React.js

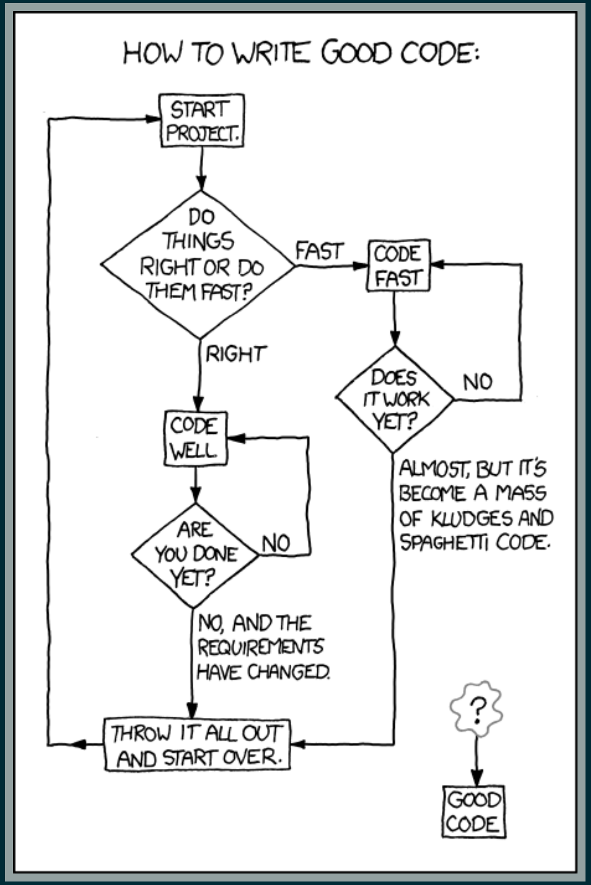
React.js for the win! - STLJS

Jeff Barczewski

codewinds.com

Who am I?

- Veteran - 26 yrs as professional developer
- Last 3+ years fulltime in JavaScript and Node.js
- Created CodeWinds to publish high quality video training courses
- @jeffbski jeff@codewinds.com


HOW TO WRITE GOOD CODE:
START PROJECT.
DO THINGS RIGHT OR DO THEM FAST?
FAST
CODE FAST
DOES IT WORK YET?
NO
ALMOST, BUT IT'S BECOME A MASS OF KLUDGES AND SPAGHETTI CODE.
RIGHT
CODE WELL
ARE YOU DONE YET?
NO
NO, AND THE REQUIREMENTS HAVE CHANGED.
THROW IT ALL OUT AND START OVER.
?
GOOD CODE

Good Code by XKCD Licensed CC BY-NC 2.5

Question for you

What are your greatest difficulties or challenges that you face when building web apps?

Agenda

- Learn why React.js is special
- Core concepts
- Family: react-router, Flux, React Native

What makes React.js special?

- Simple to learn
- Composable Components
- Declarative
- Easy to use with existing projects

Kevin Old - Coupa

"I have been surprised at how easy it was to incorporate in a very established codebase..."

Demo

codewinds.com

CodeWinds

React.js core concepts

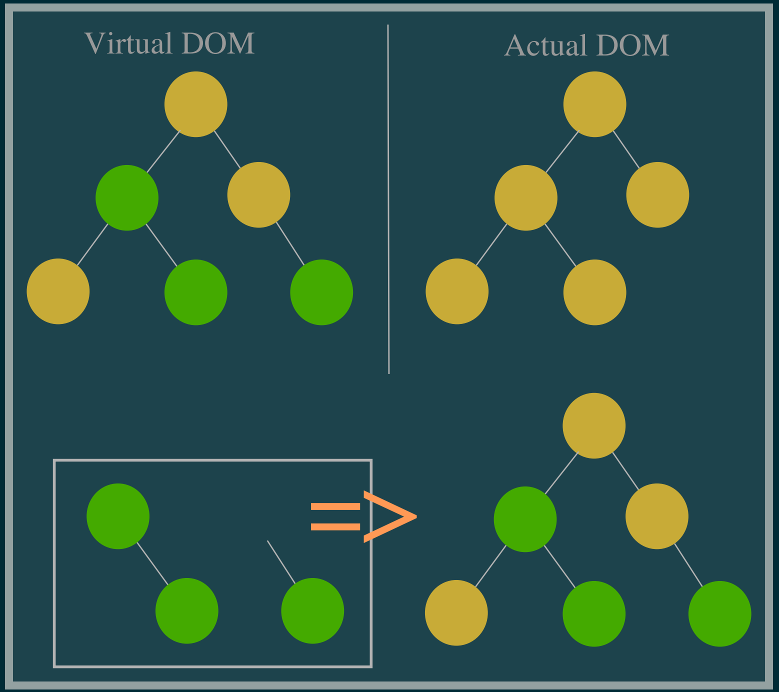
- Just the view
- Virtual DOM
- Components
- Top down data flow

