

Building Cross-Platform React Apps

Building cross-platform React apps, with component-driven design.

Quotes

"React is such a good idea that we will spend the rest of the decade continuing to explore its implications and applications."

Guillermo Rauch – founder of Vercel (ZEIT)

"Here's the simple truth: you can't innovate on products without first innovating the way you build them."

Alex Schleifer – VP of Design at Airbnb

What is React?

JavaScript library for building user interfaces.

Where UI is a function of state.

```
UI = fn(state);
```

- Declarative
 - function that takes your application state and renders an interactive UI.
- Component-Based
 - Should apply to design too
- Learn Once, Write Anywhere
 - Cross-platform (with some work)

React Renderer

```
type createElement = (type, props, children) => ReactElement;
```

```
const element = {  
  type: 'h1',  
  props: {  
    className: 'greeting',  
    children: 'Hello, world!'  
  }  
};
```

```
const App = () => (  
  <div>  
    <h1>Hello World</h1>  
  </div>  
  <main>  
    <div  
      onClick={() => alert('Hello World!')}  
      style={{ flex: 1, backgroundColor: 'green' }}  
    />  
    <HomePageContent />  
  </main>  
>);
```

```
const appRawJson = {  
  "type": "div",  
  "props": {  
    "children": [{  
      "type": "h1",  
      "props": {  
        "children": ["Hello World"]  
      }  
    }  
  ]  
}
```

```
const App = () => (  
  <View>  
    <Text>Hello World</Text>  
  </View>  
  <View>  
    <Touchable onPress={() => alert('Hello World!')}>  
      <View style={{ flex: 1, backgroundColor: 'green' }}/>  
    </Touchable>  
    <HomePageContent />  
  </View>  
)  
);
```

```
const appRawJson = {  
  "type": "view",  
  "props": {  
    "children": [{  
      "type": "text",  
      // Should be something like "native_text",  
      // since it's an internal/private API  
      "props": {  
        "children": ["Hello World"]  
      }  
    }  
  ]  
}
```

React Reconciler

[Go to code](#)

Platforms

Apps traditionally target specific platforms.

- Web – HTML / CSS -> `react-dom`
- iOS – Objective-C / Swift -> `react-native`
- Android – Java / Kotlin -> `react-native`
- Mac – Objective-C / Swift -> `react-native-macos`
- Windows – C# -> `react-native-windows`
- Linux – C++ / QT -> `react-native-desktop-qt`

react-primitives

Primitive React Interfaces Across Targets.

Exposes a common API across many React platforms, that lets you build your UI once and render it to any platform.

Great for design systems and component libraries.

react-sketchapp

React Sketch.app is an open-source library that allows you to write React components that render to Sketch documents.

- Currently built with `react-test-renderer` (no runtime)
- Build components with `react-primitives`
- Render them to Sketch
- Generate design assets/library to hand over to a designer
- Svg support – (d3.js)

react-figma

React Figma is an open-source library that allows you to write React components that render to Figma documents.

- Built with `react-reconciler` (access to runtime events)

Lona

A tool for defining design systems and using them to generate cross-platform UI code, Sketch files, and other artifacts.