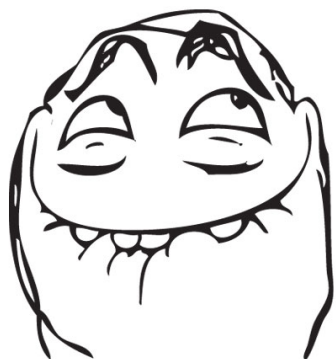


React



~~Not made by Google~~

What is ReactJS

- JS Framework
 - Uses JSX as proprietary language
- Models Data



Find us on
Instagram



What are other options?

- AngularJS
- Angular 2.0
- Blaze
- EmberJS
- BackboneJS



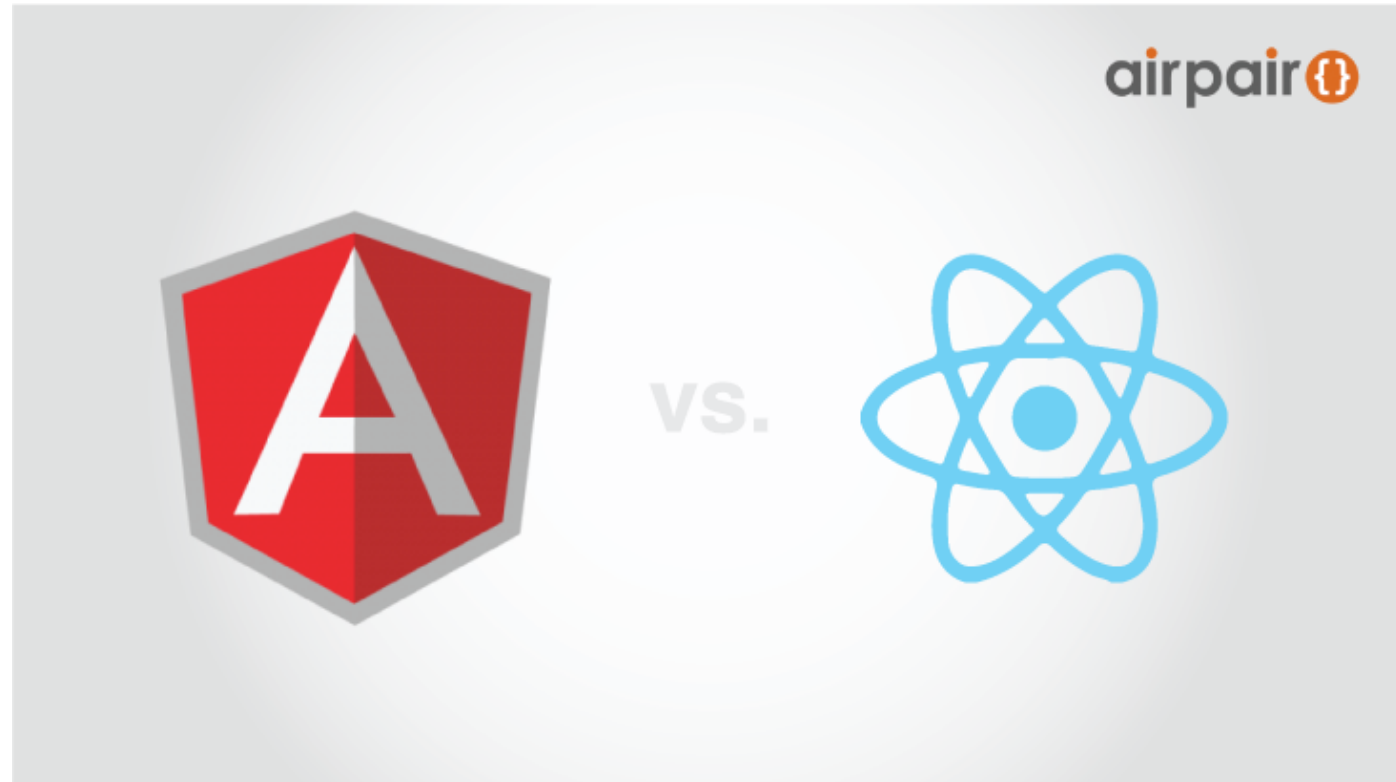
So why are we going with React?

React

- Larger Projects = Good performance
- Very modular
- Can be used in legacy projects
- React apps are light
- Uses states, reduces mutability concerns
- HTML mixed with JS

Angular

- MVC pattern
- Heavy framework
- Exponentially slow the larger a project becomes
- Angular2.0 is very “react-like”
- It was great last year, but it’s 2016



What's the deal with Angular?!

