're missing something"  
  };  
  context.done();  
}  
};
```

index.js

Save

▶ Run

</> Get function URL

```
1 var stripe = require('stripe')('sk_test_js5LRkmS70sYHtVc7XYvK9OB');
2 // ^ this is a stripe testing key
3
4 module.exports = function(context, req) {
5   context.log('starting to get down');
6
7   //If we have a request body, an email, and a token, let's get started
8   if (
9     req.body &&
10     req.body.stripeEmail &&
11     req.body.stripeToken &&
12     req.body.stripeAmt
13   ) {
14     stripe.customers
15       .create({
16         email: req.body.stripeEmail,
17         source: req.body.stripeToken
18       })
19       .then(customer => {
20         context.log('starting the stripe charges');
```

Logs

|| Pause Clear Copy logs Expand ▼

```
2018-01-15T17:21:26 Welcome, you are now connected to log-streaming service.
```

View files Test

Headers

There are no headers

+ Add header

Request body

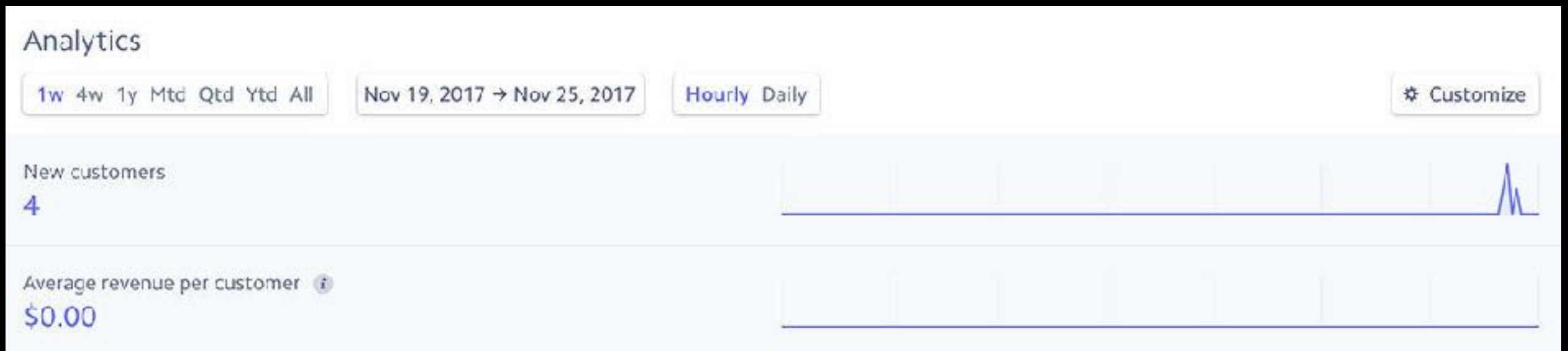
```
1 {
2   stripeEmail: "nobody@example.com",
3   stripeToken: "tok_visa",
4   stripeAmt: "50.00"
5 }
