SAM OLLASON

Seccl

React Native and native iOS: my experience

OVERVIEW

Refresher on React Native:

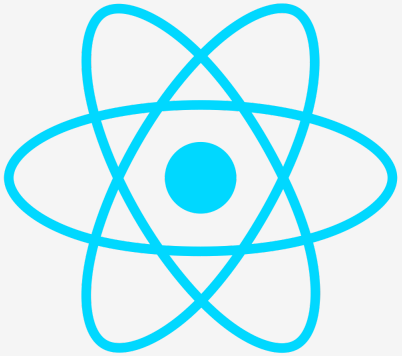
What it is and how it works

My Experiences:

Native iOS v React Native v React for the web

React Native - an overview

React Native



React Native

- React: JavaScript library for building UIs
- React Native: JavaScript library for building cross-platform mobile apps
- Write once, run anywhere (almost)

REACT NATIVE

How it works - rendering elements

- Same approach as React: UI components

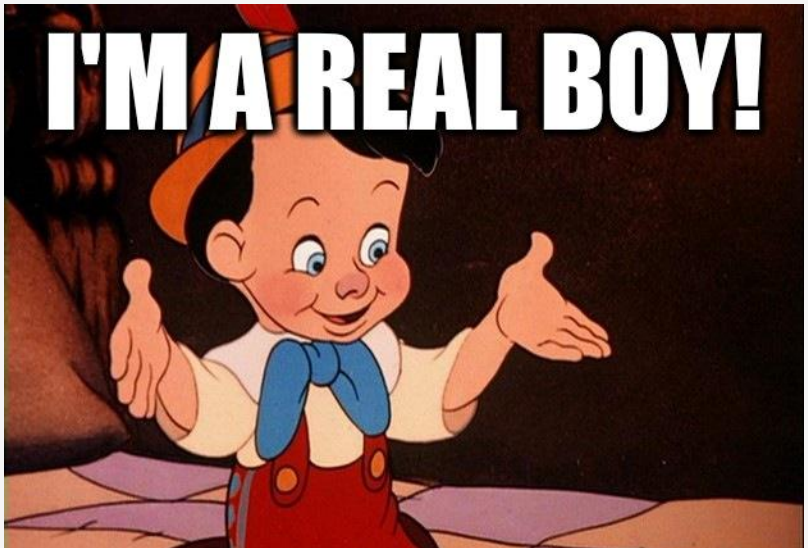
⋆ **Takeaway:** Same code (almost) as React for web projects

REACT NATIVE

A real native app

- Generates actual native UI modules

⋆ **Takeaway:** User doesn't know its not a true native app!



COMPARISON AREAS

- Language
- Tooling and Infrastructure
- UI Styling
- Connecting UI elements to logic
- Project Roadmap
- Deploying
- Bugs and support

Language

LANGUAGE

React Native

- Writing in React ... for native apps
- 'Almost' identical, so shallow learning curve
- Some React Native pre-made components easier to work with than React
- JavaScript under the surface

LANGUAGE

Native iOS App

- Swift (or Objective-C...)
- Type checking, type inference
- Easy to learn to start with, but then some challenges

LANGUAGE

Conclusion

- Swift > React Native (JavaScript)
- Swift > React for web (JavaScript)
- React Native > React for web
- ⭑ **Swift has a best developer experience**

Tooling and Infrastructure

TOOLING

Native iOS

- IDE: Xcode, Emulators: Xcode
- Swift for iOS apps compiled
- Slow and frustrating for quick iterations

TOOLING

React Native

- IDE: WebStorm, Emulators: Xcode
- Hot reloading

TOOLING

Conclusion

- Same emulator experience
- Hot reloading beats compilation

★ React native > Native iOS

★ React native same as React for the web

UI Styling

UI STYLING

Native iOS

- Xcode Interface Builder
- Initially enjoyed the fine-grained control
- In the end found cluttered and frustrating

pomodoroPlus | Build pomodoroPlus: **Succeeded** | 14/10/2019 at 13:55

- Home View Controller Scene
- Timer View Controller Scene
- Photo View Controller Scene
  - Photo View Controller
    - View
      - Safe Area
      - Home
      - Image View
  - First Responder
  - Exit
  - Unwind segue to "unwindToHome...
- T Scene
- Navigation Controller Scene

