

**Comprehensive React Native**

**Course Number:** RCT-102  
**Duration:** 4 days

**Overview**

[React Native](https://facebook.github.io/react-native/) is used by organizations worldwide to create cross-platform phone/tablet apps with one codebase. 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, making them faster and more reliable.

In this React Native training class, attendees learn how to use React Native to create cross-platform native apps quickly and easily. Students start from scratch and build up to a comprehensive app that uses modern techniques. In addition, participants learn best practices to consume RESTful data from a NodeJS/Express server and present it to the user in a multi-screen, interactive app.

**Prerequisites**

All attendees must have a very strong grasp of React, Redux and advanced JavaScript. If your group doesn’t yet have this experience, we could readily add one or two days to the beginning of your course to appropriately prepare them.

**Materials**

All React Native training attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

* Every attendee must have a Mac computer, with a current version of macOS
* Xcode (needed for the iOS emulator)
* Android Studio (needed for the Android Virtual Device)
* Node.js (which includes npm), and the ability to install npm modules
* Web browsers that you want to test
* A text editor, such as Visual Studio Code
* Other free software and lab files that Accelebrate will specify

**Objectives**

* 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

**Outline**

* Course Overview
* Hello React Native
  + What is React Native?
  + What does it do for us? Why choose it?
  + Pros and cons
  + Architecture
  + Sharing with web projects
  + What React Native code looks like
  + Leveraging your React knowledge
* React and Redux reviews (when needed)
  + Redux reminder
  + Reducers, actions, state, store, and middleware
  + React reminder
  + SFCs vs class-based components
  + Composition
  + JSX structure and rules
  + props
  + state
  + Controlled and uncontrolled components
  + Virtual DOM vs the real DOM
* The Development Process
  + Where do I even start?
  + react-native vs. create-react-native-app
  + Which is better for given situations
  + The React Native team's recommendations
  + What is expo?
  + Creating a new React Native app
  + How to run it on a tethered device
  + How to run it on a wireless device
  + How to run it in an Android emulator
  + How to run it on an iOS simulator
  + Debugging in a browser window
  + Logging, breakpoints, stepping through
  + YellowBoxes and RedBoxes
* Single-value Controls
  + Components overview
  + Categories of components
  + Text
  + Text props and events
  + TextInput
  + props and events and the event object
  + Image
  + Differences between HTML and React Native images
  + Reserving space for them
  + Local images vs remote images
  + resizeMode
* Platform-specific Development
  + How can we develop differently on the different platforms?
  + Why would we ever do this?
  + Technical roadblocks
  + The DatePicker - iOS vs Android
  + Using the Platform module
* Layout Components
  + Components review
  + View
  + SafeAreaView
  + ScrollView
  + Pinch-to-zoom
  + KeyboardAvoidingView
  + How to create modal views
  + Controlling the OS's status bar
* Flexbox for Native Layouts
  + Why flexbox?
  + Where it came from
  + Flexbox on the web is NOT flexbox on native
  + Containers and items
  + flexDirection
  + flexBasis vs width/height
  + flexShrink, flexGrow
  + The flex shorthand
  + justifyContent and alignContent
  + flexWrap
* Styling React Native Apps
  + How React Native styles differ from CSS
  + How to apply styles
  + How to control style inheritance
  + Style arrays
  + Four methods of defining styles
  + Common properties
  + Cross-platform fonts
  + Conditional and programmatic styles
* Navigation
  + What is navigation, really?
  + How to get React Navigation
  + The three types of navigators
  + StackNavigator
  + Routing object
  + Navigation config
  + How to pass params when navigating
  + TabNavigators
  + Three types of TabNavigators
  + How to set icons
  + DrawerNavigator
  + Examples and demos
* Ajax in React Native
  + Why it must be different on a device
  + The fetch API
  + How to show a loading indicator
  + How to make requests and populate affordances
  + Security in a native environment
* List Components
  + Components review
  + Pickers
  + FlatList
  + SectionList
* Touchables and Buttons
  + The Button API
  + Button events and props
  + Why touchables?
  + TouchableWithoutFeedback
  + TouchableNativeFeedback
  + TouchableOpacity
  + TouchableHighlight
  + How to disable a touchable
* Conclusion