Denied!

Securing your Applications with Better User Authorization

RVA.JS, March 2022

Brian Childress

Application Architect

I build things @ 

#werehiring

brianchildress.co

Agenda

- What is Authorization?
- Types of Authorization
- Ways to Authorize Users/Apps

Terminology

Authentication vs. Authorization

AuthN
vs.
AuthZ

- authentication *who you are?*

- authorization *what you can do?*

Photo by Bill Anastas on Unsplash

Authentication

Photo by Helena Lopes from Pexels

Authorization

The old way

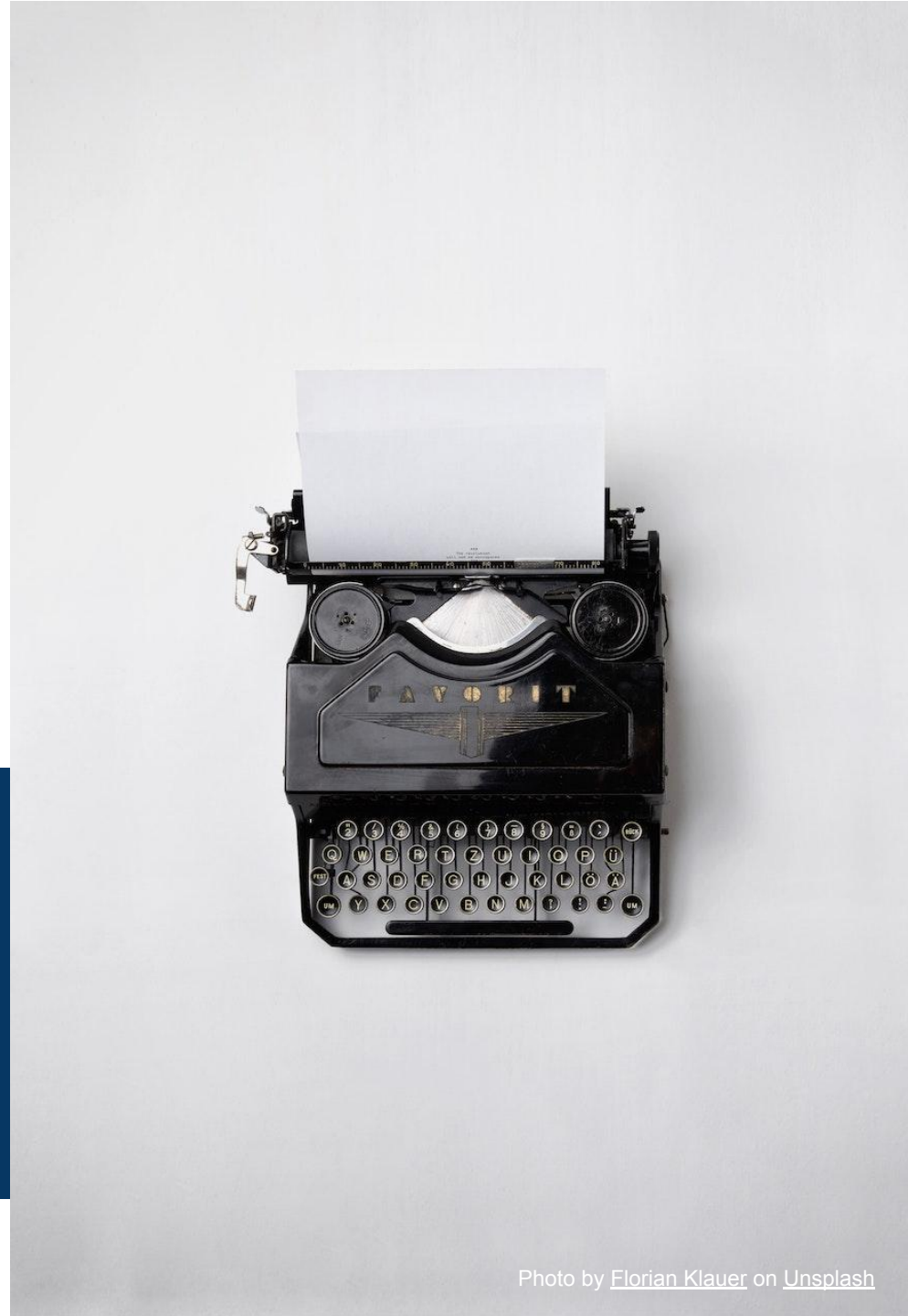
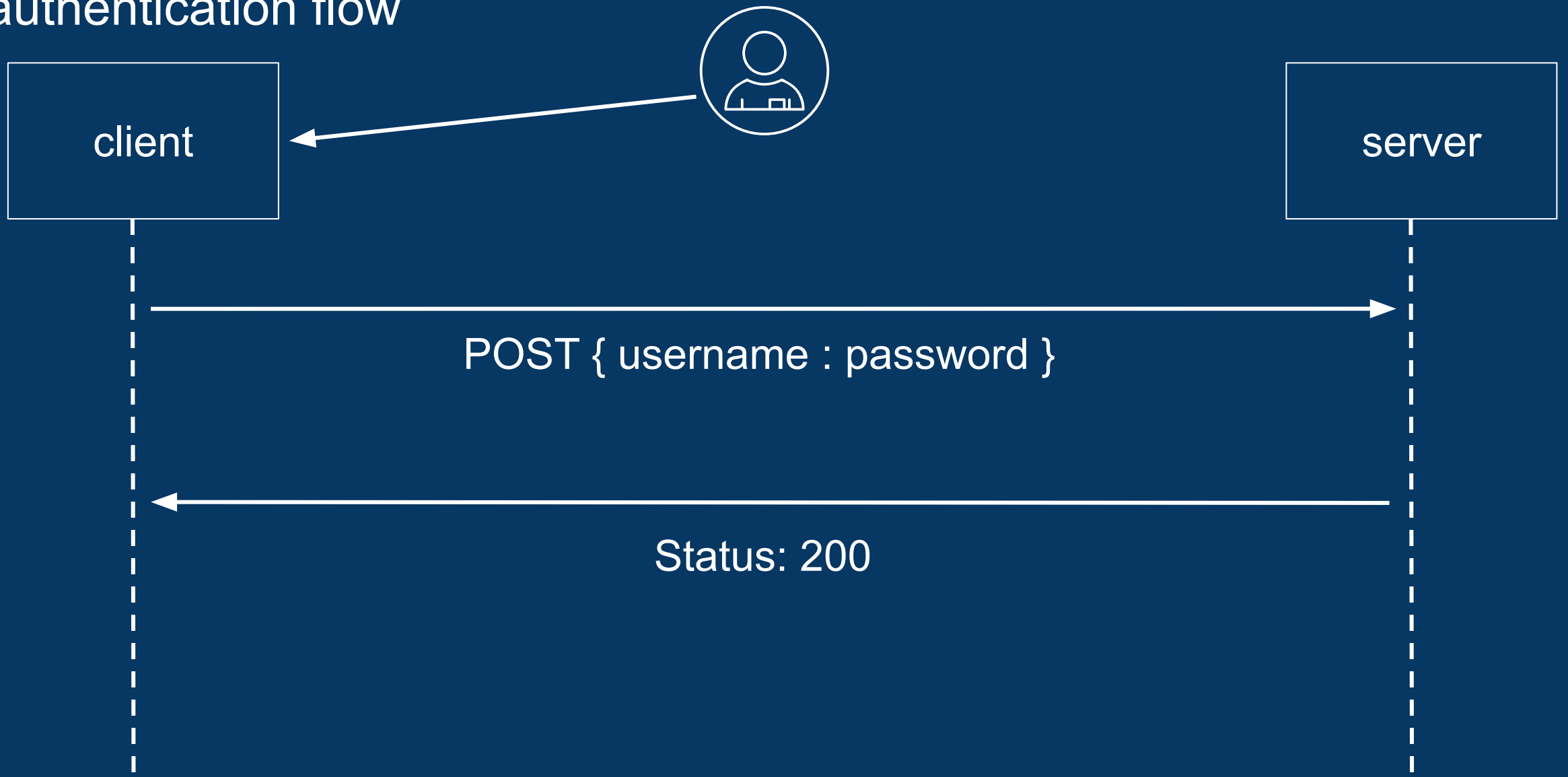