```

Output

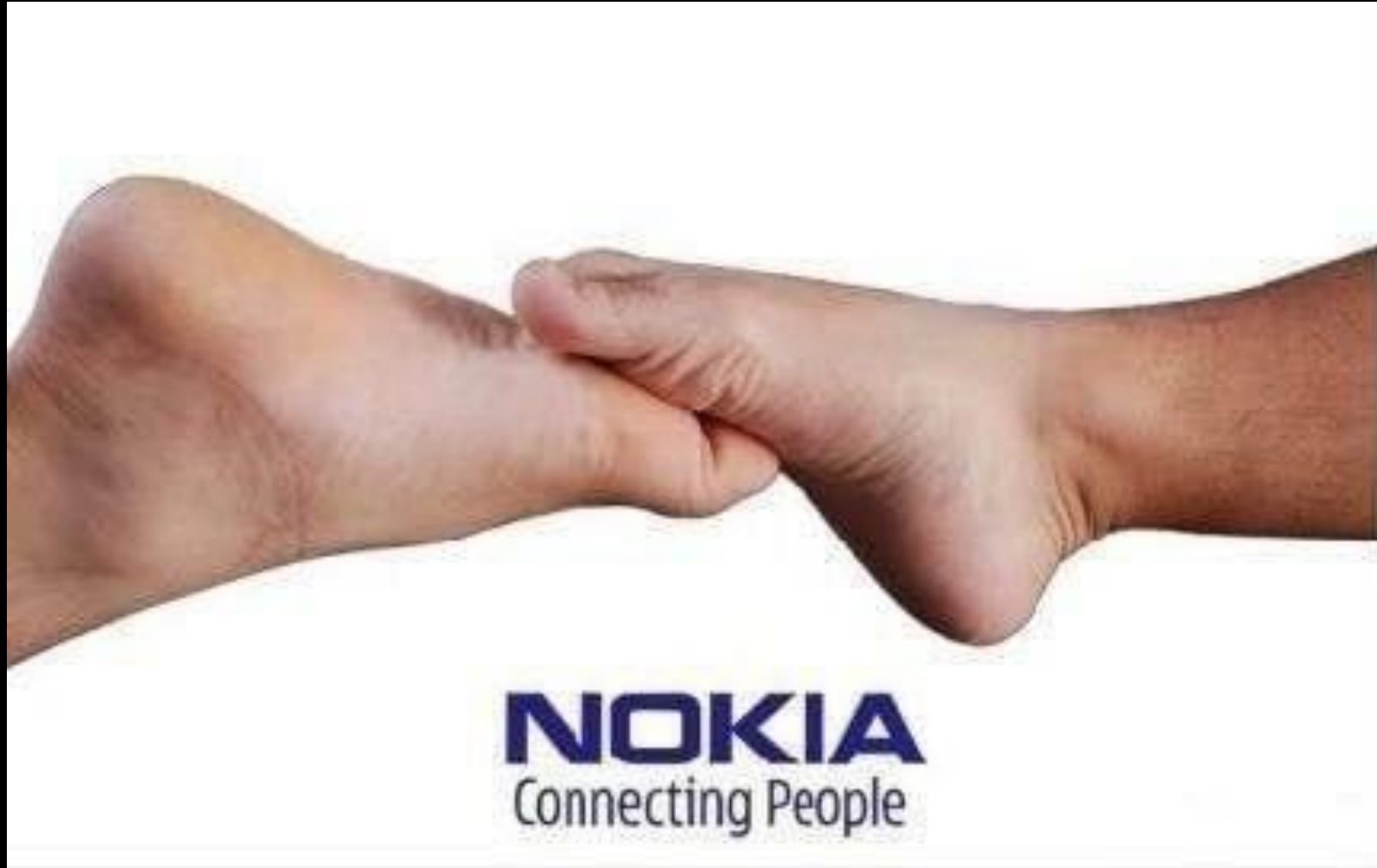
▶ Run

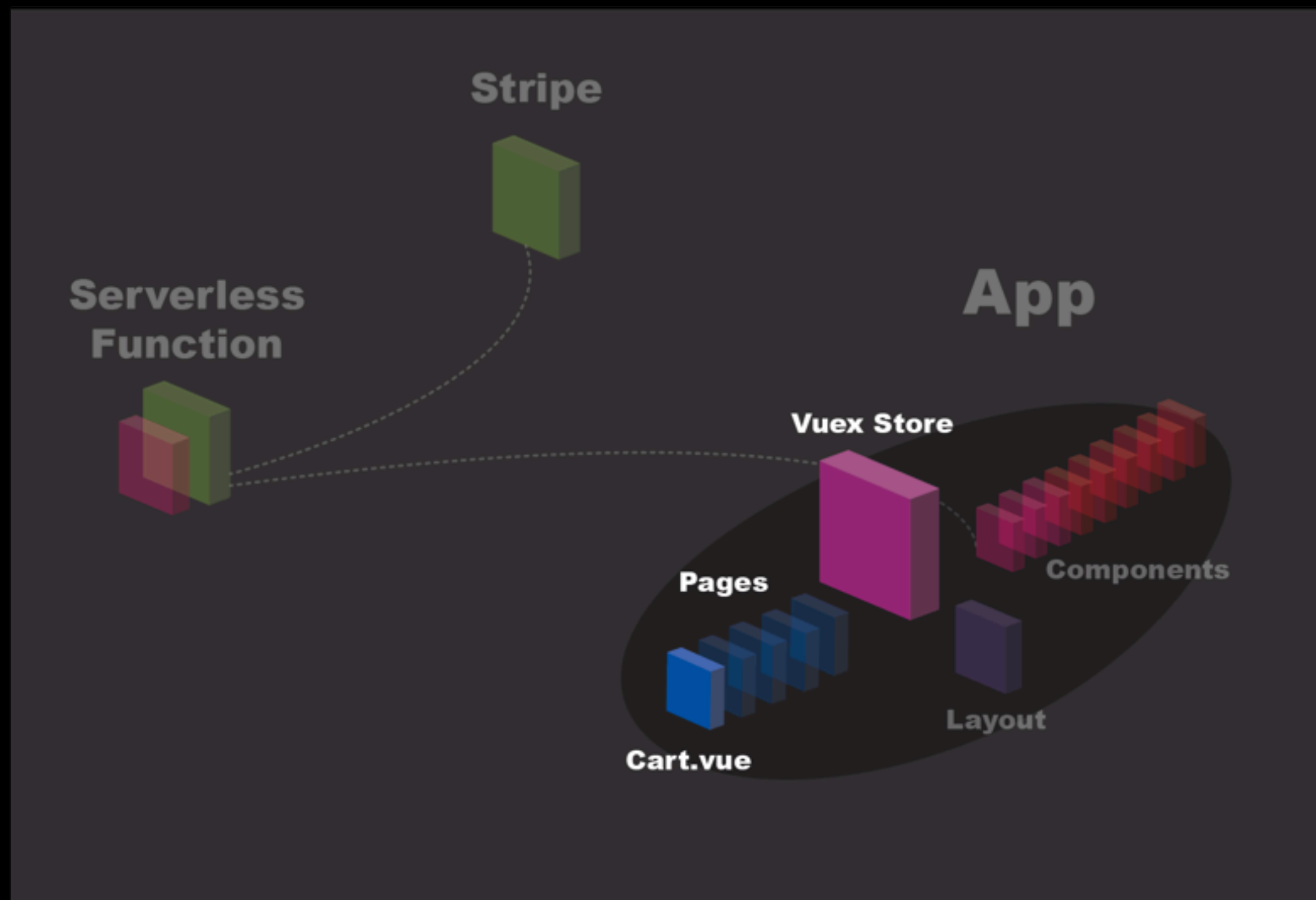


SUCCESS!



POST ON CSS-TRICKS





OUR CART

VUEX HAS A MANIFEST OF ALL OF OUR PRODUCTS, ITEMS, AND TOTAL

The screenshot shows the Shoppity website interface. At the top, there are navigation links for "Women's", "Men's", and "Sale", along with a shopping cart icon. The main banner features the "Shoppity" logo and images of people wearing jackets. Below the banner, there are three product cards: "Khaki Suede Polish Work Boots" priced at \$149.99, "Camo Fang Backpack Jungle" priced at \$39.99, and "Parka and Quilted Liner Jacket" priced at \$49.99. Each card has an "ADD ITEM" button. To the left of the product cards, there is a filter section with a price slider (set to \$0 to \$400), a toggle for "Only Show Sale Items" (set to "YES"), and a "Special Sale!" message stating "Shop now because half our items are greatly".

The Vuex DevTools extension is open at the bottom of the browser window. The "Base State" tab is selected, showing the following state:

