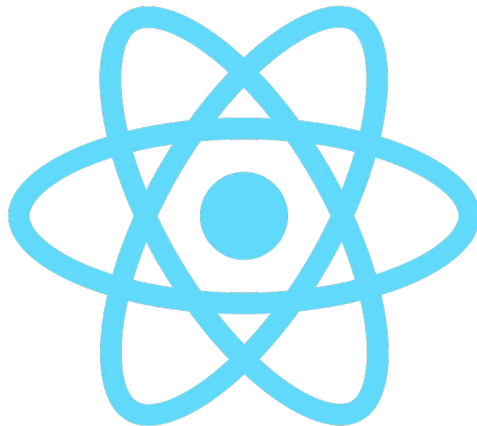


# React & React Native

# React

Library for building User Interfaces

- declarative
- component-based
- small API surface
- learn once, write everywhere
  - React Native
  - React for AR/VR
  - React for IoT



# Learn React (API) in 5 minutes

- Component = piece of UI
- The `render()` method
  - light representation of the UI
  - written in JSX
- 2 kinds of data
  - props (=external data)
    - `this.props.name`
  - state (=internal data)
    - `this.state.foo`
    - `this.setState({ foo: 'bar' })`
- **data change  $\Rightarrow$  render**

```
class HelloMessage extends React.Component {  
  render() {  
    return (  
      <div>  
        Hello {this.props.name}  
      </div>  
    );  
  }  
}
```

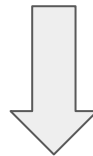
```
<HelloMessage name="Taylor" />
```

# Virtual Layout (DOM)

```
<ul>  
  <li key="1">Task 1</li>  
  <li key="2">Task 2</li>  
</ul>
```



```
<ul>  
  <li key="1">Task 1</li>  
  <li key="2">Task 2</li>  
  <li key="3">Task 3</li>  
</ul>
```



```
<ul>  
  <li key="2">Task 2</li>  
  <li key="3">Task 3</li>  
</ul>
```



## PATCH:

- Insert `<li key="3">Task 3</li>` at the end of `<ul>`
- Delete `li` with key "1"

# Component Life Cycle

- Mounting
  - [componentWillMount\(\)](#)
  - [componentDidMount\(\)](#)
- Update
  - [componentWillReceiveProps\(\)](#)
  - [shouldComponentUpdate\(\)](#)
  - [componentWillUpdate\(\)](#)
  - [componentDidUpdate\(\)](#)
- Unmounting
  - [componentWillUnmount\(\)](#)
- Error Handling
  - [componentDidCatch\(\)](#)

# Example of a React Component /1

```
1 class CountdownTimer extends React.Component {
2
3   constructor(props) {
4     super(props);
5     this.state = {
6       secondsRemaining: props.secondsRemaining,
7     };
8   }
9
10  tick() {
11    this.setState({
12      secondsRemaining: this.state.secondsRemaining - 1,
13    });
14
15    if (this.state.secondsRemaining <= 0) {
16      clearInterval(this.interval);
17    }
18  }
19 }
```

```
19
20  componentDidMount() {
21    this.interval = setInterval(() => this.tick(), 1000);
22  }
23
24  componentWillUnmount() {
25    clearInterval(this.interval);
26  }
27
28  render() {
29    return (
30      <div>Seconds Remaining: {this.state.secondsRemaining}</div>
31    );
32  }
33 }
```

# Composition

```
35 class CountdownPage extends React.Component {  
36     render() {  
37         return (  
38             <div>  
39                 <CountdownTimer secondsRemaining="60" />  
40                 <CountdownTimer secondsRemaining="5" />  
41             </div>  
42         );  
43     }  
44 }
```

# React Native

- React for mobile apps
- Build entire apps using javascript
  - Can add native code if needed
- Cross-platform
- Real native app
- No compile wait times
- Hot-reloading
- CSS-like for styling

React	React Native
<code>&lt;div&gt;</code>	<code>&lt;View&gt;</code>
<code>&lt;span&gt;</code>	<code>&lt;Text&gt;</code>
<code>&lt;li&gt;, &lt;ul&gt;</code>	<code>&lt;FlatList&gt;</code>
<code>&lt;img&gt;</code>	<code>&lt;Image&gt;</code>



# Example of a React Component /2



```
1 class PointsCounter extends React.Component {
2   render() {
3     return (
4       <View style={styles.container}>
5         <Text style={styles.pointsText}>
6           {this.props.points}
7         </Text>
8         <Image
9           style={styles.pointsPic}
10          source={require('../assets/images/NavBarPoir
11            />
12        </View>
13      );
14    }
15  }
16  const mapStateToProps = (state) => {
17    return {
18      points: authentication.selectors.numStars(state),
19    };
20  };
21  export default connect(mapStateToProps)(PointsCounter);
```

```
23 const styles = StyleSheet.create({
24   container: {
25     alignItems: 'center',
26     justifyContent: 'center',
27     flexDirection: 'row',
28     marginRight: 10,
29   },
30   pointsText: {
31     top: 1,
32     fontSize: 22,
33     lineHeight: 30,
34     color: 'white',
35     fontFamily: 'Nunito-SemiBold',
36   },
37 });
```