Just the view

```
class Greeting extends React.Component {
  render() { // the heart of React.js
    // pass in any type of js objects, methods, or fns
    return <div>{ this.props.user.name }</div>;
  }
}
```

Virtual DOM

- Render virtual tree
- Fast diff
- Adapters to DOM, ...



Components

```
class Name ... // can be in another file
class Greeting extends React.Component {
  render() { // focus on my props
    return (
      <div>
       Hello
        <Name value={this.props.name} />
      </div>
    );
  }
}
```

Top down data flow

```
// from main.js
<App name={appName} items={items} />
```

```
// elsewhere in App's render()
return (<div>
  <MainTitle title={this.props.name} />
  <PagingList collection={this.props.items} />
</div>);
```

```
// elsewhere in MainTitle's render()
return <h1>{this.props.title}</h1>;
```

React.js API - JSX

```
return ( // JSX is optional but really give it a try
  <div>
    <h1>{this.props.title}</h1>
    <MyDate val={this.props.date} />
  </div>
);
```

```
// transforms into
return React.createElement('div', {}, [
  React.createElement('h1', {}, this.props.title),
  React.createElement(MyDate, { val: this.props.date }, [])
]);
```

React.js API - JSX p2

```
const errorStyle = {
 color: 'red',
 marginRight: '10px'
};
const html = marked('The quick brown fox...');
return (
  <div className="foo bar" data-baz="a"> { /* comment */ }
    <span style={errorStyle}>{this.props.msg}</span>
    <input autoFocus="true" onClick={this.myfn.bind(this)} />
    <div dangerouslySetInnerHTML={{ __html: html }} />
  </div> );
```

React.js API - render, props

```
var mainDiv = document.querySelector('#mainDiv');
React.render(<App title={myTitle} items={myItems} />,
             mainDiv); // apply here
```

```
// elsewhere in App's render()
return (
  <div>
    <h1>{this.props.title}</h1>
    <ul>{this.props.items.map(i =>
          <li key={i.id}>{i.name}</li>)}
    </ul>
    ...
```

React.js API - state, events

```
class MyComp extends React.Component {
  constructor(...args) {
    super(...args);
    this.state = { count: 0 };
  }
  render() { return (
    <button onClick={this.clicked.bind(this)} >
      {this.state.count} </button> );
  }
  clicked(e) { this.setState({ count: this.state.count + 1 }); }
```

React.js API - forms

```
// uncontrolled comps, set orig value, can watch w/events
return (<div>
  <input name="first" defaultValue={this.props.first}
    onChange={this.firstChanged.bind(this)} />
  <textarea name="foo" defaultValue={this.props.foo} />
</div>); // <select> also works similarly
```

```
// controlled comp, force value w/state, this.state = {}
return (<div><input name="first" value={this.state.first}
    onChange={this.firstChgd.bind(this)} /></div>);
...
firstChgd(e) { this.setState({ first: e.target.value }); }
```

React.js API - refs

```
return (<form onSubmit={this.sendData.bind(this)}>
    <input name="foo"/>
    <button ref="submitButton">Send</button>
</form>);
```

```
sendData(e) {
  e.preventDefault();
  var submitButton = React.findDOMNode(this.refs.submitButton);
  submitButton.disabled = true; // re-enable after post completes
  // send data, then submitButton.disabled = false;
}
```

React.js API - prop validation

```
MyComponent.propTypes = {
  foo: React.PropTypes.object, // any object
  bar: React.PropTypes.shape({ f: React.PropTypes.string }),
  baz: React.PropTypes.array, // also arrayOf(propType...)
  cat: React.PropTypes.func.isRequired, // fn + required
  dog: React.PropTypes.node, // number, string, array, element
  egg: React.PropTypes.any.isRequired, // anything + required
  fig: React.PropTypes.instanceOf(Message),
  gib: React.PropTypes.oneOf(['optionA', 'optionB']), // enum
  hib: function (props, propName, compName) { // custom
    if (...) { return new Error('my validation error'); } }};
```