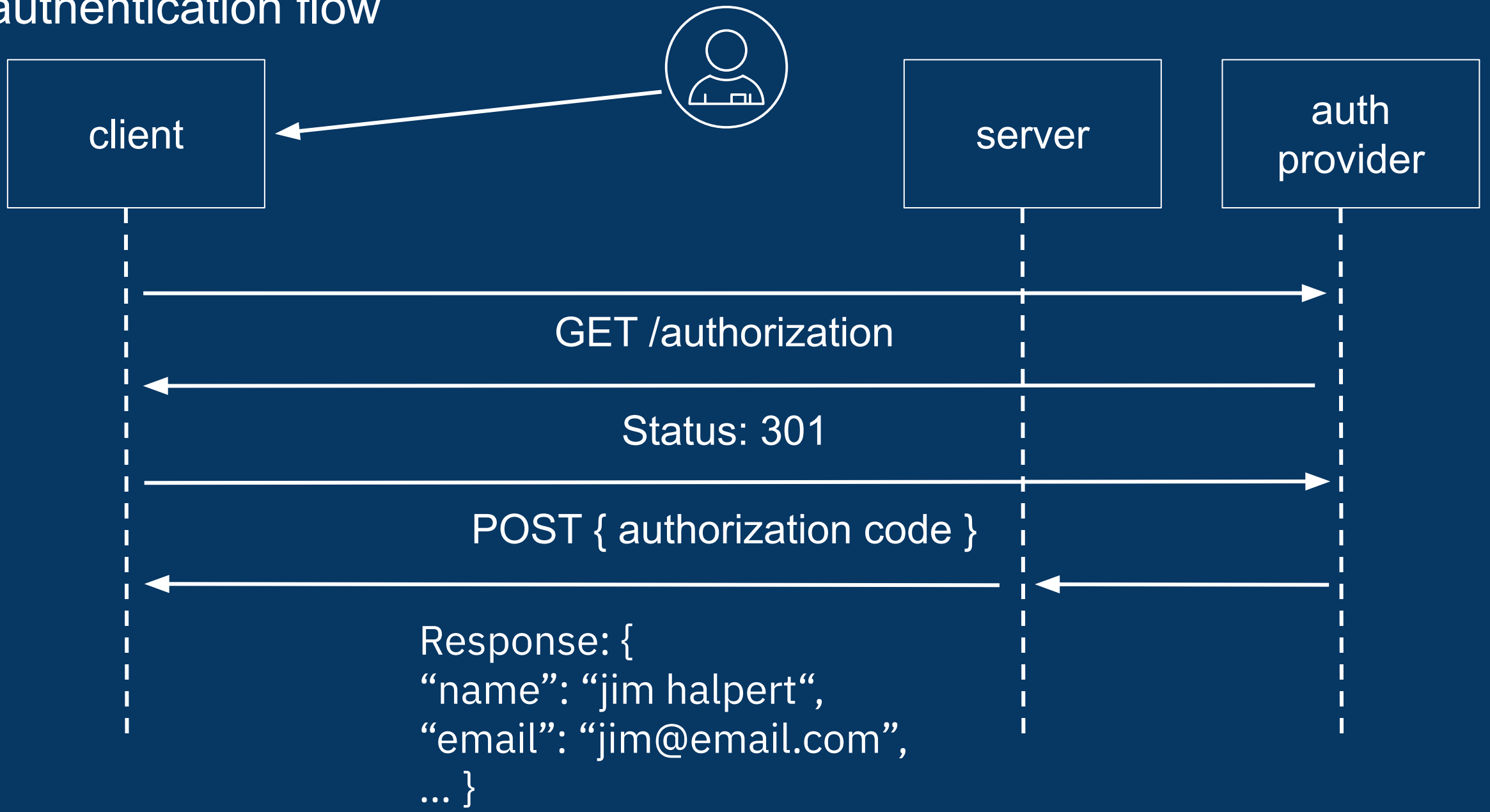
Photo by Florian Klauer on Unsplash

authentication flow



The new way

authentication flow



- authentication *who you are?*

- authorization *what you can do?*

- authorization *is hard!*

{ JSON Web Token }

*JSON Web Token (JWT) is an open standard (RFC 7519) that defines a **compact and self-contained** way for securely transmitting information between parties as a JSON object.*