View as: iPhone 8 (wC hR)

```swift
//
//  TimerViewController.swift
//  pomodoroPlus
//
//  Created by Samuel Ollason on 23/11/2018.
//  Copyright © 2018 Samuel Ollason. All rights reserved.
//

import UIKit

class TimerViewController: UIViewController {

    // In hours
    var selectedTime = 00
    var time         = 999
    var seconds      = 60 //This variable will hold a starting value of seconds. It could be any amount above 0.

    var timer          = Timer()
    var isTimerRunning = false //This will be used to make sure only one timer is created at a time.
    var pauseActive    = false

    @IBOutlet weak var timeRemaningLabel: UILabel!
    @IBOutlet weak var pauseButton: UIButton!
    @IBOutlet weak var resumeButton: UIButton!
    @IBOutlet weak var resetButton: UIButton!
    @IBOutlet weak var pictureButton: UIButton!

    @IBAction func pauseButtonTapped(_ sender: Any) {
        if pauseButton.isEnabled {
            // If here then user wants to pause timer from running
            // so we stop timer and mark that the user has paused
            // countdown
            timer.invalidate()

            resumeButton.isEnabled = true;
```

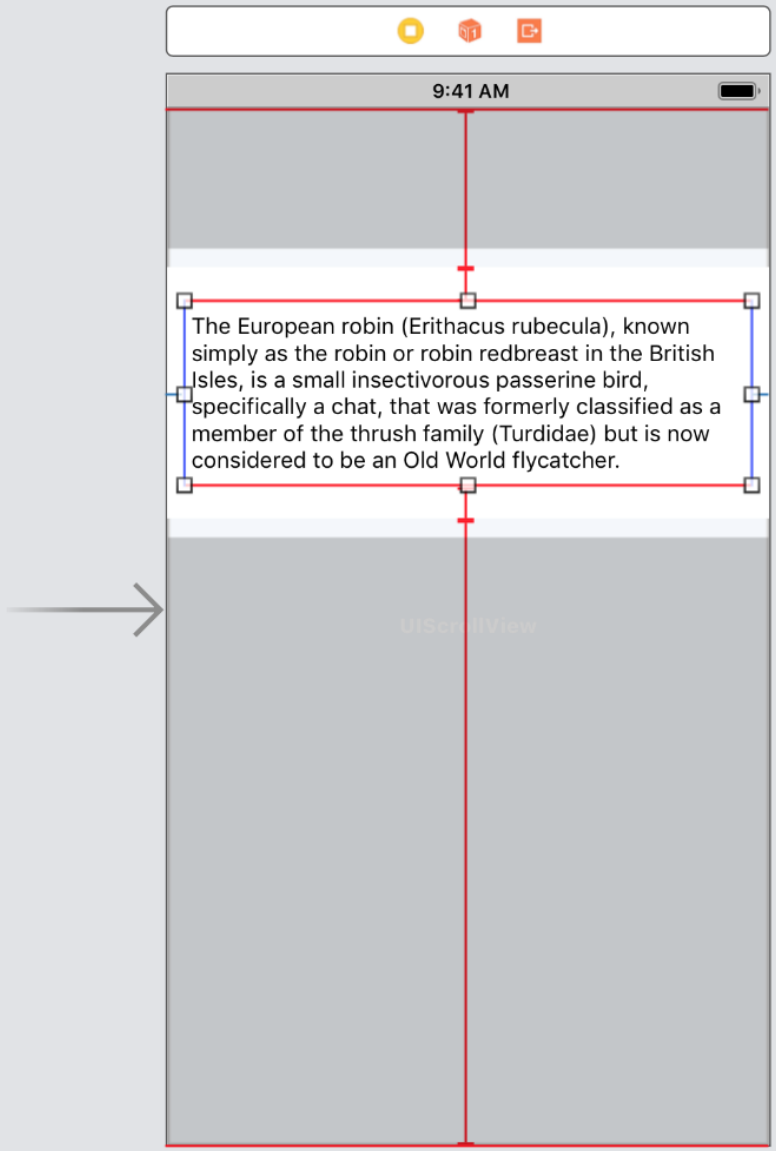
yout-With-ScrollView-and-TextView > A...w > ... > ... > V...e > V...r > V... > S...w > View > Text View