```
state: {
  cart: Object (empty)
  cartTotal: 0
  products: Array(12)
  sale: false
}
getters: {
  men: Array(6)
  sale: Array(5)
}
```

Cart.vue Base Logic

```
<div v-if="cartTotal > 0">  
  <!--we'll add our checkout here-->  
</div>
```

```
<div v-else-if="cartTotal === 0 && success === false" class="empty">  
  <!--we'll add our empty state here-->  
</div>
```

```
<div v-else>  
  <!--we'll add success here-->  
</div>
```

OUR CART

```
<div v-if="cartTotal > 0">
  <h1>Cart</h1>

  <div class="cartitems"
    v-for="item in cart"
    key="item">
    <div class="carttext">
      <h4>{{ item.name }}</h4>
      <p>{{ item.price | usdollar }} x {{ item.count }}</p>
      <p>Total for this item: <strong>{{ item.price * item.count }}</strong></p>
    </div>
    
  </div>

  <div class="total">
    <h3>Total: {{ total | usdollar }}</h3>
  </div>

  <!--we're going to add our checkout here-->
</div>
```

Cart

Camo Fang Backpack Jungle

\$39.99 x 1

Total for this item: **39.99**



Parka and Quilted Liner Jacket

\$49.99 x 1

Total for this item: **49.99**



Total: \$89.98

```
computed: {  
  ...  
  total() {  
    return Object.values(this.cart)  
      .reduce((acc, el) => acc + (el.count * el.price), 0)  
      .toFixed(2);  
  }  
}
```

Cart

Camo Fang Backpack Jungle

\$39.99 x 1

Total for this item: **39.99**



Parka and Quilted Liner Jacket

\$49.99 x 1

Total for this item: **49.99**



Total: \$89.98

Please enter your payment details:

Email

name@example.com

Credit Card

Test using this credit card: **4242 4242 4242 4242**, and
enter any 5 digits for the zip code

 Card number MM / YY CVC

PAY WITH CREDIT CARD

EASIEST WAY

VUE STRIPE CHECKOUT

REPO

Vue Stripe Checkout



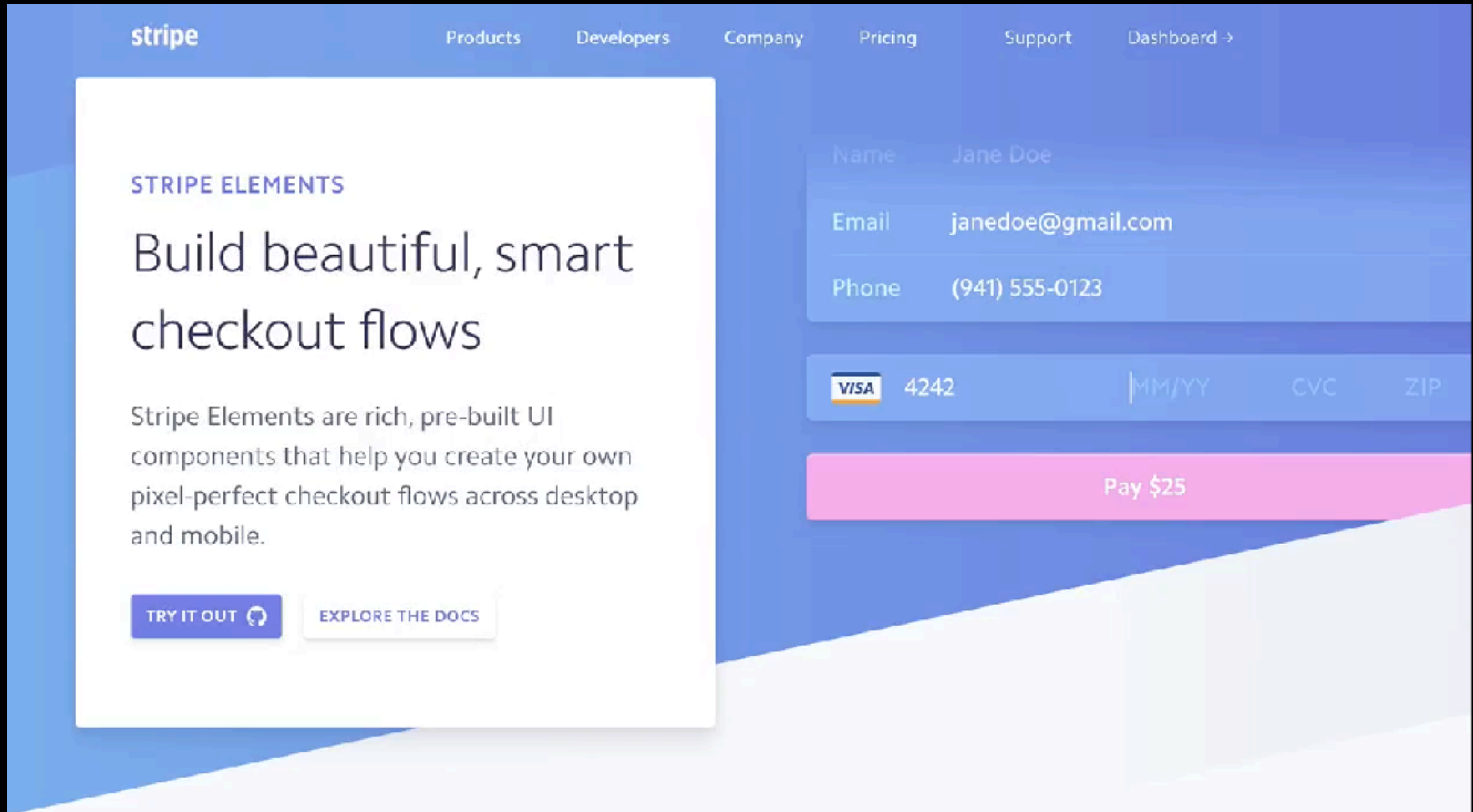
SHUT UP AND TAKE MY MONEY!

Sample card:

4242 4242 4242 4242

WE'LL USE:

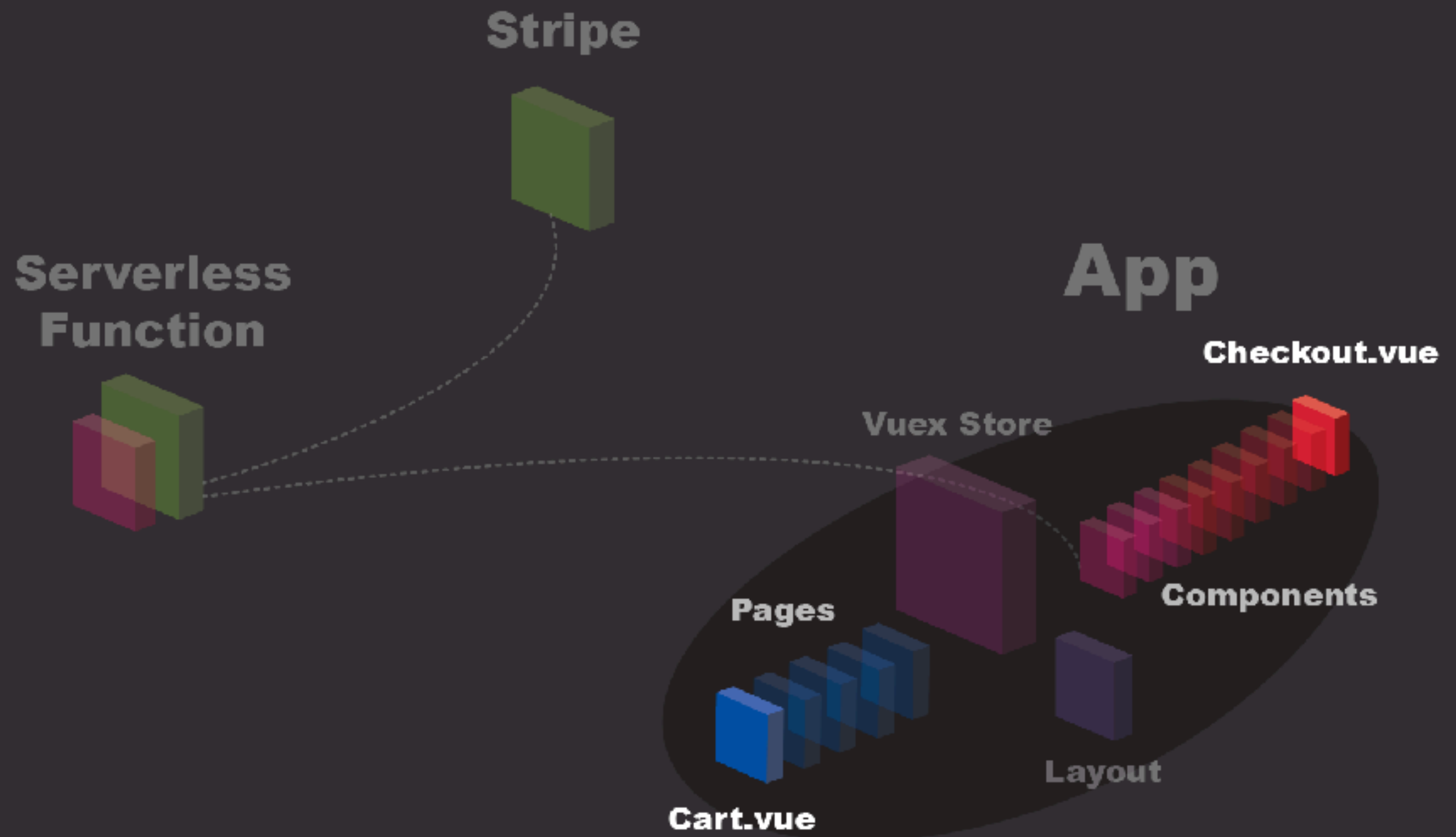
STRIPE ELEMENTS