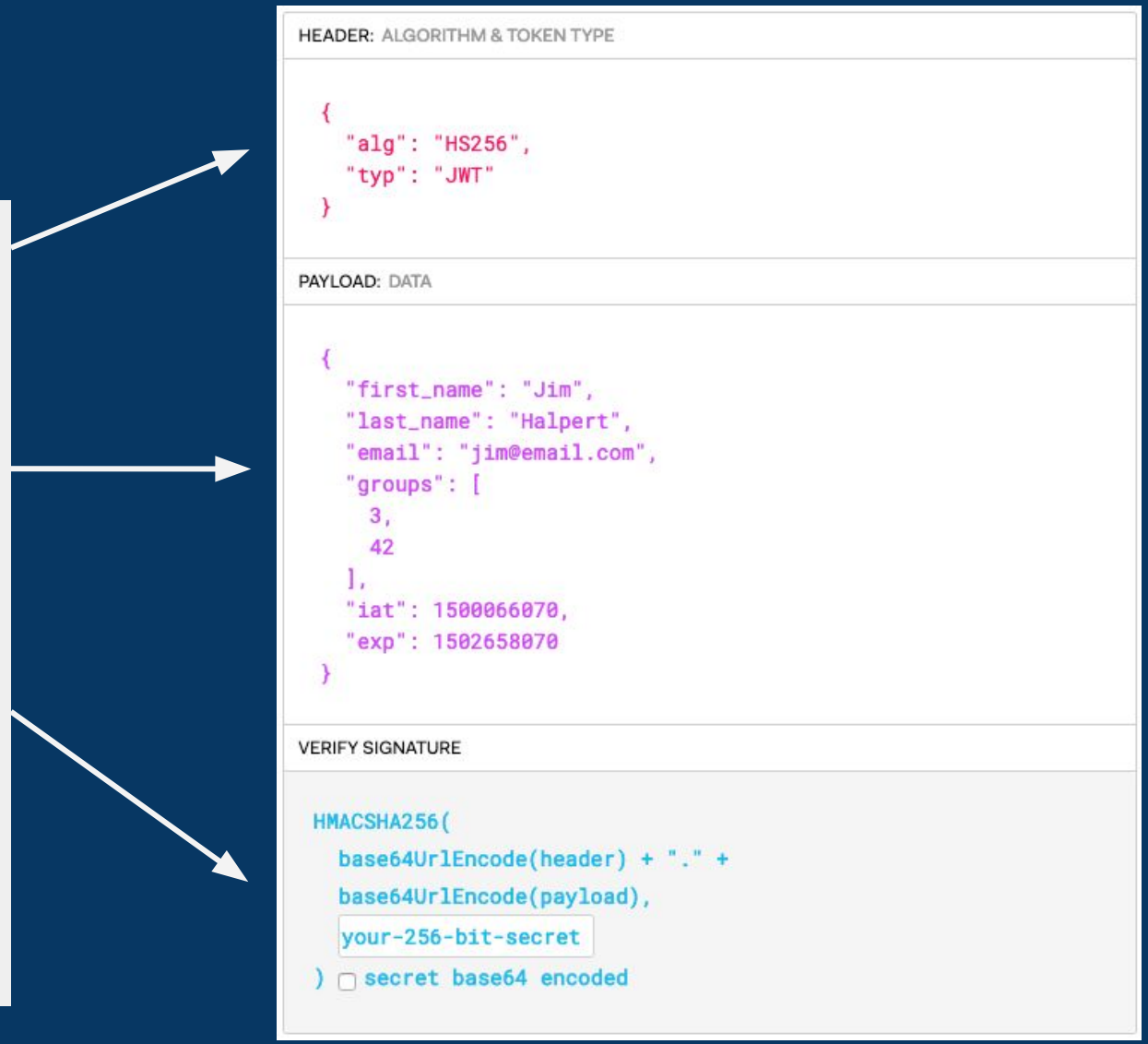
https://jwt.io/

JSON Web Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.

eyJmaXJzdF9uYW1lIjoiSmltIiwibGFzdF9uYW1lIjoiSGFscGVydCIsImVtYWlsIjoiamltQGVtYWlsLmNvbSIsImdyb3VwcyI6WzMsNDJdLCJpYXQiOjE1MDAwNjYwNzAsImV4cCI6MTUwMjY1ODA3MH0.