React.js API - default props

```
// cached and used as the defaults
MyComponent.defaultProps = {
  foo: 'default value',
  bar: 1
};
```

React.js API - Lifecycle

React Component Lifecycle

Mount

componentWillMount()
componentDidMount()

Updates

componentWillReceiveProps(nextProps)
shouldComponentUpdate(nextProps, nextState)
componentWillUpdate(nextProps, nextState)
componentDidUpdate(prevProps, prevState)

Unmount

componentWillUnmount()

React.js API - perf tuning

```
return (
  <ul>
    { items.map(i => <li key={i.id}>{i.name}</li>) }
  </ul>
);
```

```
import PureComponent from 'react-pure-render/component';
class Foo extends PureComponent {
  // implements shouldComponentUpdate with shallow compare
  // works for primatives and immutable objects
}
```

React.js API - render string

```
// can render to string on server
const str = React.renderToString(<App items={myItems} />);


// alternatively if not using React in browser,
// renderToStaticMarkup also renders to string, but
// doesn't include the React attributes (id's and checksums)
// needed for reconciliation
const str = React.renderToStaticMarkup(<App items={myItems} />);
```

Demo

React.js family

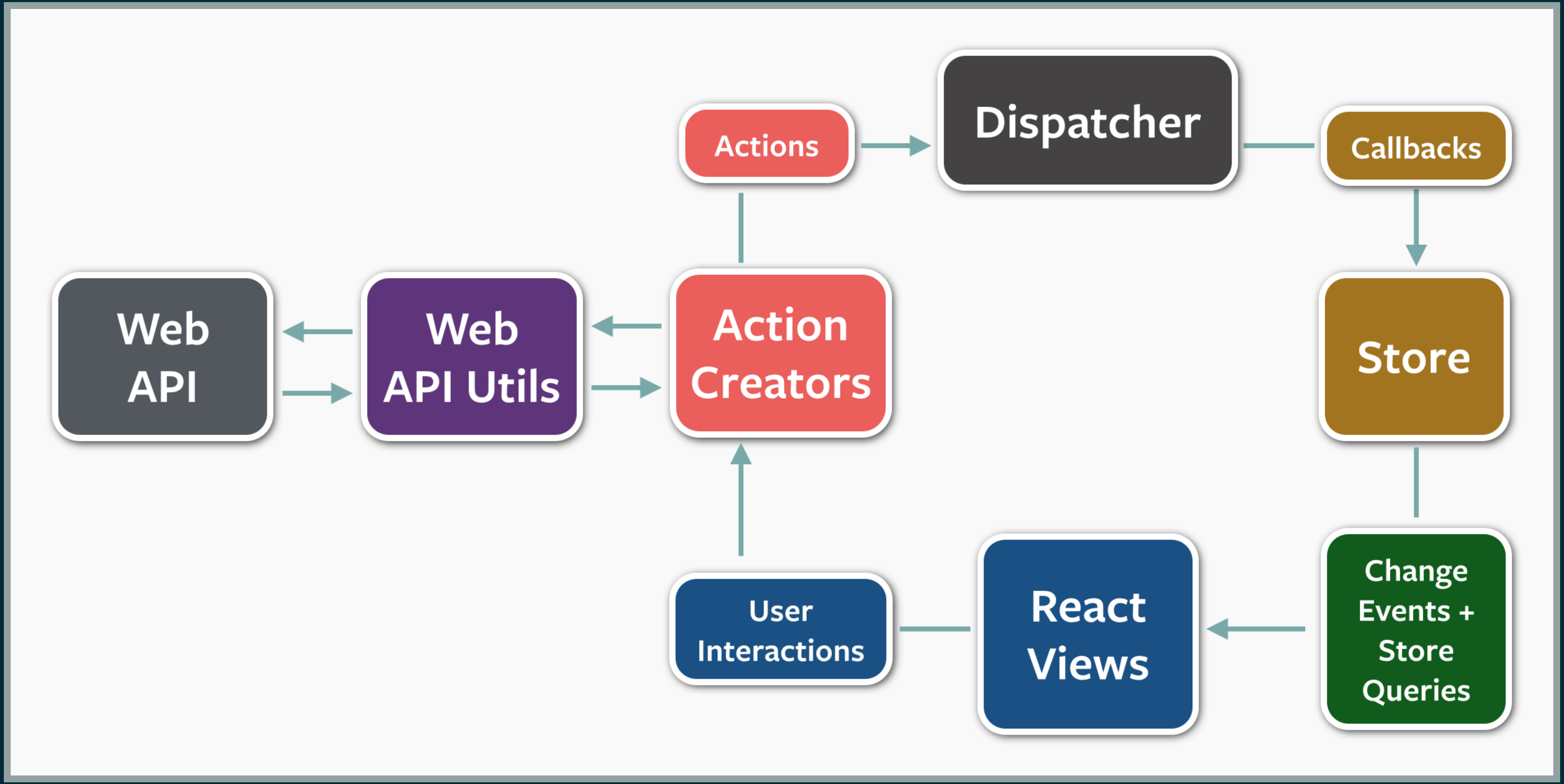
- react-router
- Flux
- React Native

react-router

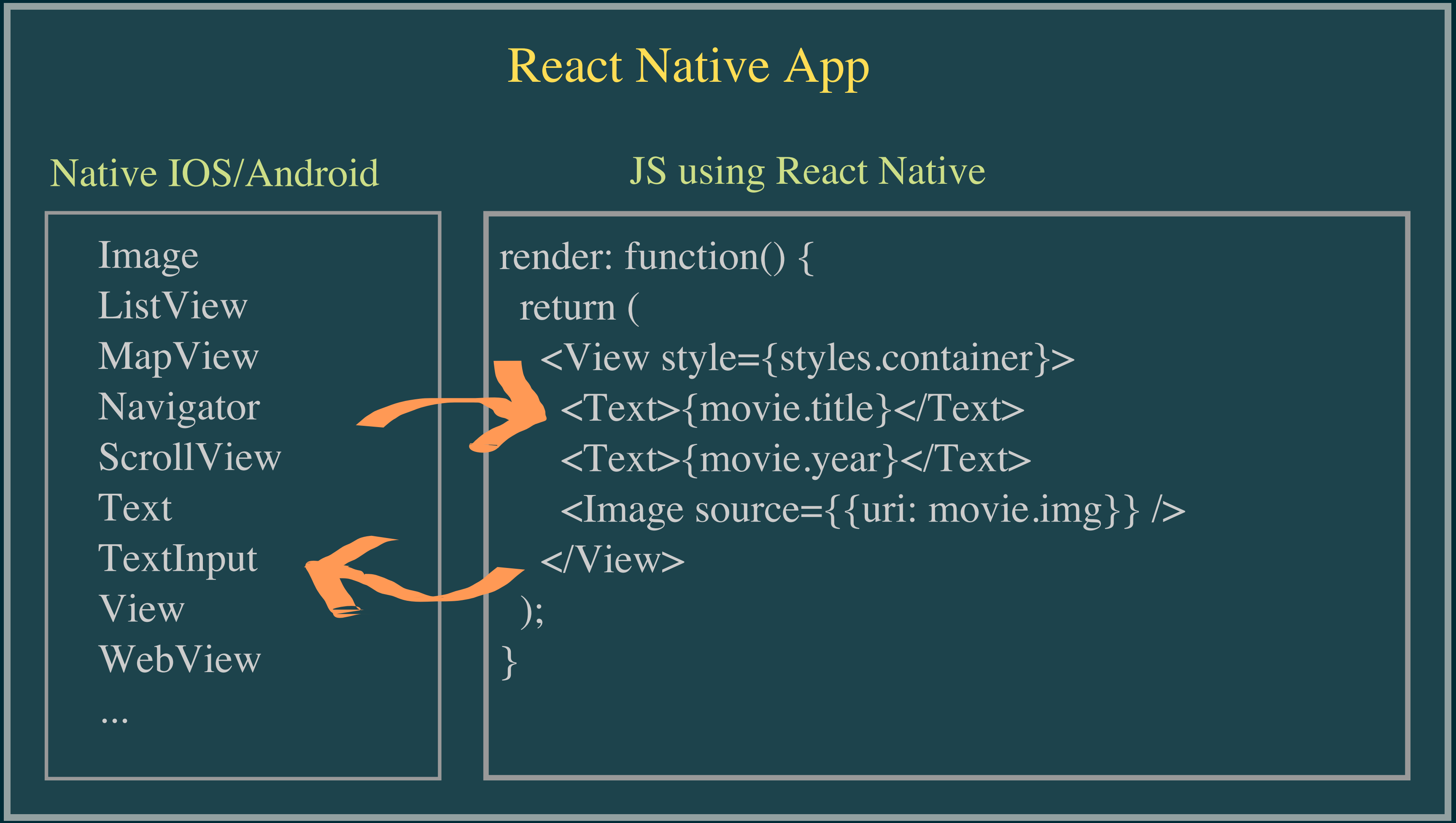
```
const routes = (<Route handler={App} path="/">
                  <DefaultRoute handler={Home} />
                  <Route name="about" handler={About} />
                </Route>);
Router.run(routes, Router.HistoryLocation, (Handler, props) =>
 // can fetch data for props.routes here
 React.render(<Handler />, document.querySelector('#appDiv'));
});
```

```
return (<div><h1>App<h1> { /* in App render() */ }
          <RouteHandler />
        </div>);
```