```
npm i vue-stripe-elements-plus
```

OR

```
yarn add vue-stripe-elements-plus
```



DEFAULT STRIPE-ELEMENT

```
<template>
  <div id='app'>
    <h1>Please give us your payment details:</h1>
    <card class='stripe-card'
      :class='{ complete }'
      stripe='pk_test_XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
      :options='stripeOptions'
      @change='complete = $event.complete'
    />
    <button class='pay-with-stripe'
      @click='pay'
      :disabled='!complete'
    >
      Pay with credit card
    </button>
  </div>
</template>
```

DEFAULT STRIPE-ELEMENT

```
import { stripeKey, stripeOptions } from './stripeConfig.json'
import { Card, createToken } from 'vue-stripe-elements-plus'

export default {
  data () {
    return {
      complete: false,
      stripeOptions: {
        // see https://stripe.com/docs/stripe.js#element-options for details
      }
    }
  },
  components: { Card },
  methods: {
    pay() {
      // createToken returns a Promise which resolves in a result object with
      // either a token or an error key.
      createToken().then(data => console.log(data.token))
    }
  }
}
```

WHAT WE'LL NEED:

```
data() {  
  return {  
    submitted: false,  
    complete: false,  
    status: '',  
    response: '',  
    stripeOptions: {  
      // you can configure that cc element.  
    },  
    stripeEmail: ''  
  }  
},
```

WHAT WE'LL NEED:

```
props: {  
  total: {  
    type: [Number, String],  
    default: '50.00'  
  },  
  success: {  
    type: Boolean,  
    default: false  
  }  
},
```

HERE WE GO!