u0FBAULAuRxsTbHjubF3XmALKolez4vMEWOCgI3pFxk



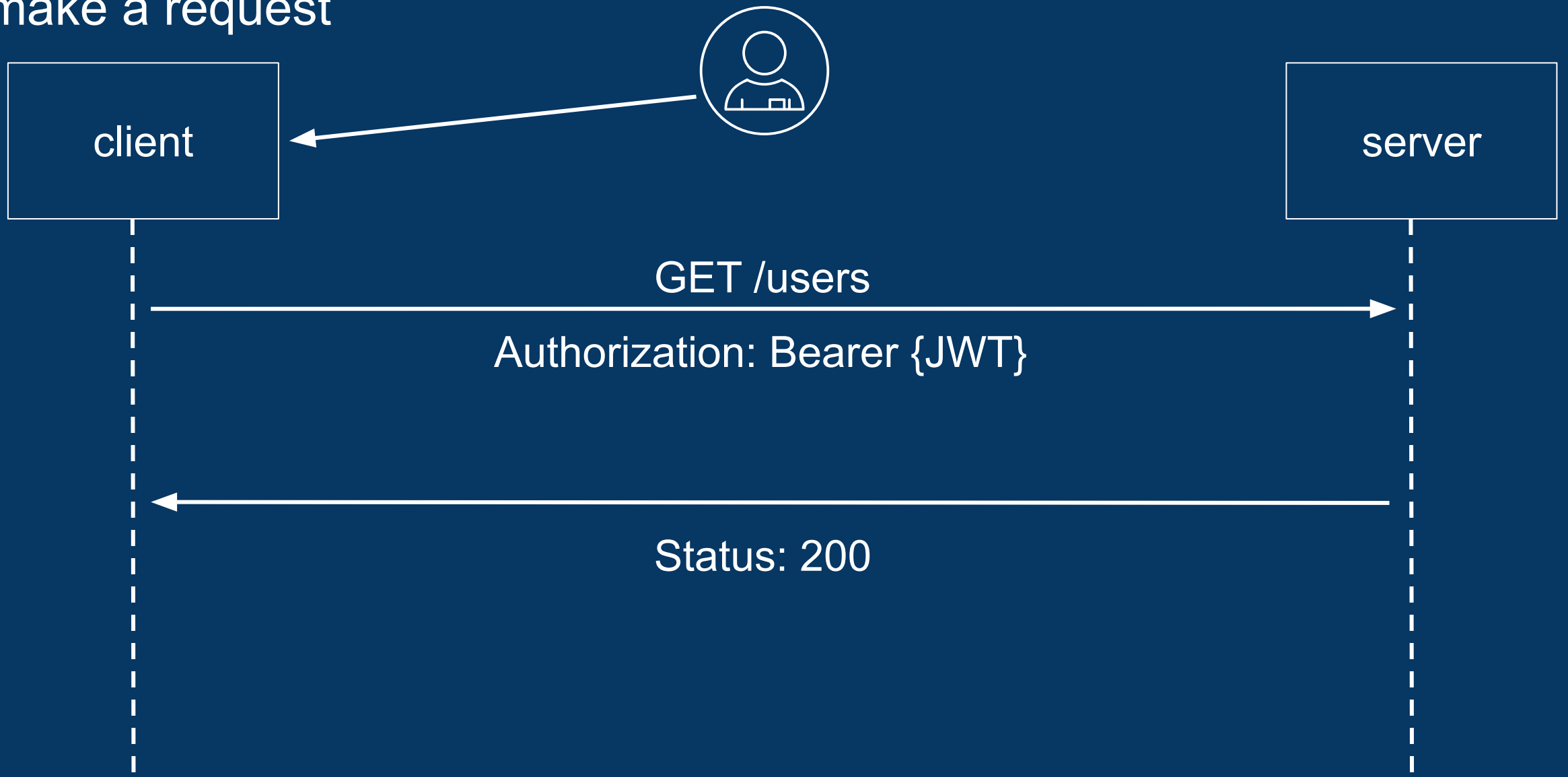
JSON Web Token

```
{
  "sub":  "1234567890",
  "admin":  true,
  "name":  "Jim Halpert",
  "email":  "jim@email.com",
  "iat":  1516239022,
  "exp":  1516240022,
  "roles":  [
            "manager" ,
            "superAdmin"
   ]
}
```

JSON Web Token


```
{
  "sub":  "1234567890",
  "admin":  true,
  "name":  "Jim Halpert",
  "email":  "jim@email.com",
  "iat":  1516239022,
  "exp":  1516240022,
  "groups":  [
          6,
          42,
          56
   ]
}
```

make a request



types of authorization

role-based (RBAC)

role	canRead	canEdit	canDelete
user	🔥		
editor	🔥	🔥	
admin	🔥	🔥	🔥

types of authorization

claims-based

user	canReadProducts	canEditProducts	canDeleteProducts
bob	🚀		
sue	🚀	🚀	
morgan	🚀	🚀	🚀

