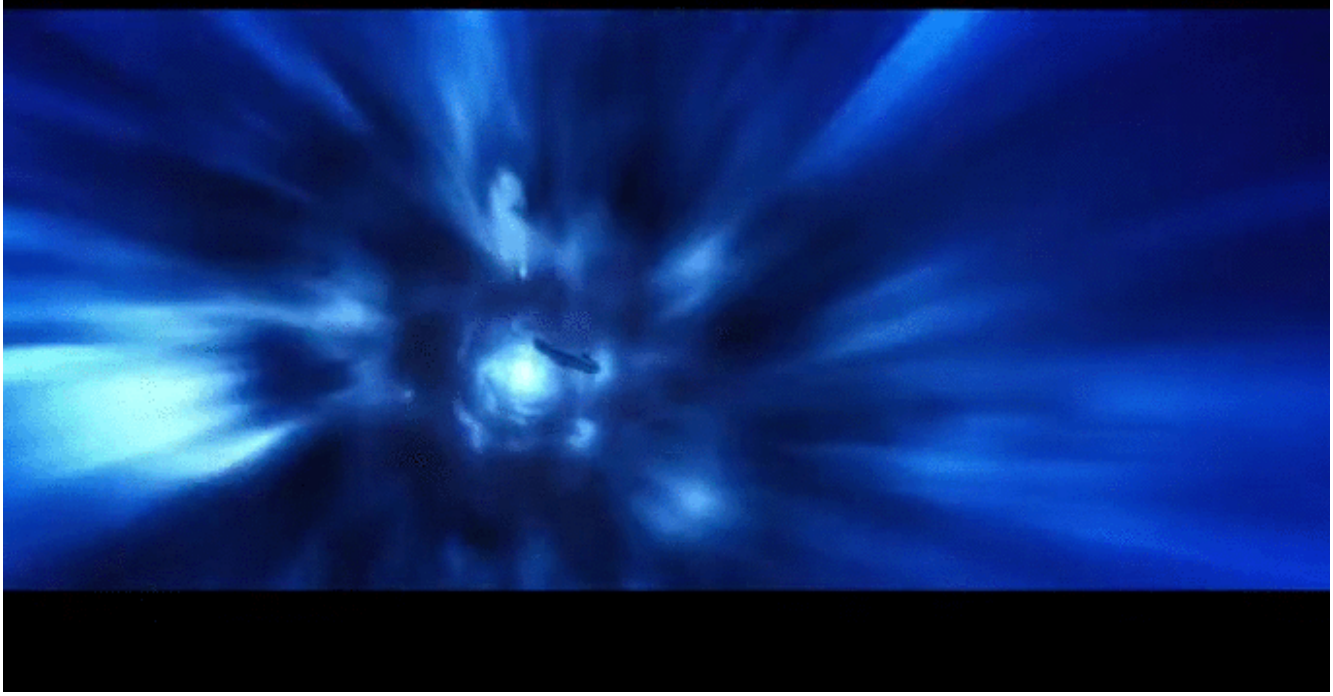
SCALING MY FIRST ENTERPRISE REACT APP!

Jennifer Van

React Rally 2017

@sugargreenbean minh.j.van@gmail.com



ABOUT ME

Jennifer Van of Capital One, First of Her Name, the Unburnt, Queen of the Machines Learning, Khaleesi of Javascript, Breaker of (Bit)Chains, and Mother of Dragons

NOW, LET'S UNPACK THIS TITLE: SCALING ENTERPRISE REACT APP

What does "enterprise" mean?


YEAH, I program in
JAVA
Just
JAVA
help me please I've been stuck in this enterprise dev job for the past 5 years and I'm slowly deteriorating
this isn't a meme it's a legitimate call for help


help me please I've been stuck in this enterprise dev job for the past 5 years and I'm slowly deteriorating
this isn't a meme it's a legitimate call for help

Help me. We're talking about enterprise.

WHAT ARE THE HALLMARKS OF ENTERPRISE PRODUCTS?

Persist

I'M GONNA LIVE FOREVER

WHAT ARE THE HALLMARKS OF ENTERPRISE PRODUCTS?

Persist

Growth

IT'S ONLY GONNA GET BIGGER

WHAT ARE THE HALLMARKS OF ENTERPRISE PRODUCTS?