cene

a
ew

xt View
nstraints
bottom = Text View.bottom...
trailing = Text View.trailing...
Text View.top = top + 20
Text View.leading = leading...
raints
ew.top = top + 100
ttom = View.bottom + 419
ew.leading = leading
iling = View.trailing
ew.width = width
nts
r

ry Point



Return Key
Auto
Secu

Scroll View

Indicators Defau
Show
Show

Scrolling Scro
Pagi
Dire

Bounce Bou
Bou
Bou
Bou

Zoom
Mi

Content Touch Dela
Can

Keyboard Do no

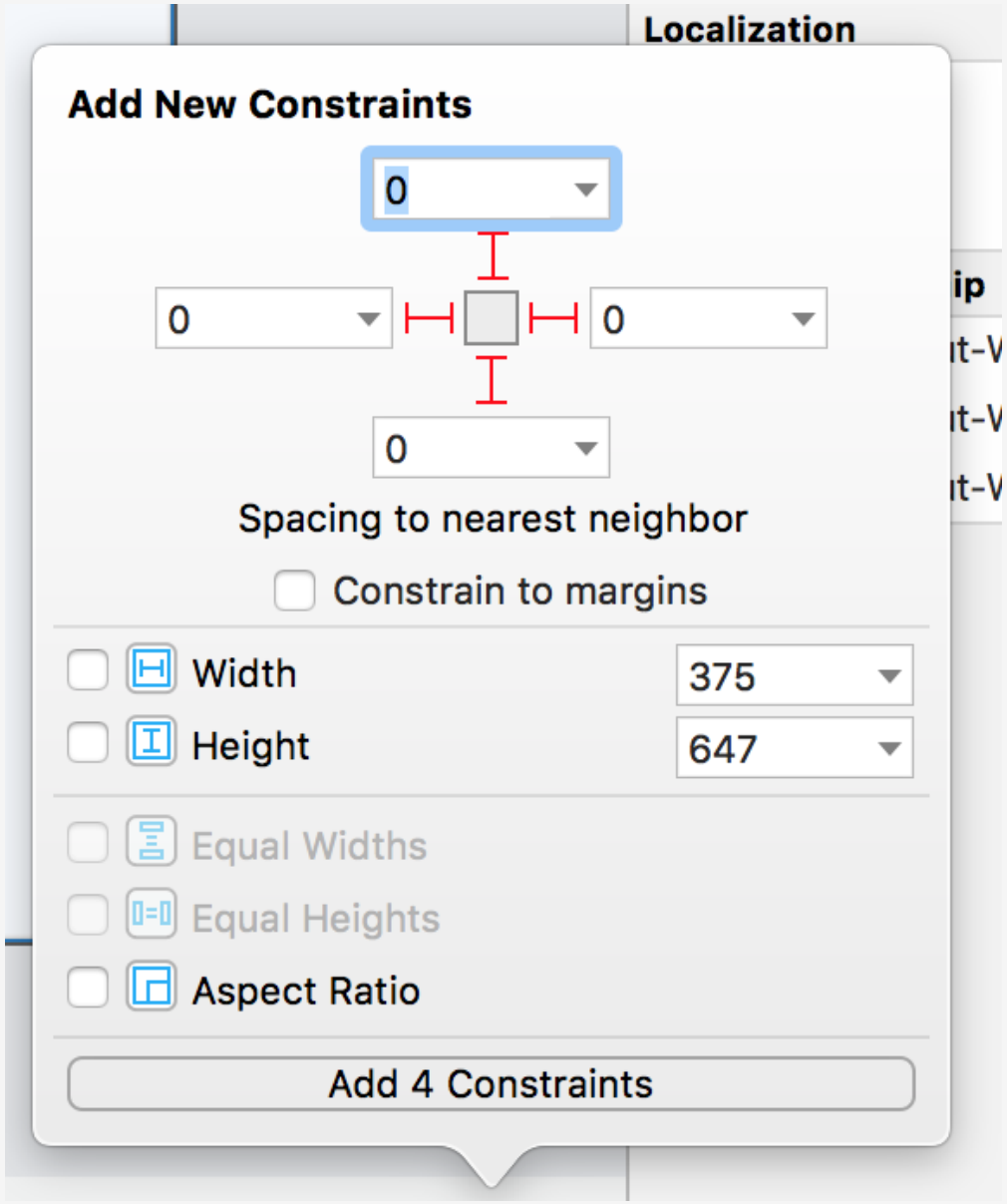
View

Content Mode Scale
Semantic Unspe
Tag

Interaction User
Mult

Alpha
Background
Tint

Drawing Opa
Hido


Localization
Add New Constraints
0
0
0
0
Spacing to nearest neighbor
Constrain to margins
Width
375
Height
647
Equal Widths
Equal Heights
Aspect Ratio
Add 4 Constraints

UI STYLING

React Native

- Programmatically style w/ FlexBox + CSS
- Very similar to React
- Handles resizing automatically

```
return(
    <ScrollView>
        <View style={styles.weekContainer}>
            {forecast.map((point) => {
                const day = getDay(point.dt_txt);
                const uri = `http://openweathermap.org/img/w/${point.weather[0].icon}.png`;

                return <View style={styles.weekdayContainer} key={point.dt_txt}>
                    <View style={styles.dateTime}>
                        <Text style={{fontSize: 15, color: textColour}}>{day}</Text>
                    </View>
                    <View style={styles.weather}>
                        <Image
                            source={{uri: uri}}
                            style={styles.icon}
                        />
                    </View>

                    <View style={styles.tempMax}>
                        <Text style={{fontSize: 15, fontWeight: 'bold', color: textColour}}>{point.main.temp_max}</Text>
                    </View>

                    <View style={styles.tempMin}>
                        <Text style={{fontSize: 15, color: textColour}}>{point.main.temp_min}</Text>
                    </View>
                </View>
            })}
        </View>
    </ScrollView>
)


t styles = StyleSheet.create({
weekContainer: {
    flex: 1,
    justifyContent: 'center',
},

weekdayContainer: {