Flux

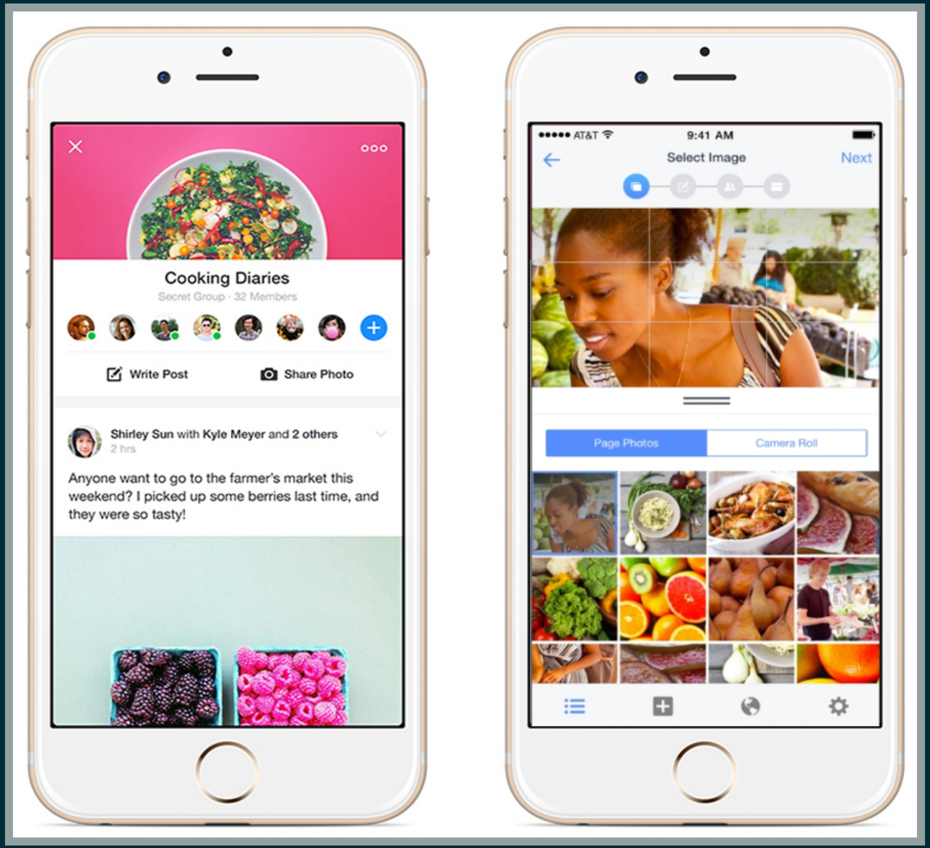

Actions
Dispatcher
Callbacks
Web API
Web API Utils
Action Creators
Store
User Interactions
React Views
Change Events + Store Queries

React Native



```
render: function() {
  return (
    <View style={styles.container}>
      <Text>{movie.title}</Text>
      <Text>{movie.year}</Text>
      <Image source={{uri: movie.img}} />
    </View>
  );
}
```

React Native p2


Cooking Diaries
Secret Group · 32 Members
Write Post
Share Photo
Shirley Sun with Kyle Meyer and 2 others
2 hrs
Anyone want to go to the farmer's market this weekend? I picked up some berries last time, and they were so tasty!
AT&T
9:41 AM
Select Image
Next
Page Photos
Camera Roll

Future

Relay/GraphQL

```
{
  user(id: 3500401) {
    id,
    name,
    isViewerFriend,
    profilePicture(size: 50)  {
      uri,
      width,
      height
```

FalcorJS

Resources

- components: http://react.parts http://npmsearch.com or Google (site:github.com react xyz)
- http://reactjs.com/
- http://reactnative.com/
- http://codewinds.com/

Summary

- React.js is a game changer
- Use it in your projects today
- Enjoy coding in a new way!
- http://codewinds.com/stljs
  - Continuing React.js coverage
  - React.js video training - summer 2015