Introduction ReactNative

by Michał Taberski

MICHAŁ TABERSKI
michaltaberski.com

Agenda

1. About me
2. What is React Native, and what is not?
3. How does it work?
4. Advantages of React Native approach
5. Quick start
6. Demo
7. Bonus?!

About me

- Web developer, freelance contractor,
- always open for networking, good food and Fifa challenge ;)
- JavaScript developer
  - yesterday: Ruby on Rails and BackboneJS
  - today: ReactJS, Redux and stuff
  - tomorrow: maybe Elm? (http://elm-lang.org/)

React Native is…

React Native **is not**…

- **it's not** a web based technology **like Phonegap,** (no HTML, CSS animations, jQuery)

- **it's not** "one app for all platforms solution"*** like Titanium**

*** -** although its possible you shouldn't do this

but it is…

"A FRAMEWORK FOR BUILDING NATIVE APPS USING REACT"

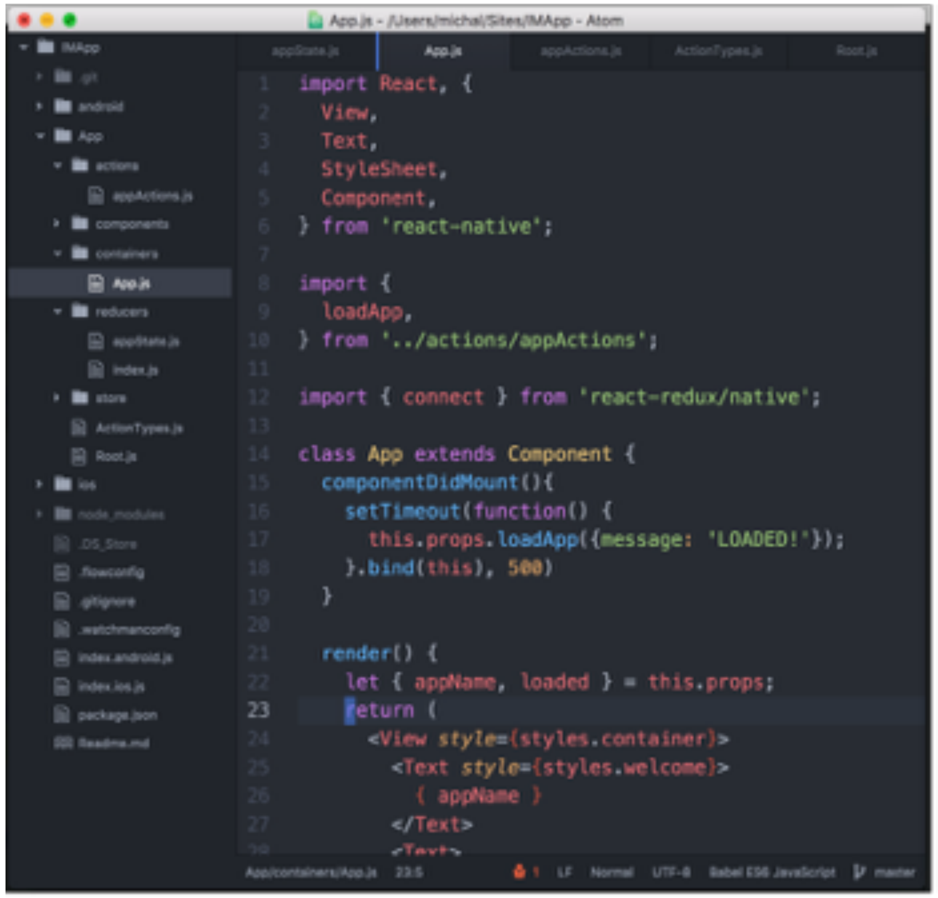
– official React Native homepage
(https://facebook.github.io/react-native/)

which means

- you can write **NATIVE APPS** with JavaScript (ES6)
- your code **looks just like ReactJS** code - JSX syntax, declarative UI describing etc.
- unlike ReactJS, ReactNative's output is not HTML, code it's not executed on frontend
- you can still use a lot of good JS libs you use already on browser (Redux, Immutable.JS, testing frameworks etc.)
- it supports **Apple iOS, and Android**

So… how does it work?

How does it work?