Persist

Growth

Tested

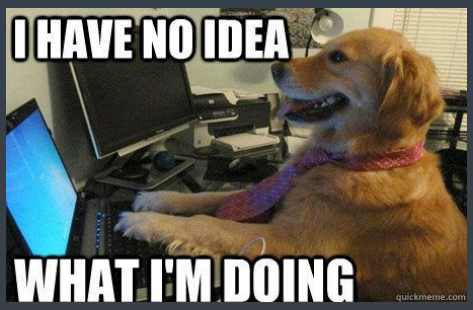
YA GOTTA TEST

LET'S CONTINUE UNPACKING: SCALING ENTERPRISE REACT APP

What does it mean to scale?

SCALING INCORRECTLY IS A DUMPSTER FIRE

WHAT WAS OUR USE CASE?


I HAVE NO IDEA
WHAT I'M DOING
quickmeme.com

AS YOUR PRODUCT GROWS, HOW DO YOU KEEP REACT FAST?

1. Define presentation vs. container components

```
2.   var [name, dateLastChanged,
address] = [
3.     props.name,
4.     props.dateLastChanged,
5.     props.address
6.   ]
7.   return <div>{ name } lives at {
address } as of { dateLastChanged }.
</div>;
8. }
9.
```

Only concerned with the way things look

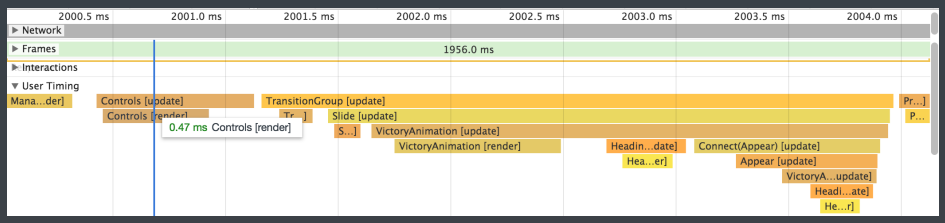
Concerned with how things work

AS YOUR PRODUCT GROWS, HOW DO YOU KEEP REACT FAST?

1. Define presentation vs. container components
2. Utilize performance timeline to determine bottlenecks

HOW TO UTILIZE PERFORMANCE TIMELINE TO DETERMINE BOTTLENECKS

1. Append ?react_perf
2. Open Chrome DevTools Performance tab and press record (don't do this more than 20 seconds)
3. Interact with app
4. Click stop recording
5. Open up user timing and inspect component tree



AS YOUR PRODUCT GROWS, HOW DO YOU KEEP REACT FAST?

1. Define presentation vs. container components
2. Utilize performance timeline to determine bottlenecks
3. Remove unnecessary renders

```
3.      if(this.props.someVal !==
nextProps.someVal) {
4.        //do somethin' fancy!
5.        //render will be called
6.        return true;
7.      }
8.      else {
9.        //do nothing!
10.       return false;
11.     }
12.   }
13.   render() {
```

If child's props don't change, then don't re-render

```
17.
```

AS YOUR PRODUCT GROWS, HOW DO YOU KEEP REACT FAST?

1. Define presentation vs. container components
2. Utilize performance timeline to determine bottlenecks
3. Remove unnecessary renders
4. Prioritize network calls

PRIORITY OF RENDERING ELEMENTS

- Highest: HTML
- Highest: Styles
- High: Ajax/XHR/fetch
- Depends: Images
- Depends: Scripts
- Low: Font

HOW TO PRIORITIZE NETWORK CALLS

1. Open Dev tools
2. Go to Network tab
3. Right click to show "Priority" column

```
1. <link rel="preload" as="font"
href="font.woff" type="font/woff2"
crossorigin />
2. <link rel="preconnect"
href="https://fonts.gstatic.com/"
crossorigin>
3.
```

Add preconnect for remote fonts such as Google Fonts

AS YOUR PRODUCT GROWS, HOW DO YOU KEEP REACT FAST?

1. Define presentation vs. container components
2. Utilize performance timeline to determine bottlenecks
3. Remove unnecessary renders
4. Prioritize network calls
5. Code splitting

CODE SPLITTING!