    flexDirection: 'row',
    alignItems: 'center',
    paddingLeft: 30,
    paddingRight: 30,
    padding: 10,
    margin: 1,
},
dateTime: {
    flex: 2,
```

UI STYLING

Conclusion

- React Native the same as React for the web
- React Native > Native iOS
- ★ React native the easiest

UI and logic

UI AND LOGIC

Native iOS

- MVC paradigm
- View: storyboard area
- Controller: Swift source file
- Make sure appropriate files open/closed
- Make sure 'connections' deleted properly

pomodoroPlus | Build pomodoroPlus: **Succeeded** | 14/10/2019 at 13:55

- Home View Controller Scene
- Timer View Controller Scene
- Photo View Controller Scene
  - Photo View Controller
    - View
      - Safe Area
      - Home
      - Image View
  - First Responder
  - Exit
  - Unwind segue to "unwindToHome...
- T Scene
- Navigation Controller Scene

View as: iPhone 8 (wC hR)

```swift
//
//  TimerViewController.swift
//  pomodoroPlus
//
//  Created by Samuel Ollason on 23/11/2018.
//  Copyright © 2018 Samuel Ollason. All rights reserved.
//

import UIKit

class TimerViewController: UIViewController {

    // In hours
    var selectedTime = 00
    var time         = 999
    var seconds      = 60 //This variable will hold a starting value of seconds. It could be any amount above 0.

    var timer          = Timer()
    var isTimerRunning = false //This will be used to make sure only one timer is created at a time.
    var pauseActive    = false

    @IBOutlet weak var timeRemaningLabel: UILabel!
    @IBOutlet weak var pauseButton: UIButton!
    @IBOutlet weak var resumeButton: UIButton!
    @IBOutlet weak var resetButton: UIButton!
    @IBOutlet weak var pictureButton: UIButton!

    @IBAction func pauseButtonTapped(_ sender: Any) {
        if pauseButton.isEnabled {
            // If here then user wants to pause timer from running
            // so we stop timer and mark that the user has paused
            // countdown
            timer.invalidate()

            resumeButton.isEnabled = true;
```

UI AND LOGIC

React Native

- Markup and logic all in one file
- Markup with JSX
- JSX: *"combining markup and JavaScript"*

ctacle-boilerplate 〉 presentation 〉 index.js

Add Configuration...

