

# React Native for Web

What, Why, How

# Hello

- My name **Tomasz Czajęcki**
- I am a Software Engineer in **Software Mansion**
- Using React Native for the past 2 years
- My Twitter handle is **@tchayen**
- I didn't know I will be presenting until 3h ago

# Cross platform in React Native



# Why

- Less code means less bugs
- Less code means less developers (which means lower costs)
- One source of truth for business logic
- (ideally) one source of truth for styling

# Code sharing

- React is pretty shareable by design
- JS works the same way
- Hooks work in both React Native and ReactDOM
- Even React components do, with the exception of HTML elements

## What is actually missing?

```
<div/>
```

```
some label
```

```
.text { font-size: 15px; }
```


```
<View />
```

```
<Text>some label</Text>
```

```
style={{ fontSize: 15 }}
```

# Nozbe/ZACS

README.md



ZERO  
ABSTRACTION  
COST  
STYLING

👤 Component styling with no performance penalty for React and React Native ⚡

License MIT npm v0.9.3

## What is zacs ?

ZACS turns React components that look like this:

```
import zacs from '@nozbe/zacs'  
import style from './style'  
  
const Box = zacs.view(style.box, { isHighlighted: style.highlighted })  
  
const rendered = <Box isHighlighted />
```

Into **optimized** code that looks like this (**web**):

# React Native for Web

## Initial commit



Nicolas Gallagher committed on 9 Jun 2015



**How does it work**

# What comes included

- `<View>` and `<Text>` for each platform
- `fileName.web.js` and `fileName.native.js` extensions
- `Platform.OS === 'web'`
- Ports of non-obvious APIs like `Animated`, `PanResponder`

# Setup for multiple platforms isn't that easy

If you wanted to share code and use yarn workspace, you had to:

- Create project using create-react-app
- Hack it using rewired package to allow code outside of /src directory
- Hack RN in several places to look for node\_modules in a directory above
- Accept VSCode to occasionally go crazy about TypeScript typings not knowing what you are doing

# Expo for Web

Game changer I didn't expect:

- Exactly the same code base located in just one directory
- Even after giving up control to Expo, framework gives it back with ability to override `index.html` and service worker files
- You can enjoy all other features of the Expo ecosystem

```
const Spinner = ({ size }) => {
  const [animation] = useState(new Animated.Value(0));

  const runAnimation = () => {
    animation.setValue(0);
    Animated.timing(animation, {
      toValue: 1,
      duration: 1000,
      easing: Easing.bezier(0.4, 0.15, 0.6, 0.85),
      useNativeDriver: Platform.select({ ios: true, android: true }),
    }).start(runAnimation);
  };

  useEffect(runAnimation, []);

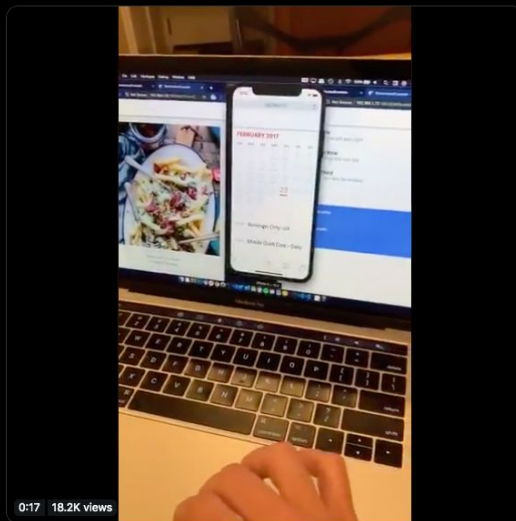
  const rotate = animation.interpolate({
    inputRange: [0, 1],
    outputRange: ['0deg', '360deg'],
  });

  return <Animated.View...
```



Evan Bacon 🍷 Vancouver ✖  
@Baconbrix

I got most of Reanimated (by @kzzzf) running in the browser with @expo web & @reactnative web 😊💙 The gestures use react-native-gesture-handler for web which is already published 🔥



# Summary

- React Native for Web is finally allowing for full React code reuse between web and mobile
- Expo for Web is a really handy tool and significantly increases DX
- Currently there is no easy way to merge existing apps to native ones
- But on the other hand I can see a lot of reasons to go with Expo for Web for new React Native projects and can't imagine a fast pacing startup picking any other tech stack at the moment

**Thanks**