```
pay() {
  createToken().then(data => {
    // we'll change this to know it was submitted
    this.submitted = true;
    // this is a token we would use for the stripeToken below
    console.log(data.token);
    axios
      .post(
        'https://sdras-stripe.azurewebsites.net/api/charge?code=zWwbn6LLqMxuyvwbWpTFXdRxFd7a27KCRCEseL7zEqbM9ijAgj1c1w==',
        {
          stripeEmail: this.stripeEmail, // send the email
          stripeToken: data.token.id, // testing token
          stripeAmt: this.total // send the amount
        },
        {
          headers: {
            'Content-Type': 'application/json'
          }
        }
      )
    ...
  })
}
```

index.js

Save

▶ Run

</> Get function URL

```
1 var stripe = require('stripe')('sk_test_ja5LRkmS7QsYHtVc7XYvK90B');
2 // * this is a stripe testing key
3
4 module.exports = function(context, req) {
5   context.log('starting to get down');
6
7   //if we have a request body, an email, and a token, let's get started
8   if (
9     req.body &&
10     req.body.stripeEmail &&
11     req.body.stripeToken &&
12     req.body.stripeAmt
13   ) {
14     stripe.customers
15       .create({
16         email: req.body.stripeEmail,
17         source: req.body.stripeToken
18       })
19       .then(customer => {
20         context.log('starting the stripe charges');
```

> View files
Test

HERE WE GO!

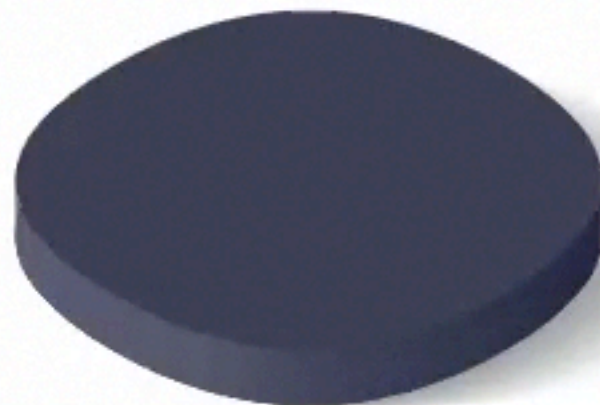
```
...  
.then(response => {  
  this.status = 'success';  
  this.$emit('successSubmit');  
  this.$store.commit('clearCartCount');  
  // console logs for you :)  
  this.response = JSON.stringify(response, null, 2);  
  console.log(this.response);  
})  
.catch(error => {  
  this.status = 'failure';  
  // console logs for you :)  
  this.response = 'Error: ' + JSON.stringify(error, null, 2);  
  console.log(this.response);  
});  
});
```

pay()

- Uses Axios to post to our function to our function URL
- Tracks whether we've submitted the form or not, with `this.submitted`
- It sends the email, token, and total to our serverless function
- If it's successful, commits to our Vuex store, mutation clears our cart, and emits to the parent cart component that the payment was successful.
- If it errors, it changes the status to failure, and logs the error response for help with debugging

WHILE THE FUNCTION WORKS ITS MAGIC ✨

[LINK TO CODEPEN](#)



IN THE CASE OF FAILURE



Template

```
<div v-if="status === 'failure'">
  <h3>Oh No!</h3>
  <p>Something went wrong!</p>
  <button @click="clearCheckout">Please try again</button>
</div>
```

Script

```
clearCheckout() {
  this.submitted = false;
  this.status = '';
  this.complete = false;
  this.response = '';
}
```

IN THE CASE OF SUCCESS



Total for this item: 29.99



Long Linen-blend Shirt

\$18.99 x 1

Total for this item: 18.99



Total: \$48.98

Please enter your payment details:

Email

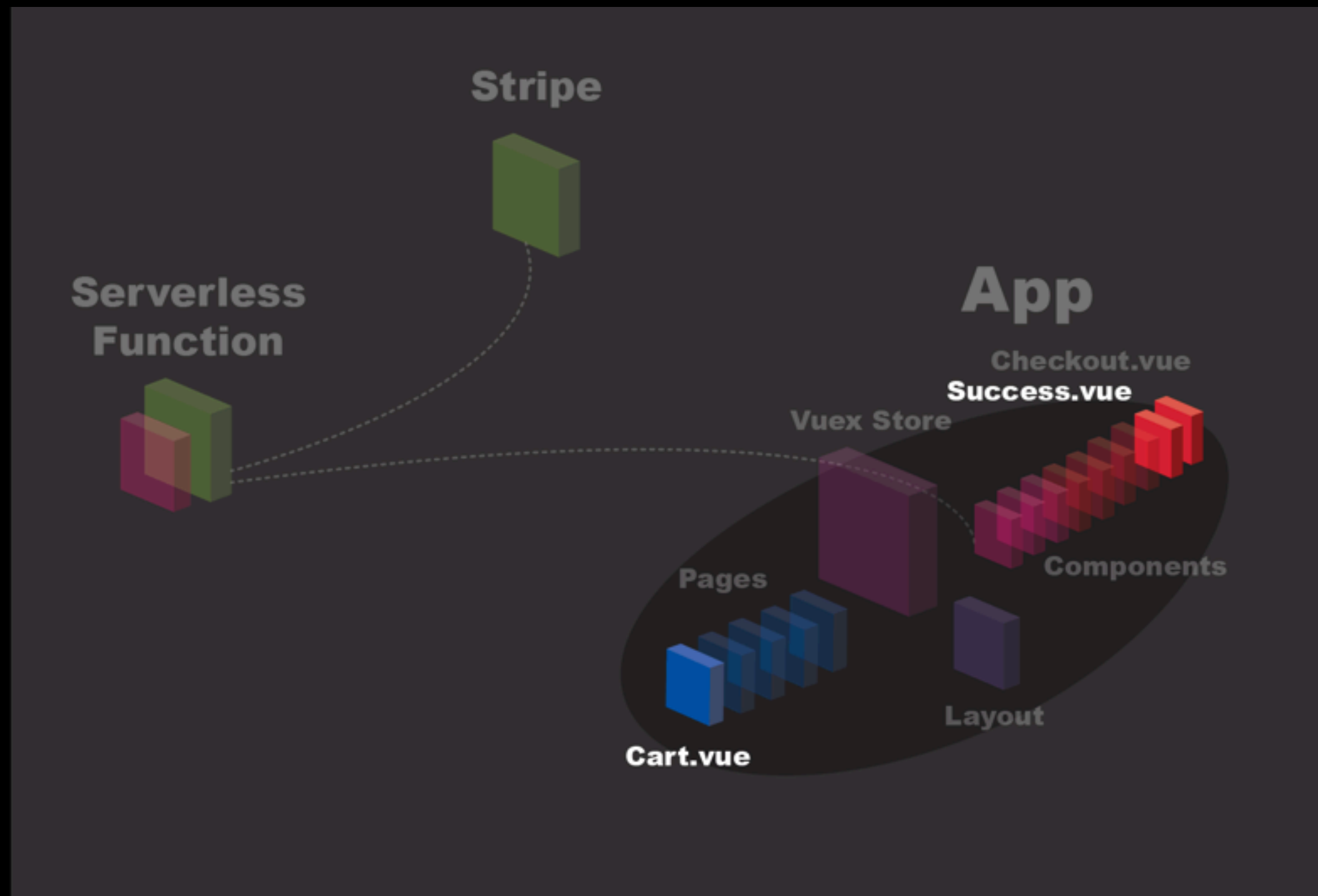
name@example.com

Credit Card

Test using this credit card: 4242 4242 4242 4242, and
enter any 5 digits for the zip code

 4242 04 / 20 242 42424

PAY WITH CREDIT CARD



AppSuccess.vue