index.js

```
import createTheme from 'spectacle/lib/themes/default';

const images = {
    formidagon: require('../assets/formidable-logo.svg'),
    goodWork: require('../assets/good-work.gif'),
    bridge: require('../assets/bridge.png'),
    XcodeSShot: require('../assets/XcodeSShot.png'),
    WebStormSShot: require('../assets/WebStormSShot.png'),
    pino: require('../assets/pino.jpg'),
    InterfaceBuilderStyling1: require('../assets/InterfaceBuilderStyling1.png'),
    InterfaceBuilderStyling2: require('../assets/InterfaceBuilderStyling2.png'),
    FlexBoxStyling: require('../assets/FlexboxStyling.png'),
    RNLogo: require('../assets/react-native-logo.png'),
    sponsor1: require('../assets/sponsor1.png'),
    sponsor2: require('../assets/sponsor2.png'),
    jsxExample: require('../assets/jsxExample.png'),


    firstSlide: require('../assets/firstSlide.png'),
}

// Require CSS
require('normalize.css');

const theme = createTheme(
    args: {
        primary: 'whitesmoke',
        secondary: '#1F2022',
        tertiary: '#03A9FC',
        quaternary: '#CECECE'
    },
    {
        primary: 'Montserrat',
        secondary: 'Helvetica'
    }
);

export default class Presentation extends React.Component {
    render() {
        return (
            <Deck
                transition={['zoom', 'slide']}
                transitionDuration={500}
                theme={theme}
            >

                <Slide
                    transition={['slide']}
                    bgImage={images.firstSlide}
                    // bgDarken={0.75}
                >
```

images

Version Control Terminal

s: WebStorm is ready to update. (25 minutes ago)

45:53 LF UTF-8 4 spaces

UI AND LOGIC

Conclusion

- ★ React Native > native iOS
- ★ React Native === React for the web

Project Roadmap

PROJECT ROADMAP

React Native

- Two choices: Expo CLI vs 'React Native CLI'
- Not always clear at which to pick
- Lots of community approaches to projects

PROJECT ROADMAP

Native iOS

- Clearly linear learning path
- *The* Apple way of doing things

PROJECT ROADMAP

Conclusion

- ★ Preferred guided nature of native iOS

Deploying

React Native: React Native CLI

- Open Xcode project - then its the same as a native release

React Native: Expo CLI

Deployment

- Bundle, upload with Apple desktop tool
- App is agnostic at and beyond this point

To publish to others with Expo app

- Over the air updates - share demos with others

DEPLOYING

Conclusion

- React Native CLI === Native iOS
- Expo CLI > Native iOS (from what I've read)

Bugs and support

BUGS AND SUPPORT

React Native

- Lots of moving parts
- Breaking changes between versions

BUGS AND SUPPORT

React

- Easier than React Native to find solutions as bigger community
- Still lots of moving parts

BUGS AND SUPPORT

Native iOS

- Easy to find solution
- Often clear if it was best practice or or not

BUGS AND SUPPORT

Conclusion

- ★ Native iOS > React for the web
- ★ React for the web > React Native

OVERALL

Language:	Native iOS (Swift)
Tooling and Infrastructure:	React Native
UI Styling:	React Native
UI connecting to Logic:	React Native
Project Roadmap:	Native iOS
Deploying:	React Native
Support:	Native iOS

RESOURCES

- Article by Sam Ollason on Medium
- GitHub
- Slides
- Created with Spectacle ... using React!!!

Thanks to our headline sponsor, category sponsors and partners

Cognisess

Recognise talent. Enable potential.

aws

SPACE IN BATH

storm

BRISTOL+BATH CREATIVE R+D

ALTRAN

twiDAQ

onesub

INVEST IN BATH

BATH Business Improvement District

bathdigitalfestival.co.uk

@

...and to our festival supporters!

EVIDENT

PURE PLANET

resolution

mud

Rocketmakers

REAL TIME CONSULTANTS

Claritum

UNIVERSITY OF BATH

royds withy king
lawyers ahead of the curve

WE DISCOVER. WE GROW
Girlguiding
Bath Division

ITEC

Software SOLVED

Mayden Academy

RIVIAM
digital care

datasharp | integrated communications

bathdigitalfestival.co.uk

@