Course Outline

React Native

Course Summary

Description

React Native is used by organizations worldwide to create cross-platform phone/tablet apps with one code base. This means that you can write code easily that will run on iPhones, iPads, and Android phones and tablets without having to rewrite it in two or more languages -- just one. And that one language is easy to learn if you know React and JavaScript because it is React and JavaScript! Furthermore, these apps run natively and can be deployed to the Apple AppStore or the Google Play store, not web apps that run in some adapter or translator. So they're faster and more reliable.

In this class you'll learn to use React Native to create cross-platform native apps quickly and easily with a 50/50 mix of lecture and real-world labs. You'll start from scratch and build up to a comprehensive app which uses modern techniques and best practices to consume RESTful data from a NodeJS/Express server and present it to the user in a multi-screen, interactive app.

Objectives

After taking this course, students will be able to:

- Create cross-platform iOS and Android apps
- Set up a device emulator on your laptop and deploy apps to it
- Use expo to create and run iOS and Android apps
- Explain the architecture of a device app
- Apply the most useful React Native components
- Write app code that works differently on the different platforms
- Use flexbox on devices to control the layout of your apps
- Style your app efficiently using best practices
- Use stack navigators, drawer navigators and tab navigators to change app scenes
- Consume RESTful data in a handheld device and present it to the user

Topics

- Course Overview
- Hello React Native
- React and Redux reviews (when needed)
- The Development Process
- Single-value Controls
- Platform-specific Development
- Layout Components
- Flexbox for Native Layouts
- Styling React Native Apps
- Navigation
- Ajax in React Native
- List Components
- Touchables and Buttons

Audience

This course is designed for seasoned developers who want to create iOS and Android apps.

Prerequisites

Before taking this course, you should have familiarity with React and a very strong grasp of advanced JavaScript. Please ask about our JavaScript and React courses which will prepare you for this course.

Duration

Four days

Course Outline

React Native

Course Outline

I. Course Overview

II. Hello React Native

A. What is React Native?
B. What does it do for us? Why choose it?
C. Pros and cons
D. Architecture
E. Sharing with web projects
F. What React Native code looks like
G. Leveraging your React knowledge

III. React and Redux reviews (when needed)

A. Redux reminder
B. Reducers, actions, state, store, and middleware
C. React reminder
D. SFCs vs class-based components
E. Composition
F. JSX structure and rules
G. props
H. state
I. Controlled and uncontrolled components
J. Virtual DOM vs the real DOM

IV. The Development Process

A. Where do I even start?
B. react-native vs. create-react-native-app
C. Which is better for given situations
D. The React Native team's recommendations
E. What is expo?
F. Creating a new React Native app
G. How to run it on a tethered device
H. How to run it on a wireless device
I. How to run it in an Android emulator
J. How to run it on an iOS simulator
K. Debugging in a browser window
L. Logging, breakpoints, stepping through
M. YellowBoxes and RedBoxes

V. Single-value Controls

A. Components overview
B. Categories of components
C. Text
D. Text props and events
E. TextInput
F. props and events and the event object
G. Image
H. Differences between HTML and React Native images
I. Reserving space for them
J. Local images vs remote images
K. resizeMode

VI. Platform-specific Development

A. How can we develop differently on the different platforms?
B. Why would we ever do this?
C. Technical roadblocks
D. The DatePicker - iOS vs Android
E. Using the Platform module

VII. Layout Components

A. Components review
B. View
C. SafeAreaView
D. ScrollView
E. Pinch-to-zoom
F. KeyboardAvoidingView
G. How to create modal views
H. Controlling the OS's status bar

VIII. Flexbox for Native Layouts

A. Why flexbox?
B. Where it came from
C. Flexbox on the web is NOT flexbox on native
D. Containers and items
E. flexDirection
F. flexBasis vs width/height
G. flexShrink, flexGrow
H. The flex shorthand
I. justifyContent and alignContent
J. flexWrap

IX. Styling React Native Apps

A. How React Native styles differ from CSS
B. How to apply styles
C. How to control style inheritance
D. Style arrays
E. Four methods of defining styles
F. Common properties

Course Outline (cont'd)

G. Cross-platform fonts
H. Conditional and programmatic styles

X. Navigation

A. What is navigation, really?
B. How to get React Navigation
C. The three types of navigators
D. StackNavigator
E. Routing object
F. Navigation config
G. How to pass params when navigating
H. TabNavigators
I. Three types of TabNavigators
J. How to set icons
K. DrawerNavigator
L. Examples and demos

XI. Ajax in React Native

A. Why it must be different on a device
B. The fetch API
C. How to show a loading indicator
D. How to make requests and populate affordances
E. Security in a native environment

XII. List Components

A. Components review
B. Pickers
C. FlatList
D. SectionList

XIII. Touchables and Buttons

A. The Button API
B. Button events and props
C. Why touchables?
D. TouchableWithoutFeedback
E. TouchableNativeFeedback
F. TouchableOpacity
G. TouchableHighlight
H. How to disable a touchable