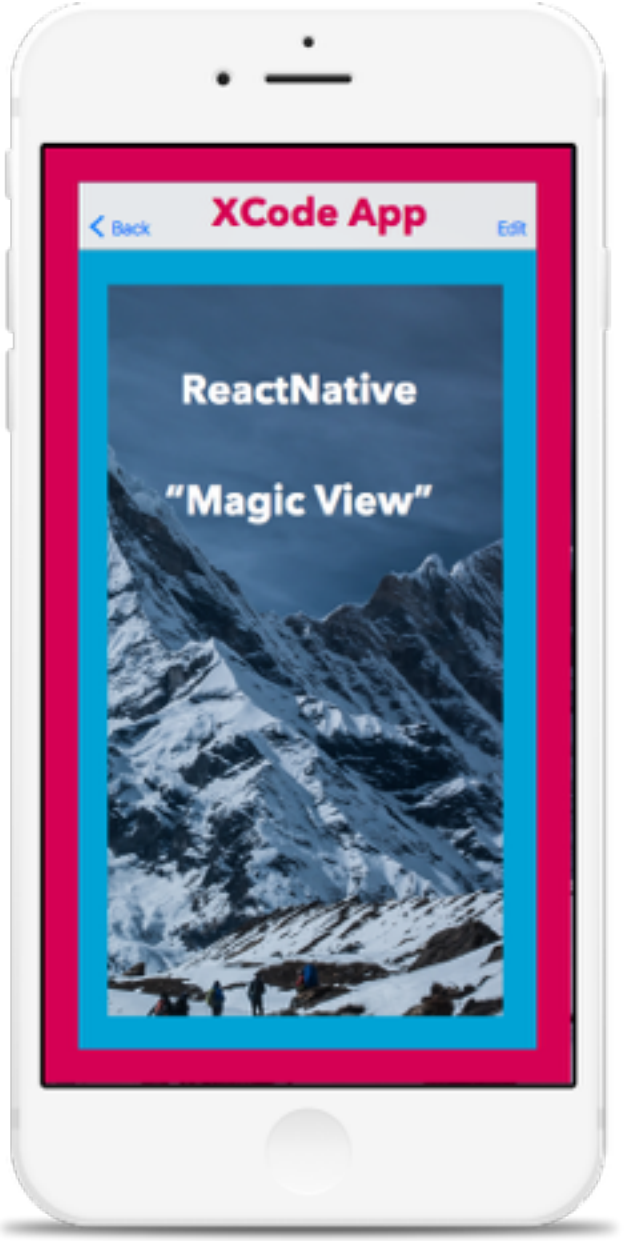
```
import React, {
  View,
  Text,
  StyleSheet,
  Component,
} from 'react-native';

import {
  loadApp,
} from '../actions/appActions';

import { connect } from 'react-redux/native';

class App extends Component {
  componentDidMount(){
    setTimeout(function() {
      this.props.loadApp({message: 'LOADED!'});
    }.bind(this), 500)
  }

  render() {
    let { appName, loaded } = this.props;
    return (
      <View style={styles.container}>
        <Text style={styles.welcome}>
          { appName }
        </Text>
        <Text>
```

→

React Native

node JS

→



*** - its simplified mental map of React Native mechanism.**

what is really under the hood you can read on

http://www.slideshare.net/ModusJesus/intro-to-react-native

MICHAŁ TABERSKI
michaltaberski.com

RN approach profits:

- you can use JavaScript for NATIVE apps for iOS and Android (cool for you, and your boss)
- you can design UI with consistent technologies you are already familiar with (Flexbox, and CSS subset)
- animations and gestures API is made with touchable display in mind

1/3

RN approach profits:

- a lot of components and logic can be shared between platforms
- you don’t need to recompile app after every code change
- a lot of good JS code can be resued on native apps (Redux, Immutable.js, test frameworks)

2/3

RN approach profits:

- React Native doesn’t enforce you to write entire app with this technology, so you can slowly embed RN to already written native app

- Sweet things for developers:

  ★ very good, descriptive error messaging

  ★ Chrome debugger

so its time to start…

QUICK START

- if you know already React you are almost done
- React Native is not HTML, but there are corresponding components:
    - ★ for block elements instead of **`<div/>`** use **`<View/>`**
    - ★ for inline elements instead of **`<span/>`** use **`<Text/>`**
    - ★ for images instead of **`<img/>`** use **`<Image/>`**
- CSS is limited, so check out the documentation before
- model your layout with flexbox

1/2

QUICK START

- Make sure you have XCode, and Node.JS on the board
- Install ReactNative CLI

```
$ npm install -g react-native-cli
```

- Init new project, and start to code

```
$ react-native init AwesomeProject
```

DEMO

Thanks!

feedback and contact
michal@michaltaberski.com