types of authorization

attribute-based

IF *editor* THEN *can edit products*

types of authorization

attribute-based

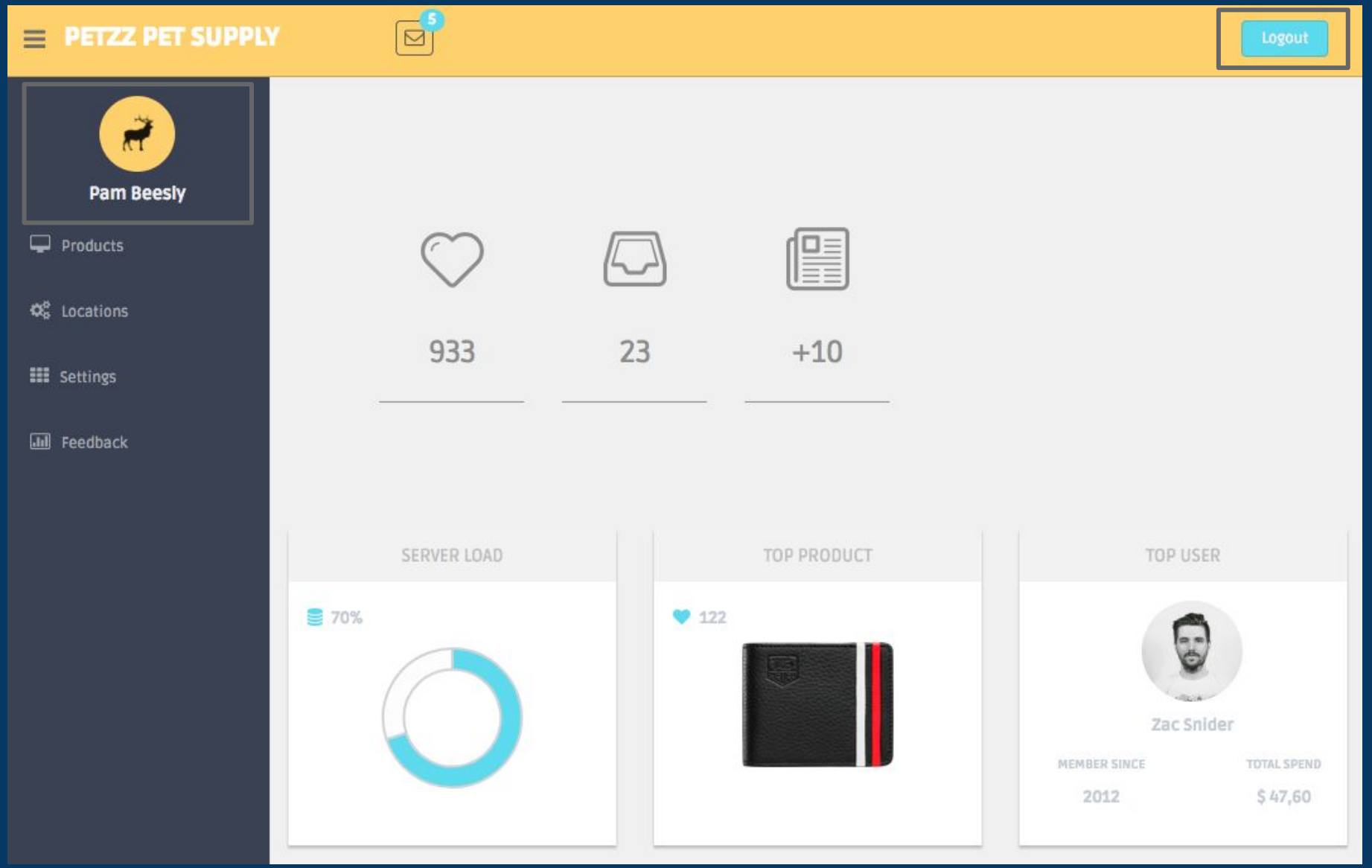
IF *before 9am* THEN *can read-only*

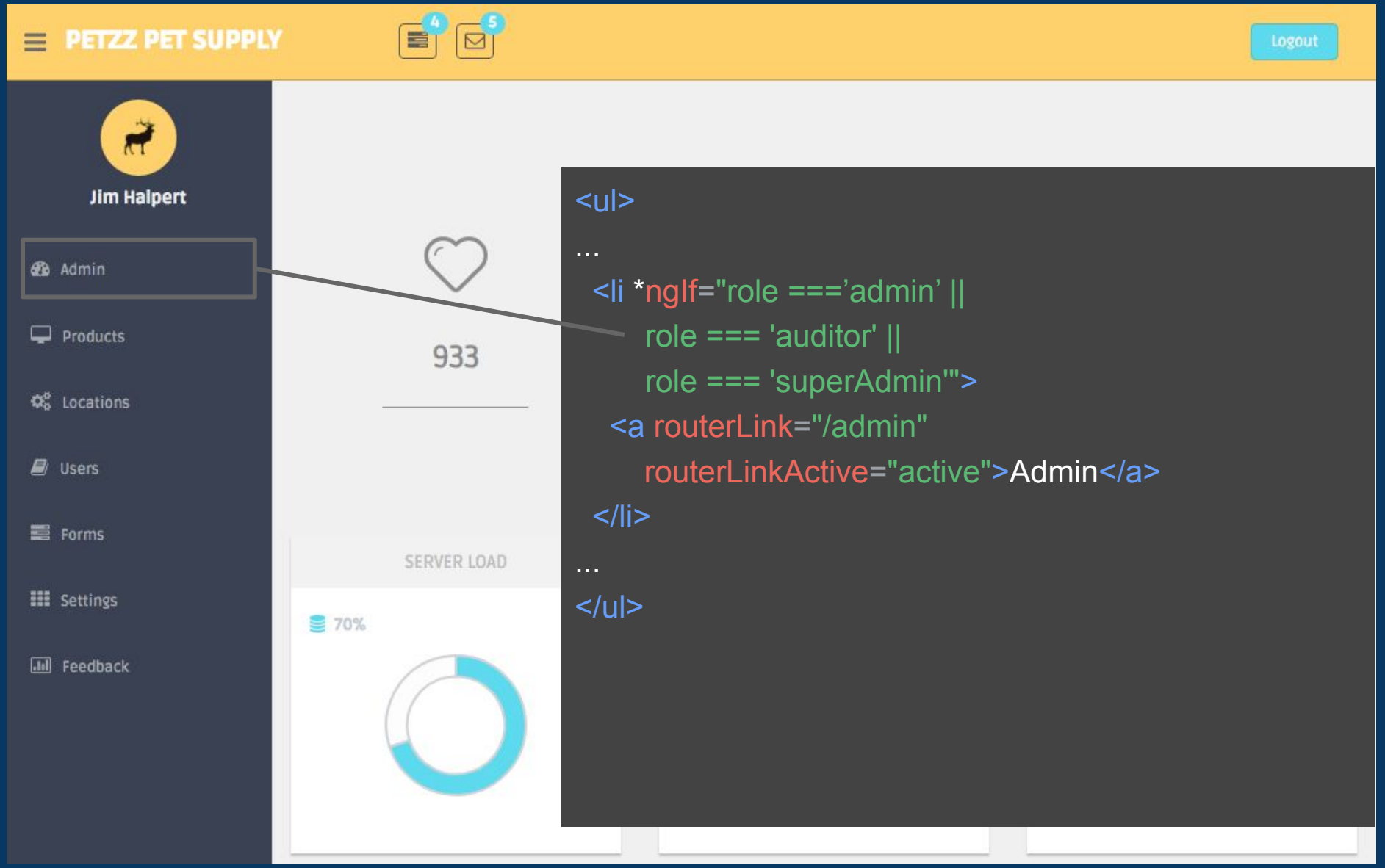
- who? what?

how do we authorize?

routes

components


PETZZ PET SUPPLY
5
Logout
Pam Beesly
Products
Locations
Settings
Feedback
933
23
+10
SERVER LOAD
70%
TOP PRODUCT
122
TOP USER
Zac Snider
MEMBER SINCE
2012
TOTAL SPEND
$ 47,60

