

Moda Jovem: Moving from Angular to React

Ilya Gurevich

Who am I?

- Graduate from Georgia Institute of Technology
- Frontend engineer
- Water polo player
- World traveler
- Music head







What is Loadsmart?

Logistics



Technology

Imagine

18,000 kg of beer from New York to Los Angeles



In the past...

Call to Broker

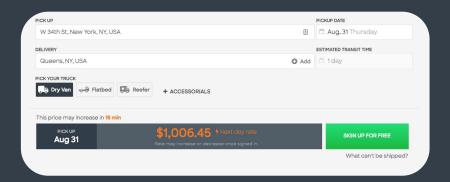


- Takes a long time
- Expensive



Happening now!

Instant Quoting



- Automated
- Algorithmic
- Accurate





Imagine



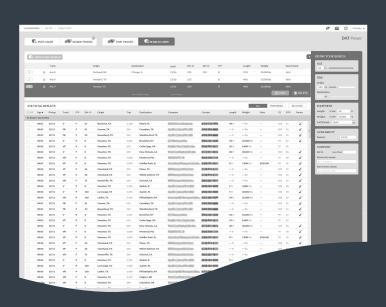
You own a truck.

You need business.

Where do you go?

In the past...

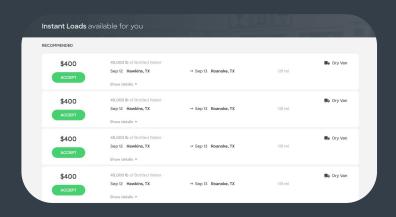
Search Load Boards



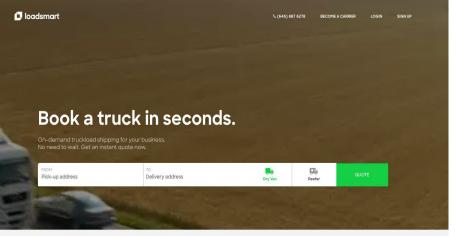
- Confusing
- Overwhelming

Happening now!

Customized Loads



- One-click interaction
- Specifically tuned for the user



WALL STREET JOURNAL

SIFREIGHT WAVES

INSIDER

facilitate booking and the tracking of goods

and carriers to become a disruptor in the freight market."

this niche by investing heavily in its

The smart way.

the time to process a spot shipment - going from 5 hours to just 18 minutes on average."

Gartner

Loadsmart was named a Cool Vendor by Gartner in Intelligent Supply Chain **Execution Technologies**

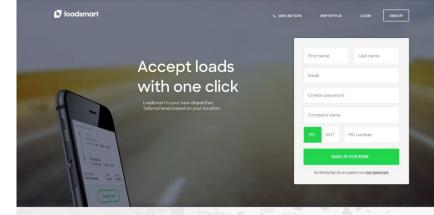


The old way.



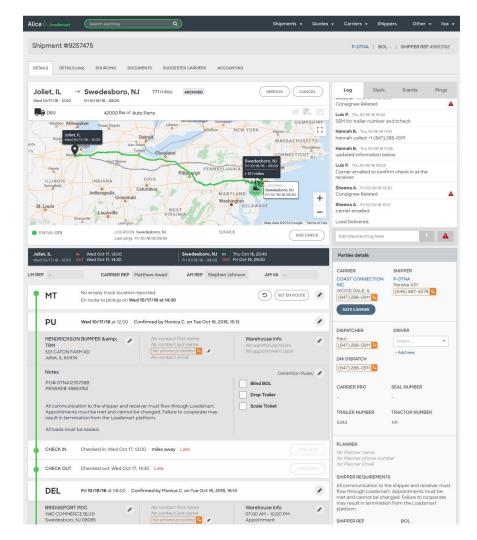
to book visibility carriers

Instant Instant Real time High quality quotes booking tracking carriers









Internal Tools

- Google Maps Tracking
- Slack Integration
- Bria Phone Client
- Real-time Shipment Management

Frontend Stack

HTML























3rd Party Tools











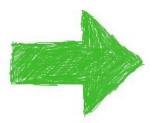


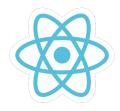




Angular to React









"Lead Angular Expert after announcement about moving the project to React.js"

Jan Matejko

Why we wanted to change

The Programmer Answer

We want a framework that's faster and simpler to develop with.



The Honest Answer?

We just wanted to update our

framework because everyone else was doing it.

Also known as **hype driven development**

But Mainly?



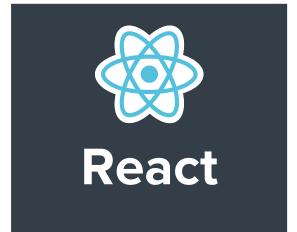
Just kidding

We did our research

Research







Time For Some Reading



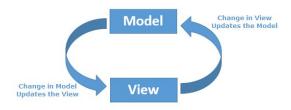


Angular 2



Pros

- Two-way data binding
- Component-based
- Better speed and performance than Angular 1
- AOT (Ahead-of-Time compilation)
- Directives and filters are awesome
- Unit test friendly



Angular 2, 4, 6, 7?