React Training's Code Splitting Example

AS YOUR PRODUCT GROWS, HOW DO YOU KEEP REDUX FAST?

1. Set state is okay

But set state is bad!

AS YOUR PRODUCT GROWS, HOW DO YOU KEEP REDUX FAST?

1. Set state is okay
2. Connect at multiple components

Connect at multiple components

AS YOUR PRODUCT GROWS, HOW DO YOU KEEP REDUX FAST?

1. Set state is okay
2. Connect at multiple components
3. Normalized state shape

```
17.         allIds : ["alert1",
"alert2"]
18.     },
19.     comments : {
20.         byId : {
21.             "comment1" : {
22.                 id : "comment1",
23.                 author : "leg0l45",
24.                 comment : "Sauron is
at the other end of the Palantir",
25.             },
26.             "comment2" : {
27.                 id : "comment2",
```

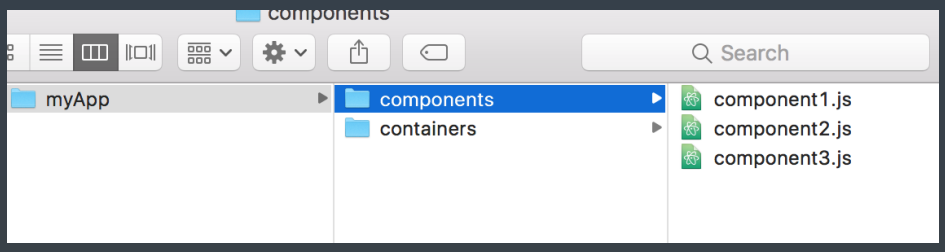
The value is the object itself

```
Balrog is rumbling beneath",
30.             },
```

YOUR DEV TEAM GROWS , HOW DO YOU KEEP THE AGILITY OF DEVELOPMENT FAST?