# Example of a React Component /3

```
16 class ContactPermissions extends React.Component {
17   _getContacts = () => {
18     this.props.dispatch(signup.actionCreators.askContactPermissions());
19   };
20
21   render() {
22     const {isLoading} = this.props;
23
24     return (
25       <View style={styles.container}>
26         <StatusBar backgroundColor="transparent" barStyle="dark-content"/>
27
28         <View style={styles.middle}>
29           <Image
30             style={styles.image}
31             source={require('../assets/images/onboardingFindFriendsEmoji_2.png')}
32           />
33           <Text style={styles.title}>
34             {t("CONTACTS_PERMISSIONS_TITLE")}
35           </Text>
36         </View>
37
38         <View style={styles.bottom}>
39           <PrimaryButton style={styles.nextButton}
40             label={t("CONTACTS_PERMISSIONS_BUTTON_TITLE")}
41             disabled={isLoading}
42             onPress={this._getContacts}
43             showLoading={isLoading}
44           />
45         </View>
46       </View>
47     );
48   }
49 }
50
```

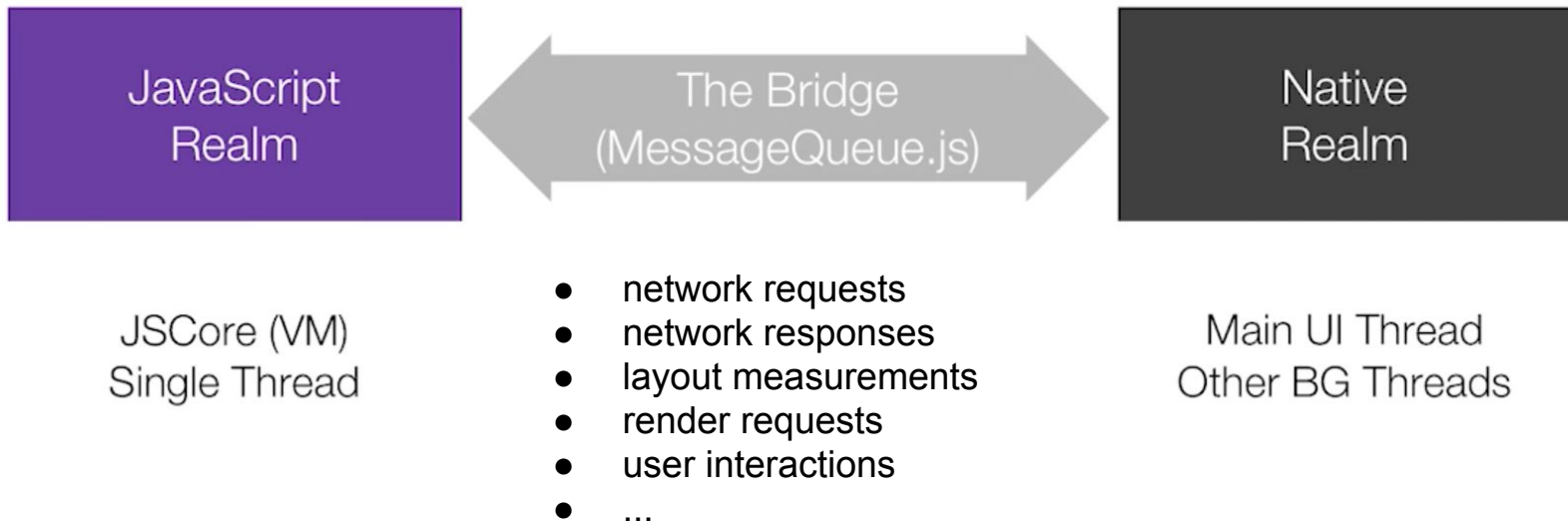
Search 15:36 90%



Cerca tra i contatti i tuoi amici che  
sono già su Uniq!