```
window.setTimeout(() => this.$emit('restartCart'), 3000);
```

THIS CODEPEN

```
<div v-if="cartTotal > 0">  
  <!--we'll add our checkout here-->  
</div>
```

```
<div v-else-if="cartTotal === 0 && success === false" class="empty">  
  <!--we'll add our empty state here-->  
</div>
```

```
<div v-else>  
  <!--we'll add success here-->  
</div>
```

```
<div v-if="cartTotal > 0">
  <h1>Cart</h1>
  ...
  <app-checkout :total="total" @successSubmit="success = true"></app-checkout>
</div>

<div v-else-if="cartTotal === 0 && success === false" class="empty">
  <h1>Cart</h1>
  <h3>Your cart is empty.</h3>
  <nuxt-link exact to="/"><button>Fill 'er up!</button></nuxt-link>
</div>

<div v-else>
  <app-success @restartCart="success = false"/>
  <h2>Success!</h2>
  <p>Your order has been processed, it will be delivered shortly.</p>
</div>
```



SECOND USE CASE

GOOGLE MAPS API POWERED DATA VIS

LET'S GET VISUAL

Series: Exploring data with Serverless and Vue



Cloud Developer Advocate Speaking

Microsoft Azure

Filter By

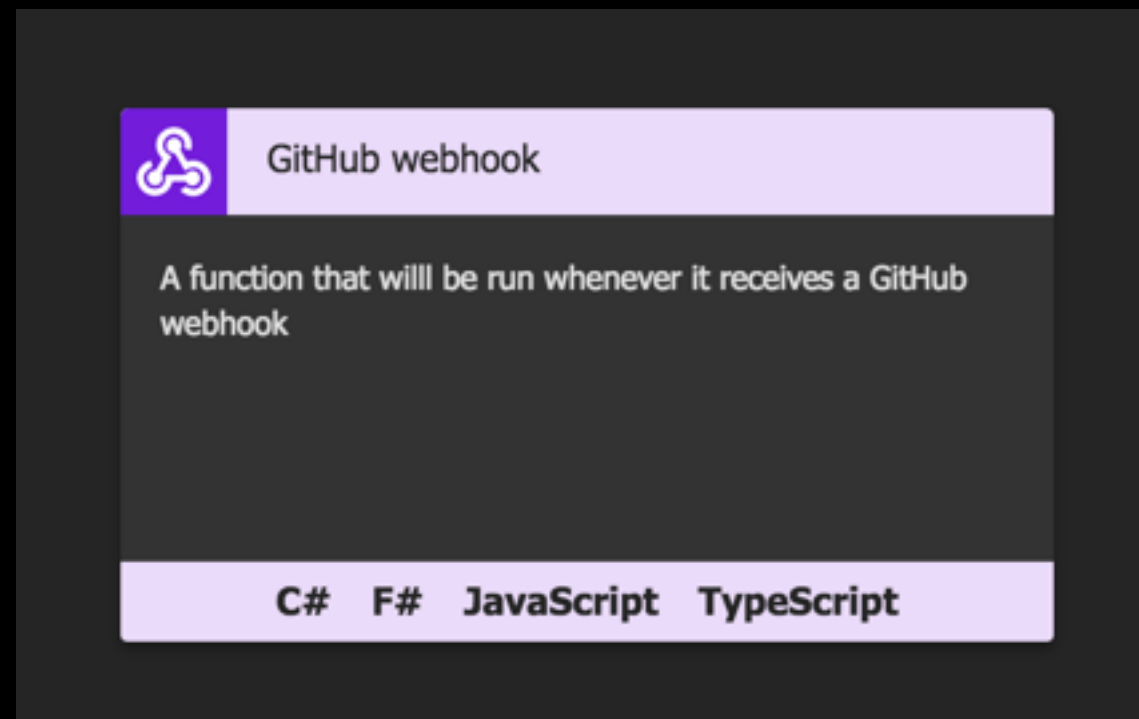
Name	Conference	From	To	Location
Abel Wang	Visual Studio Live	3/13/2017	3/17/2017	Las Vegas
Abel Wang	Tech Summit 2017 Amsterdam	3/20/2017	3/21/2017	Amsterdam
Abel Wang	Tech Summit 2017 Milan	3/20/2017	3/21/2017	Milan
Abel Wang	TechSummit 2017 Seoul	4/26/2017	4/27/2017	Seoul
Abel Wang	Velocity	6/19/2017	6/21/2017	San Jose
Abel Wang	VSLive Redmond	8/15/2017	8/16/2017	Redmond
Abel Wang	Jenkins World	8/29/2017	8/31/2017	San Francisco
Abel Wang	Minneapolis Azure Conference	9/7/2017	9/7/2017	Minneapolis
Abel Wang	Visual Studio Partner Summit	9/20/2017	9/21/2017	Redmond
Abel Wang	MSIgnite 2017	9/24/2017	9/29/2017	Orlando, FL
Abel Wang	Minnesota Developer Conference	10/2/2017	10/2/2017	Minneapolis
Abel Wang	AppConn - Budapest	10/10/2017	10/11/2017	Budapest
Abel Wang	VS Live -Anaheim	10/16/2017	10/17/2017	Anaheim
Abel Wang	AppConn - Houston	11/1/2017	11/2/2017	Houston, TX
Abel Wang	AppConn - Thailand	11/9/2017	11/10/2017	Bangkok
Abel Wang	MS Connect	11/15/2017	11/17/2017	NYC
Abel Wang	Tech Summit 2017 Amsterdam	3/20/2017	3/21/2017	Amsterdam
Abel Wang	Tech Summit 2017 Milan	3/20/2017	3/21/2017	Milan
Abel Wang	TechSummit 2017 Seoul	4/26/2017	4/27/2017	Seoul
Abel Wang	MSIgnite 2017	9/24/2017	9/29/2017	Orlando, FL

DEMO AND REPO W/ OSS CODE

What we start with looks like this

