REACT

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

PRESENTED BY

Ronn Ross

WHAT IS REACT?

A JavaScript library for building user interfaces

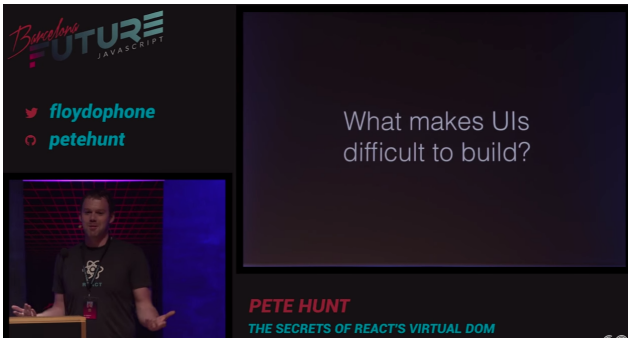
The V in MVC

> # A bridge between a declarative and an Imperative API
>
> *- Jim Sproch*

SIMPLE COMPONENT

Picture if you will an icon with three states. Simple Bell (no messages). Bell with a count (1-99 messages). Bell on fire (99+) messages.

```
if (count > 99) {
  if (!hasFire()) { addFire(); }
} else {
  if (hasFire()) { removeFire(); }
}

if (count === 0) {
  if (hasBadge()) { removeBadge(); }
  return;
}

if (!hasBadge()) { addBadge(); }

var text = count > 99 ? '99+' : count.toSTring();
getBadge().setText(text);

-------------------------------------

<Messages count={count} />

count === 0 && <Bell />
count < 99 && <Bell count={this.props.count} />
count > 99 && <Bell type="fire" />
```



Virtual DOM

PETE HUNT - SECRETS OF REACT'S VIRTUAL DOM

- Facebook
- Instagram
- Netflix
- Walmart
- Github
- PayPal
- Hip Chat
- Khan Academy
- Yahoo
- AirBnb
- Microsoft
- Twitter (fabric)
- KiteMatic
- And more ...

BASIC COMPONENT

```
//App.js
import React from 'react';
import ReactDOM from 'react-dom';

class App extends React.Component {
  render(){
    return(
      <h1>Hello World</h1>
    )
  }
}

ReactDOM.render(<App />, document.getElementById("main"));
```

COMPOSE COMPONENTS

```
//Second.js
import React from 'react';

class Second extends React.component {
  render() {
    return(
      <h2>Hello world a second time</h2>
    )
  }
}

export default Second;

//App.js
import React from 'react';
import ReactDOM from 'react-dom';
import Second from './Second.js';

class App extends React.Component {
  render(){
    return(
      <h1>Hello World</h1>
      <Second />
    )
  }
}

ReactDOM.render(<App />, document.getElementById("main"));
```

DATA DOWN

```
//App.js
import React from 'react';
import ReactDOM from 'react-dom';

class App extends React.Component {
  render() {
    return(
      <div>
        <h1>{this.props.message}</h1>
      </div>
    )
  }
}

ReactDOM.render(<App message="Hello World" />, document.getElementById("main"))
```

AND DOWN

```
//Second.js
import React from 'react';

class Second extends React.component {
  render() {
    return(
      <h1>{this.props.message}</h1>
    )
  }
}

export default Second;

//App.js
import React from 'react';
import ReactDOM from 'react-dom';

class App extends React.Component {
  render() {
    return(
      <Second message={this.props.message} event={obj} />
    )
  }
}

ReactDOM.render(<App message="Hello World" />, document.getElementById("main"))
```

COMPONENTS SHORTHAND

```
//Second.js
let Second = ({message}) => (
  <h1>{message}</h1>
);

export default Second;

//App.js
import React from 'react';
import ReactDOM from 'react-dom';

class App extends React.Component {
  render() {
    return(
      <Second message={this.props.message} />
    )
  }
}

ReactDOM.render(<App message="Hello World" />, document.getElementById("main"))
```

COMPONENT API

```
//Second.js
...
  render() {
    return(
      <div>
        <h1>{this.props.message}</h1>
        <p>Count is {this.props.count}<p>
      <div>
    )
  }
}
second.propTypes = {
  message: React.PropTypes.string,
  name: React.PropTypes.number
};
...

//App.js
...
  render() {
    return(
      <Second message="Hello World" count={1} />
    )
  }
...
```

COMPONENT LIFECYLE

```
//App.js
import React from 'react';

class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {count: props.initialCount}
  }

  componentWillMount(){}

  componentWillReceiveProps(){}

  componentWillUpdate(){}

  render() {
    return(<button onClick={::this.clickEvent}></button>)
  }

  //custom event
  clickEvent() {
    this.setState({count: this.state.count + 1})
  }

  componentWillUnmount() //any cleanup needed
}

App.propTypes = { initialCount: React.PropTypes.number }
App.defaultProps = { initialCount: 0 }
```

DEMO TIME

RICH COMPONENT ECOSYSTEM

REACT ROCKS

REACT COMPONENTS