A

Cons

- Typescript (debatable)
- Unclear documentation
- Really high learning curve



- "Angular 2 is terrible"
- Regular DOM
- Large
- Versions keep on changing, unclear usage
- Verbose and complex

VueJS

Pros

- Lightweight
- Fast build process
- Angular-type directives
- Component based
- Virtual DOM
- Two-way binding on inputs
- Amazing documentation
- Creator is active in the developer community
- Arguably faster than React in some cases





VueJS

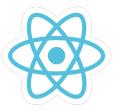
Cons

- Semi-confusing syntax
- Small community (harder to hire)
- Free-form conventions
- Majority of developers in the beginning were Chinese
- Caused a language barrier
- Lack of common plugins and components





React



Pros

- Component-based
- Virtual DOM
- Light-weight
- Seamless development experience
- Good testing compatibility
- Huge community
- Constant meaningful and informative updates
- React-friendly NPM/Yarn community



React

Cons

- React isn't a framework
- JSX learning curve
- Documentation could be better
- Nested props callbacks
- Update values on forms manually (FORMS!)
- State management is potentially difficult

Why we ultimately chose React



React over VueJS



VueJS is still too small

React has a larger community

React is better for large applications



But...

VueJS is growing

Definitely a framework to look forward to



Immediate Roadblocks



Other Immediate Roadblocks...









```
VEBPACK LOADER = 1
        CACHE': not DEBUG.
        BUNDLE_DIR_NAME': 'bundles/', # must end with slash
       'STATS FILE': os.path.join(FRONTEND PATH, 'scripts', 'webpack-stats.json'),
       'TIMEOUT': None.
        'IGNORE': ['.+\.hot-update.js', '.+\.map']
```

Questions to Question

How to do proper form validation?

Which HTTP client to use?

What even is JSX?

ES6? ES7??

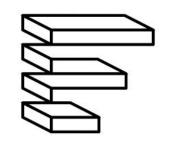


These things take time to learn...

- React Novice
 - 3-5 days
- React Expert
 - Two months

You are always learning!!





Formik

AXIOS

Immediate Simplifications

Angular is a monster compared to React

Files are much more organized

Small and fast build size

Bad things

Nested Prop Callbacks

Updating State

Updating State

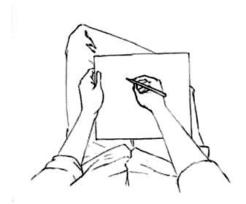
- Updating components in react can be tricky
- Component Lifecycle
 - ComponentDidUpdate
 - ComponentWillUpdate (unsafe)
 - ComponentWillReceiveProps (unsafe)
 - ComponentDidMount
 - ComponentWillMount
 - ShouldComponentUpdate
 - Etc...
- Memoization?
- Can only be learned with experience over time



Nested Prop Callbacks

One function - showAccepted()

- List
 - LoadsControl
 - Loads
 - WebList
 - AcceptDialogWeb
 - ConfirmationDialog
 - DriverLocationDialog



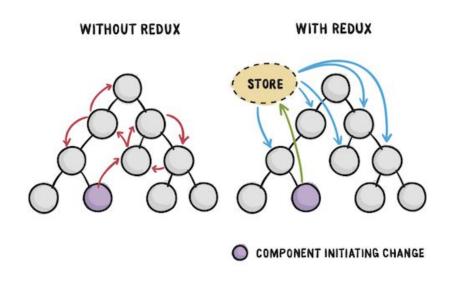
That's 7 different components

The Solution?

Use Redux

What's Redux?

- Retrieve and update props globally to be accessible via a provider from any component.
- Re-renders component in real time without memoization
- No longer necessary to use nested props callbacks!



Redux Difficulties

- Really high learning curve
- May take at least a week or two to learn properly + learning React
 - o Delays future development if on a time crunch
- Nomenclature can be obscure
- Feels very heavy
- MobX as an alternative?

Would NOT recommend using Redux without intermediate knowledge of React first



Lots of Setup for Redux

- Reducers + Combined Reducers
- Action creators
 - Loading, error, success state
- Providers
- Store configuration
- Redux devtools
- Middleware

```
import { handleActions } from 'redux-actions
export const charges = handleActions(
    SET_INCOMES: (state, action) => ({ ...state, incomes: action.payload }),
    SET_OUTCOMES: (state, action) => ({ ...state, outcomes: action.payload }),
    SET_CHARGE: (state, action) => ({ ...state, charge: action.payload })
    SET CHARGE ERROR: (state, action) => ({ ...state, chargeError: action.payload })
    SET_CHARGE_SUBMITTING: (state, action) => ({ ...state, isSubmitting: action.payload }),
    CLEAR_CHARGE: state => ({ ...state, charge: undefined, chargeError: undefined }),
import ShootCarrierButton from './ShootCarrierButton'
import store from '../../store/configureStore
      <ShootCarrierButton {...this.props}>{this.props.children}</ShootCarrierButton>
export default ShootCarrier
export const metricsMiddleware = () => next => action => {
    if (!action.payload) return next(action)
     if (action.payload.metrics === true) {
       sendMetric(action)
     delete action payload metrics
     next(action)
   } catch (error) {
     console error(error)
```

Good things

JSX with ES6/7 is awesome

Why?

- Conditional rendering is simple
- HTML + Javascript makes sense
 - Functions in markup
- Testing with jest is easy
- Not restricted to template language
- Destructuring, object rest spread, etc



Long-term implications

- We have some serious tech debt.
- We haven't converted everything from angular to react.
- Rebuilding the whole website from scratch is time-consuming.
- Given the nature of the industry a new framework will come along that completely changes everything anyways.

Final Thoughts

Was it worth it?

Yes, definitely



THANK YOU FOR YOUR TIME