```
[
  {
    "Name": "Simona Cotin",
    "Conference": "DLD 2017 Tel Aviv Israel",
    "From": "9/4/2017",
    "To": "9/7/2017",
    "Location": "Tel Aviv",
    "Link": ""
  },
  ...
]
```

MAKE THE FUNCTION



We're going to retrieve the geo-information for each item in the original data

```
getGeo(makeIterator(content), (updatedContent, err) => {  
  if (!err) {  
    // we need to base64 encode the JSON to embed it into the PUT (dear god, why)  
    let updatedContentB64 = new Buffer(  
      JSON.stringify(updatedContent, null, 2)  
    ).toString('base64');  
    let pushData = {  
      path: GH_FILE,  
      message: 'Looked up locations, beep boop.',  
      content: updatedContentB64,  
      sha: data.sha  
    };  
    ...  
  };  
};
```

```
function getGeo(itr, cb) {  
  let curr = itr.next();  
  if (curr.done) {  
    // All done processing- pass the (now-populated) entries to the next callback  
    cb(curr.data);  
    return;  
  }  
  
  let location = curr.value.Location;
```

index.js

Save

Run

</> Get function URL </> Get Github secret

View files

Test



```
1 const https = require('https');
2
3 let gContext;
4
5 // Github configuration is read from process.env
6 let GH_USER = process.env.GH_USER;
7 let GH_KEY = process.env.GH_KEY;
8 let GH_REPO = process.env.GH_REPO;
9 let GH_FILE = process.env.GH_FILE;
10
11 // Per-run cache of geographical locations, to avoid hitting Maps API unnecessarily
12 let GEO_CACHE = {};
13
14 // ENTRY POINT - retrieve data file from Github and decode/parse it before passing
15 // into getGeo (which iterates over each entry)
16 module.exports = function (context, data) {
17   // Make the context available globally
18   gContext = context;
19
20   // Do the work!
21   getGithubJson(githubFilename(),
22     (data, err) => {
23       if (!err) {
24         // No error; base64 decode and JSON parse the data from the Github response
```

HTTP method

POST

Query

clientId

default

+ Add parameter

Headers

There are no headers

+ Add header

Request body

```
1 [
2   "comment": {
3     "body": "This is a comment on a Si
4   }
5 ]
6
```

Logs

Pause Clear Copy logs Expand

```
2017-10-06T05:23:22 Welcome, you are now connected to log-streaming service.
```

Output

Run

The function updates our cda-data.json...

```
[
  {
    "Name": "Simona Cotin",
    "Conference": "DLD 2017 Tel Aviv Israel",
    "From": "9/4/2017",
    "To": "9/7/2017",
    "Location": "Tel Aviv",
    "Link": ""
  },
  ...
]
```

...which we pull into our Vuex store

The function updates our cda-data.json...

```
[
  {
    "Name": "Simona Cotin",
    "Conference": "DLD 2017 Tel Aviv Israel",
    "From": "9/4/2017",
    "To": "9/7/2017",
    "Location": "Tel Aviv",
    "Link": "",
    "Latitude": 32.0852999,
    "Longitude": 34.78176759999999
  },
  ...
]
```

...which we pull into our Vuex store

```
computed: {  
  speakerData() {  
    return this.$store.state.speakerData;  
  },  
  filteredData() {  
    const x = this.selectedFilter,  
          filter = new RegExp(this.filteredText, 'i')  
    return this.speakerData.filter(el => {  
      if (el[x] !== undefined) { return el[x].match(filter) }  
      else return true;  
    })  
  }  
}
```