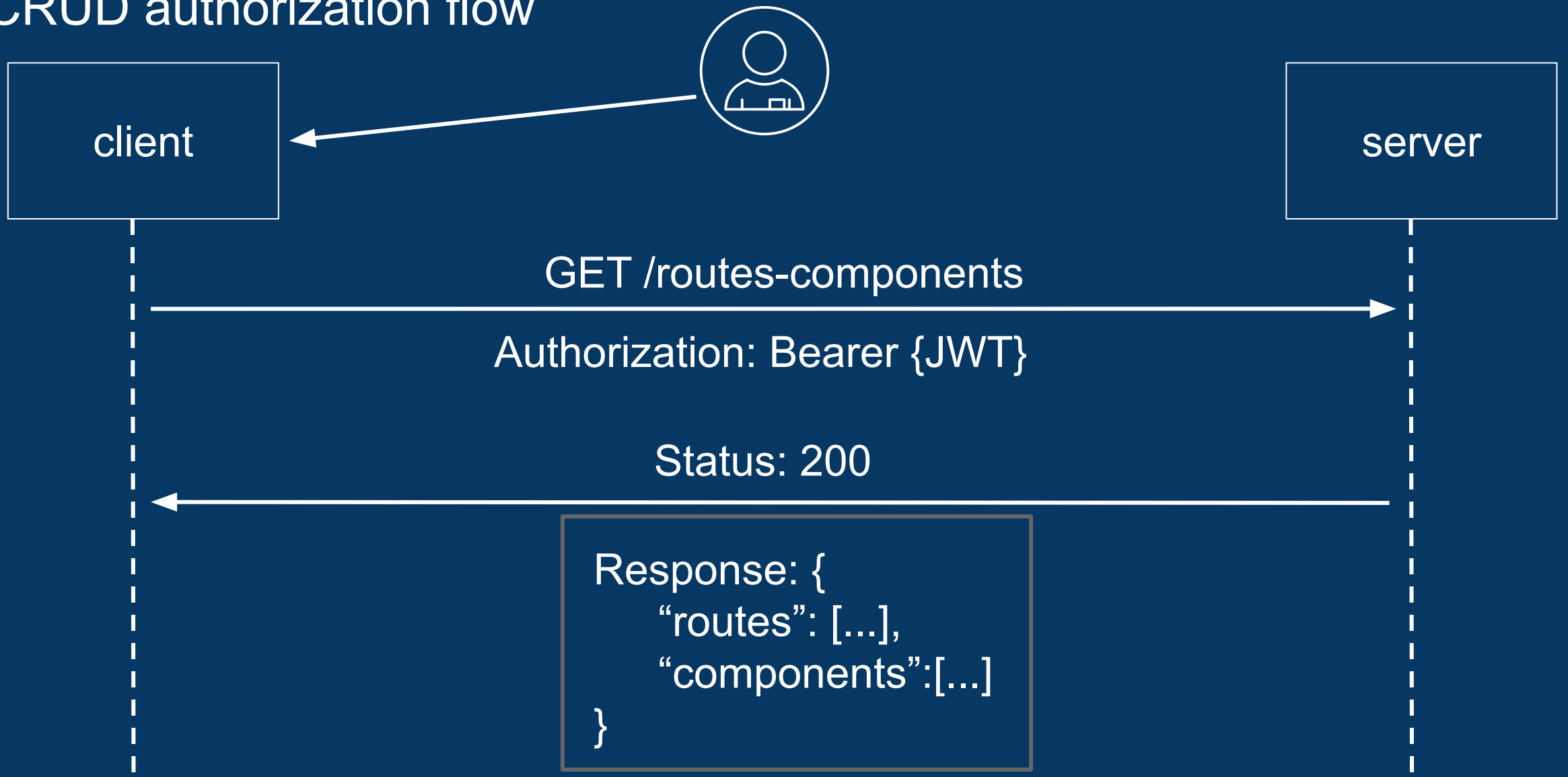

PETZZ PET SUPPLY
4
5
Logout
Jim Halpert
Admin
Products
Locations
Users
Forms
Settings
Feedback
933
SERVER LOAD
70%

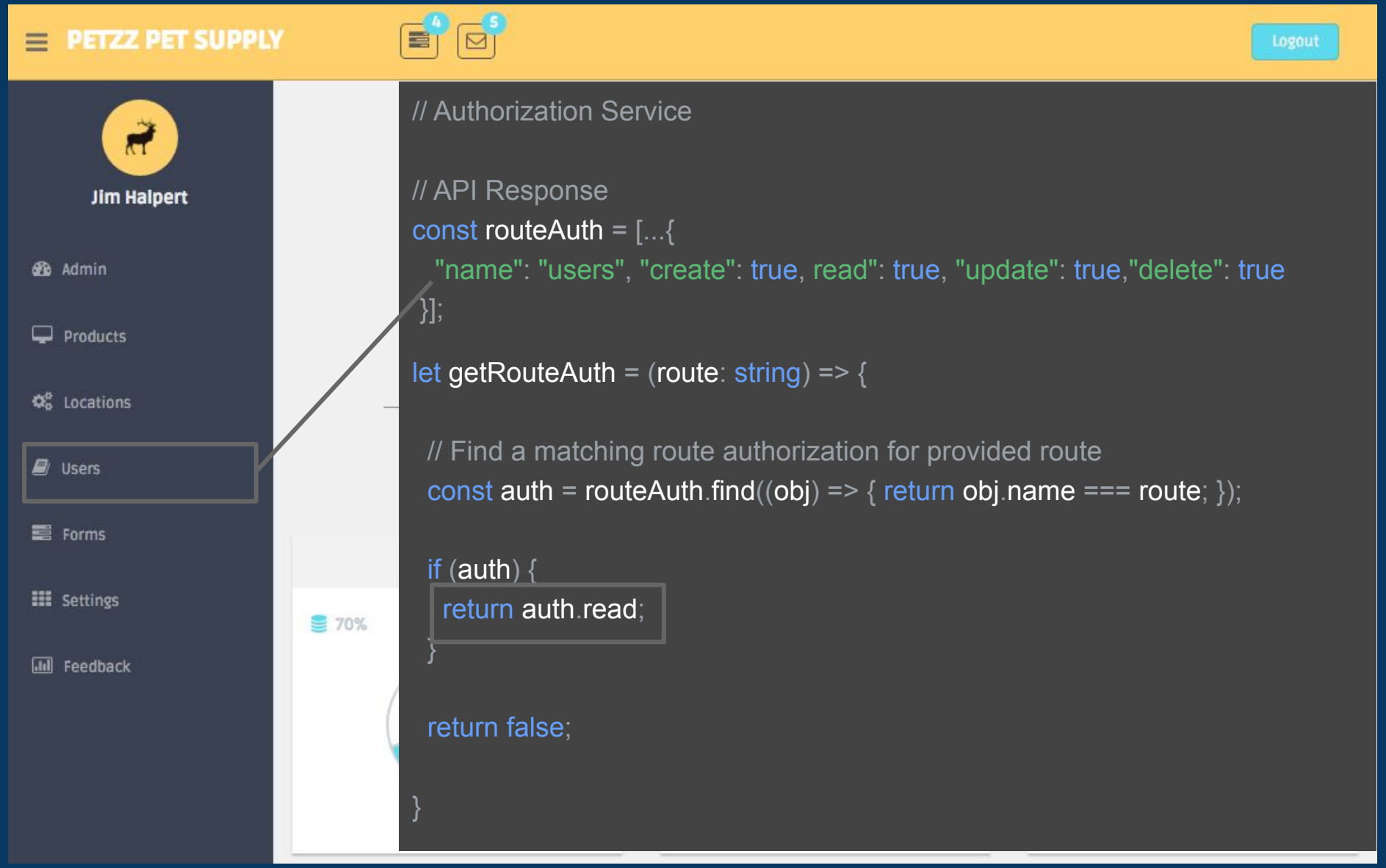
...
<li *ngIf="role ===’admin’ ||
role === 'auditor' ||
role === 'superAdmin'">
<a routerLink="/admin"
routerLinkActive="active">Admin