Watchers -> Digest Loop

+

Large App

=

CHAOS

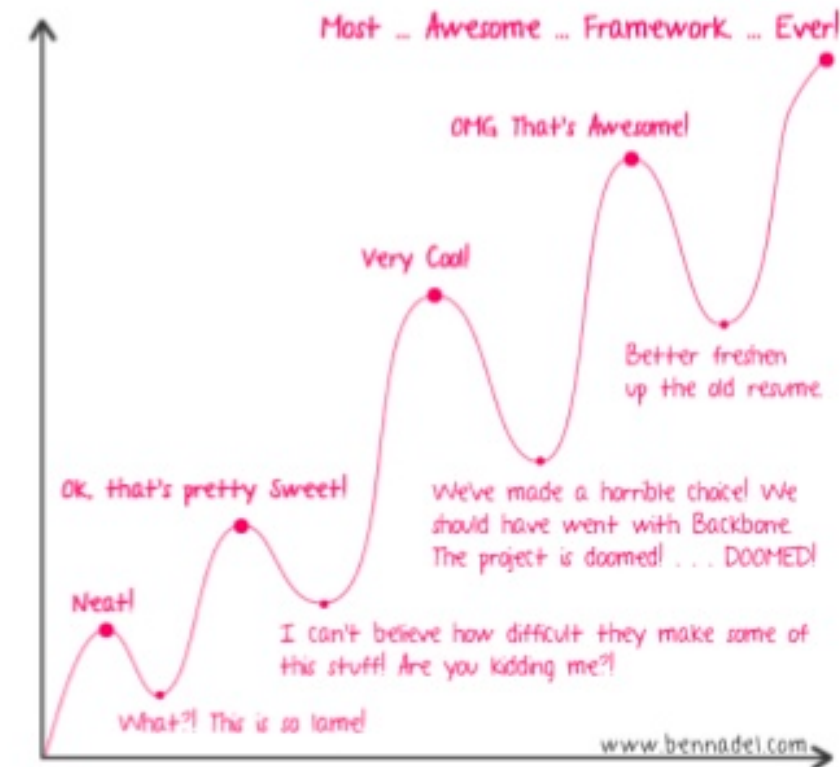
Translation: When data changes, the views you see have to change.

Each data change triggers a reload on every element.

This is terribly slow in a large app.

Optimizations are possible, but increase in numbers exponentially quickly as the application grows.

Learning curve



My Feelings About AngularJS Over Time

what is “virtual DOM”

DOM

- *Elements* of HTML become *nodes* in the DOM
- DOM provides an interface (API) to traverse and modify the nodes.
- Huge DOM trees = performance issue

Virtual DOM

- abstraction of the HTML DOM
- local and simplified copy of the HTML DOM
- can manipulated quickly without actually touching the real DOM

The Component

A component is a JSX HTML element that we render through React's Virtual DOM

Live JSX Editor

Compiled JS

```
var HelloMessage = React.createClass({
  render: function() {
    return <div>Hello {this.props.name}</div>;
  }
});

ReactDOM.render(<HelloMessage name="John" />, mountNode);
```

Hello John

State

<https://facebook.github.io/react/>

A Stateful Component

In addition to taking input data (accessed via `this.props`), a component can maintain internal state data (accessed via `this.state`). When a component's state data changes, the rendered markup will be updated by re-invoking `render()`.

Live JSX Editor

Compiled JS

```
var Timer = React.createClass({
  getInitialState: function() {
    return {secondsElapsed: 0};
  },
  tick: function() {
    this.setState({secondsElapsed: this.state.secondsElapsed + 1});
  },
  componentDidMount: function() {
    this.interval = setInterval(this.tick, 1000);
  },
});
```

Seconds Elapsed: 97

Props

<https://facebook.github.io/react/>

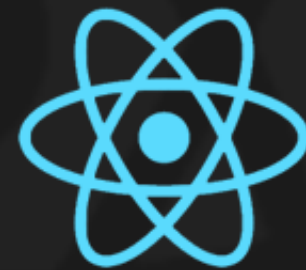
JSX (JS + HTML)

You can pass state objects between components

```
return <ul>{this.props.items.map(createItem)}</ul>;
```

To really learn .. lets do a demo! :)

BUILD A REAL TIME APP



React.js