```
<table class="scroll">
  <thead>
    <tr>
      <th v-for="key in columns">
        {{ key }}
      </th>
    </tr>
  </thead>
  <tbody>
    <tr v-for="(post, i) in filteredData">
      <td v-for="entry in columns">
        <a :href="post.Link" target="_blank">
          {{ post[entry] }}
        </a>
      </td>
    </tr>
  </tbody>
</table>
```

Cloud Developer Advocate Speaking

Microsoft Azure

Filter By

Name	Conference	From	To	Location
Abel Wang	Visual Studio Live	3/13/2017	3/17/2017	Las Vegas
Abel Wang	Tech Summit 2017 Amsterdam	3/20/2017	3/21/2017	Amsterdam
Abel Wang	Tech Summit 2017 Milan	3/20/2017	3/21/2017	Milan
Abel Wang	TechSummit 2017 Seoul	4/26/2017	4/27/2017	Seoul
Abel Wang	Velocity	6/19/2017	6/21/2017	San Jose
Abel Wang	VSLive Redmond	8/15/2017	8/16/2017	Redmond
Abel Wang	Jenkins World	8/29/2017	8/31/2017	San Francisco
Abel Wang	Minneapolis Azure Conference	9/7/2017	9/7/2017	Minneapolis
Abel Wang	Visual Studio Partner Summit	9/20/2017	9/21/2017	Redmond
Abel Wang	MSIgnite 2017	9/24/2017	9/29/2017	Orlando, FL
Abel Wang	Minnesota Developer Conference	10/2/2017	10/2/2017	Minneapolis

```
this.speakerData.forEach(function(index) {  
    ...  
  
    if (val in endUnit) {  
        // if we already have this location (stored together as key) let's  
        increment it.  
        // this way of writing it is more performant than functional,  
        if you don't believe me, let's arm wrestle.  
        if (key in endUnit[val]) {  
            endUnit[val][key][2] += magBase;  
        } else {  
            endUnit[val][key] = [lat, long, magBase];  
        }  
    } else {  
        let y = {};  
        y[key] = [lat, long, magBase];  
        endUnit[val] = y;  
    }  
})
```

THREE.JS

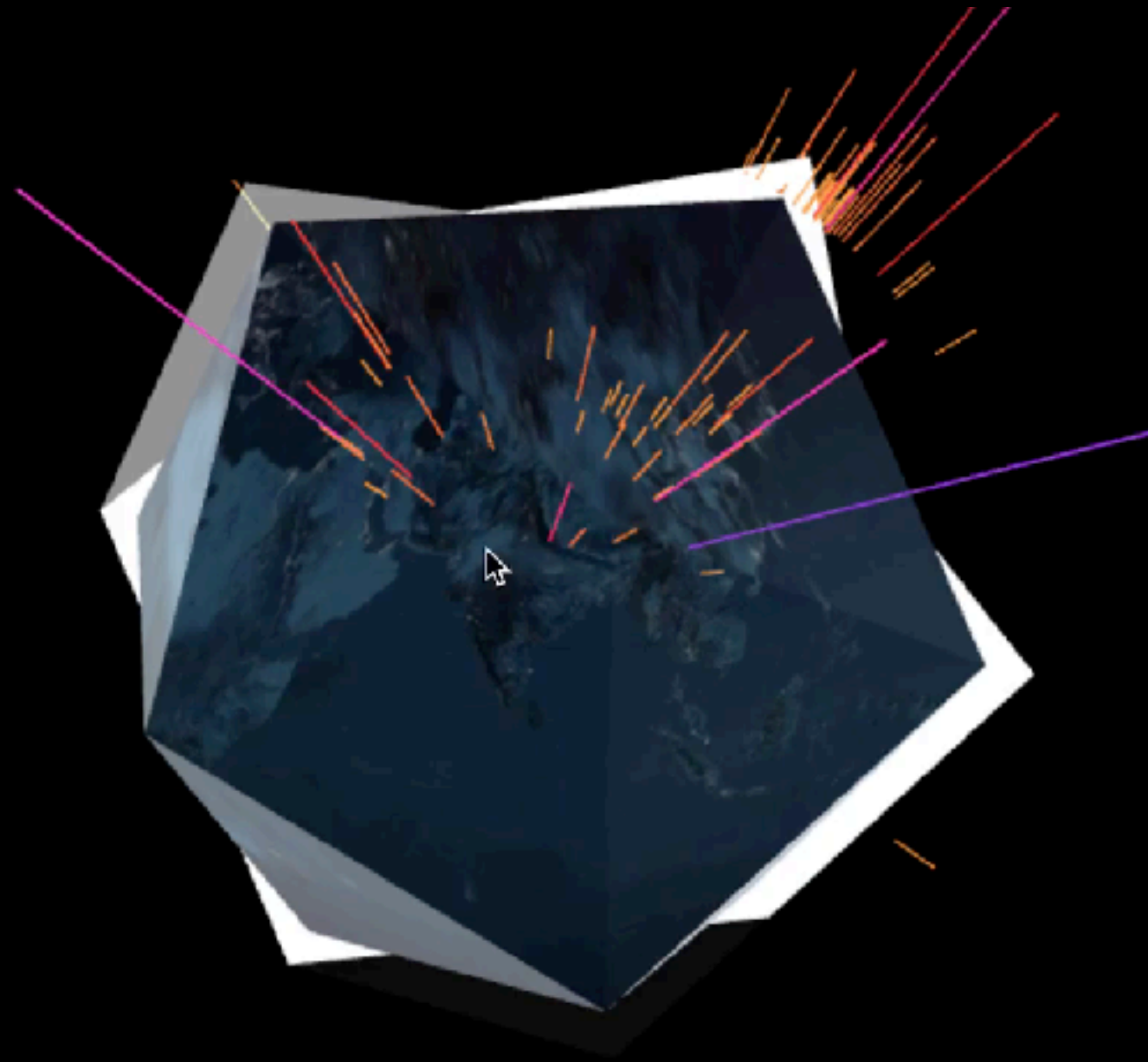
Single element

```
<div ref="container"></div>
```

Call on mounted

```
mounted() {  
  //we have to load the texture when it's mounted and pass it in  
  let earthmap = THREE.ImageUtils.loadTexture('/world7.jpg');  
  this.initGlobe(earthmap);  
}
```

```
//from  
const geometry = new THREE.SphereGeometry(200, 40, 30);  
//to  
const geometry = new THREE.IcosahedronGeometry(200, 0);
```



VUE.JS + SERVERLESS



THANK YOU!

@sarah_edo



These slides:

<http://bit.ly/2DZfxsw>