...

C	R	U	D
r	e	p	e
e	a	d	l
a	d	a	e
t		t	t
e		e	e

CRUD authorization flow




PETZZ PET SUPPLY
4
5
Logout
Jim Halpert
Admin
Products
Locations
Users
Forms
Settings
Feedback
70%
// Authorization Service
// API Response
const routeAuth = [...{
"name": "users", "create": true, read": true, "update": true,"delete": true
}];
let getRouteAuth = (route: string) => {
// Find a matching route authorization for provided route
const auth = routeAuth.find((obj) => { return obj.name === route; });
if (auth) {
return auth.read;
}
return false;
}

User

First Name

Jim

Last Name

Halpert

Address

707 West Main Street

Address 2

City

Denver

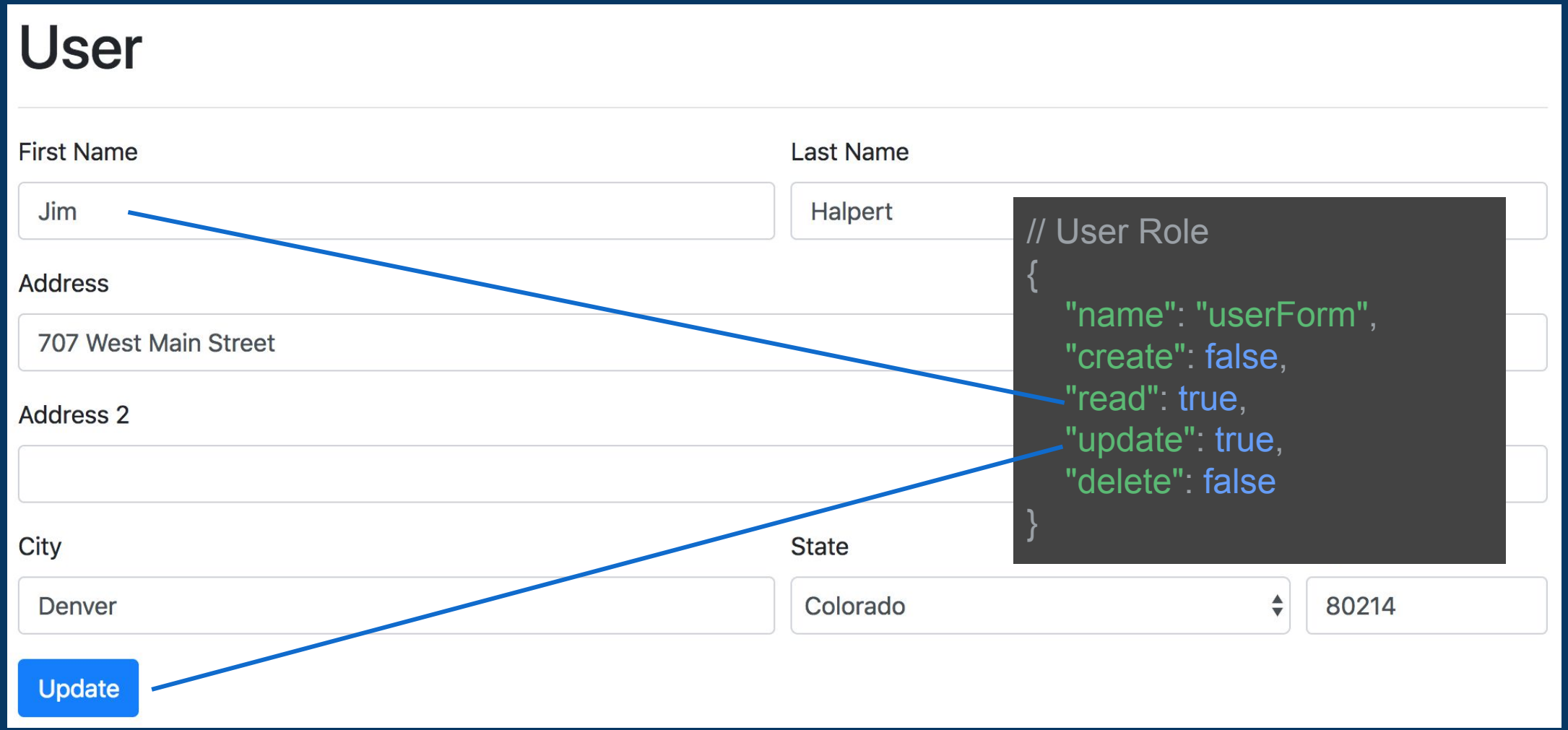
State

Colorado

Zip

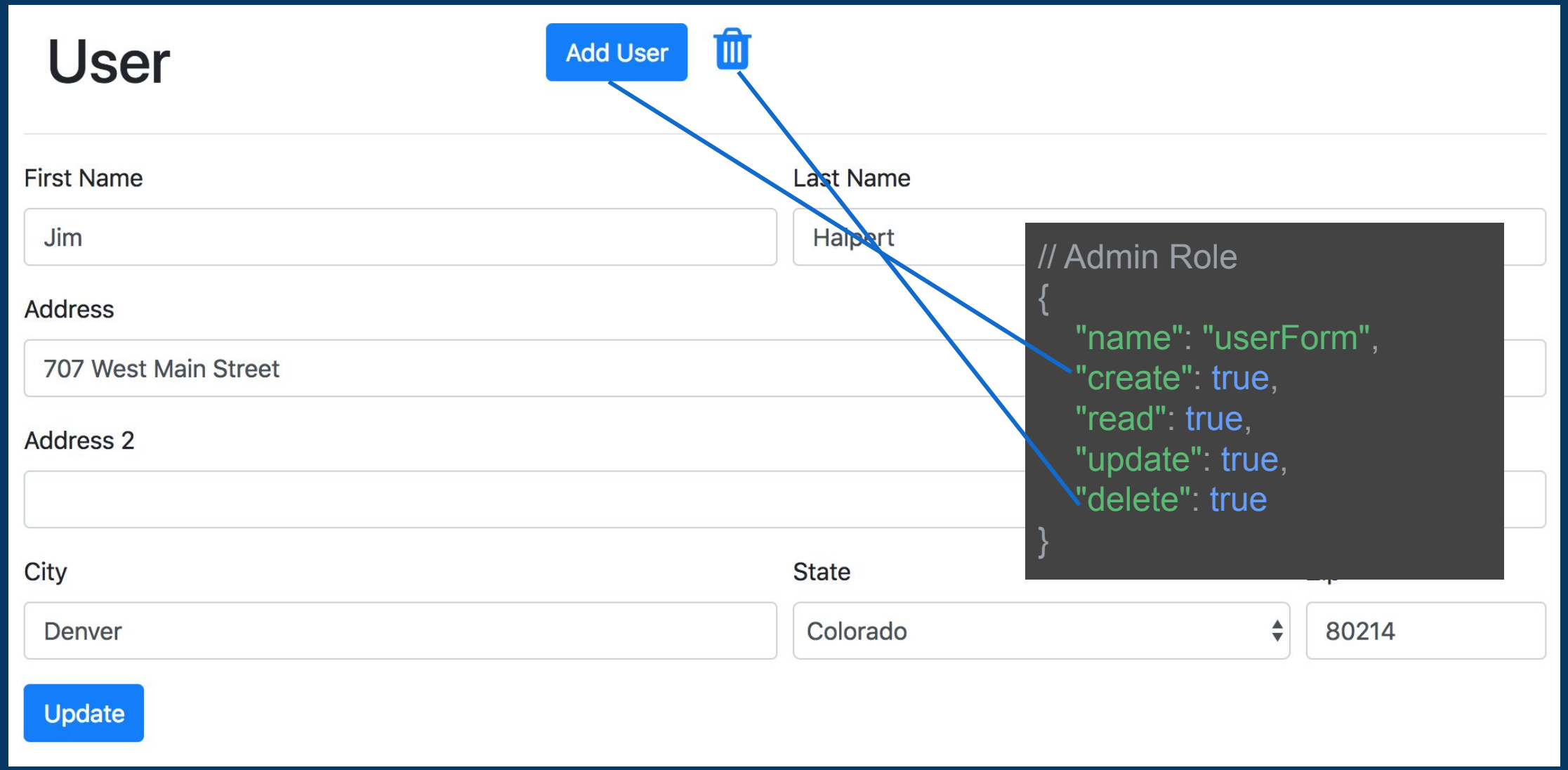
80214

Update


User
First Name
Jim
Last Name
Halpert
Address
707 West Main Street
Address 2
City
Denver
State
Colorado
80214
Update
// User Role
{
"name": "userForm",
"create": false,
"read": true,
"update": true,
"delete": false
}

```
<!-- User Component -->

<form *ngIf="auth.read">
  <div class="form-row">
    <div class="form-group">
      <label for="firstName">First Name</label>
      <input type="text" id="firstName" placeholder="First Name"
       [disabled]="!auth.create || !auth.update" *ngIf="auth.read">
    </div>
...
  <button type="submit" *ngIf="auth.update">Update</button>
</form>
<div *ngIf="!auth.read">
    You are not authorized to view this content. Please contact your administrator.
</div>
```


User
Add User
First Name
Jim
Last Name
Address
707 West Main Street
Address 2
City
Denver
State
Colorado
80214
Update
// Admin Role
{
"name": "userForm",
"create": true,
"read": true,
"update": true,
"delete": true
}

User

First Name

Jim

Last Name

Halpert

Address

707 West Main Street

Address 2

City

Denver

State

Colorado

80214

```
// Auditor Role
{
    "name": "userForm",
    "create": false,
    "read": true,
    "update": false,
    "delete": false
}
```

additional uses for **CRUD** - based authorization

- system maintenance
- feature rollout
- testing

wrapping up

- No sensitive data, don’t store what you don’t need
- Security should ALWAYS be handled by the server
- Security is important!
- Authorization is hard!

- questions?

- thank you!

brian childress
brian@brianchildress.co

- resources
  - OSO Authorization Academy
  - Auth0 Blog
  - JWT.io