Consenti Accesso

# Inner Working

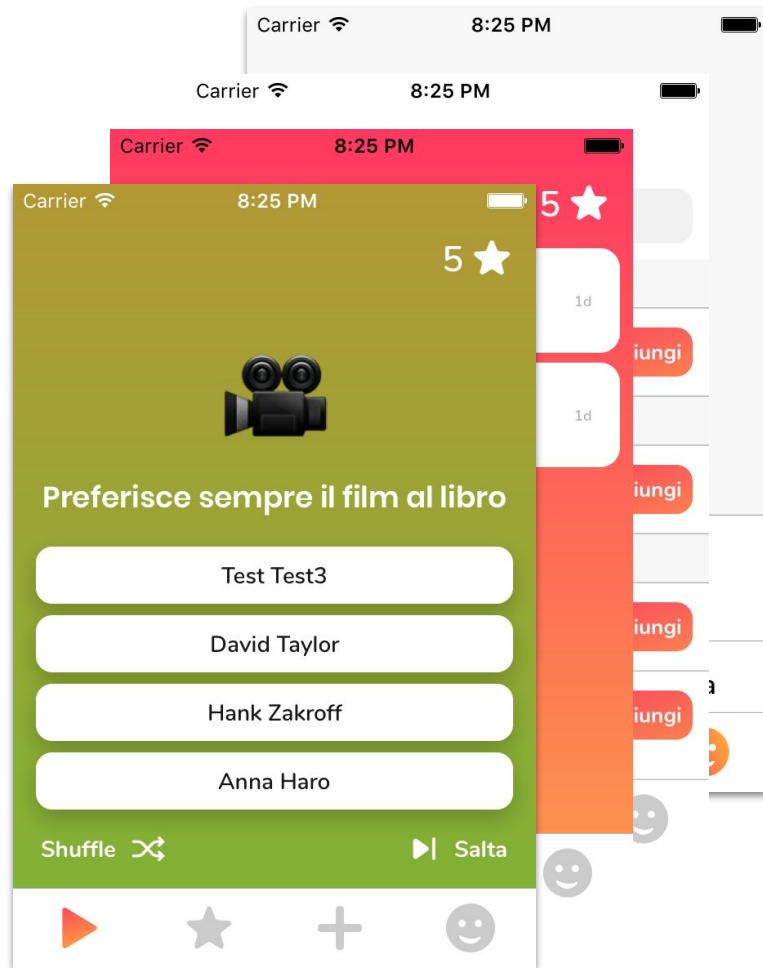


For more info:

- <https://medium.com/@rotemmiz/react-native-internals-a-wider-picture-part-1-messagequeue-js-thread-7894a7cba868>

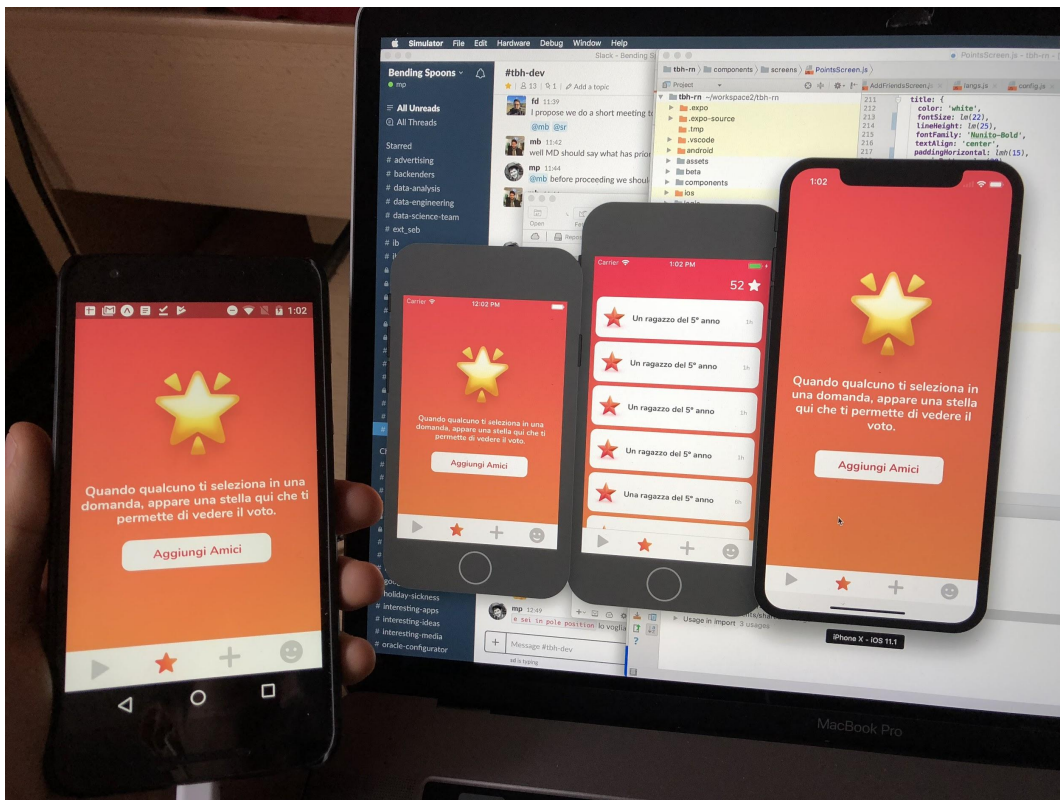
# Our experience with Uniq

- Non-trivial app
  - many interactions with native APIs: geolocation, contacts, camera, camera roll
  - lots of network requests
- 3 weeks of development (v1)
- 8k LOC (30% logic, 70% ui/style)
- Cross platform
- Support many device sizes



# What we liked

- tooling
  - Expo
  - easy setup
  - state inspector
- styling
  - declarative with css
  - live reload
  - using multiple simulators at once



# What we didn't like

- Some parts of Expo
  - still immature
  - some bugs here and there
  - painful to eject/build a standalone app
- (react-)navigation
  - slow js-animations
  - bugs

The End