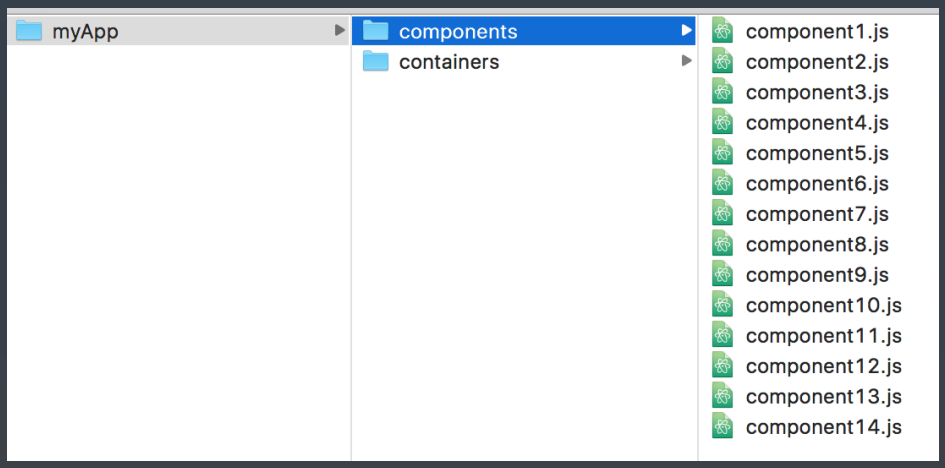
1. File structure

COMMON FILE STRUCTURE



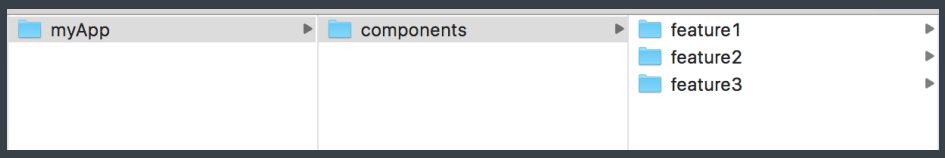
COMMON FILE STRUCTURE

COMMON FILE STRUCTURE WITH LOTS OF COMPONENTS

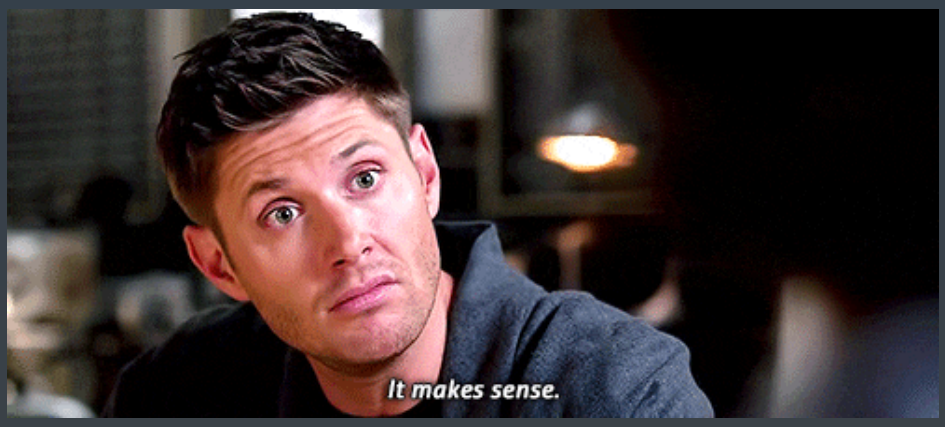


COMMON FILE STRUCTURE WITH LOTS OF COMPONENTS

CURRENT FILE STRUCTURE



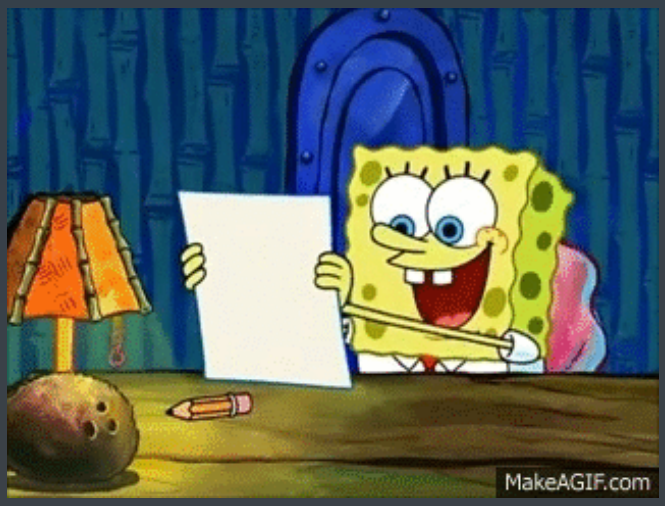
CURRENT FILE STRUCTURE



YOUR DEV TEAM GROWS, HOW DO YOU KEEP THE AGILITY OF DEVELOPMENT FAST?

1. File structure
2. Documentation & onboarding

PLZ WRITE GOOD DOCUMENTATION



YOUR DEV TEAM GROWS, HOW DO YOU KEEP THE AGILITY OF DEVELOPMENT FAST?

1. File structure
2. Documentation & onboarding
3. Static type checking

WHETHER OR NOT YOU USE THESE, YOUR JS STILL HAS TYPE

YOUR USER BASE GROWS , HOW DO YOU MAINTAIN APP USABILITY ?

1. Accessibility

WHY DOES ACCESSIBILITY MATTER?

CREATE AN ACCESSIBLE PRODUCT

1. Fully functional web app with keyboard only
2. Sufficient color contrast for users with low vision
3. Screen readers

BEGIN ENABLING WEB TO ASSISTIVE TECHNOLOGIES

1. Include Accessible Rich Internet Applications (ARIA) labels for screen readers
2. Focus control for keyboard interaction

TRY IT NOW: KEYBOARD TESTING!

With keyboard only:

1. Press tab, shift+tab, and arrow keys to change focus on different elements
2. Press enter to interact with an element

YOUR USER BASE GROWS , HOW DO YOU MAINTAIN APP USABILITY ?

1. Accessibility
2. Testing for accessibility

TESTING FOR ACCESSIBILITY

1. Keyboard testing
2. react-a11y
3. react-axe

YOUR USER BASE GROWS , HOW DO YOU MAINTAIN APP USABILITY ?

1. Accessibility
2. Testing for accessibility
3. Testing React/Redux

TESTING IN REACT/REDUX

> **...You can just walk off the stage when your talk is done**
>
> *- Jamison Dance, React Rally Coordinator*
>
> *Aug 25, 2017*