

react native training

React Native Fundamentals

The following document contains the outline of an example *React Native Fundamentals* workshop

For more information, contact us at corporatetraining@reactnative.training

1/ Warm up

- Welcome
- Introductions
- Discuss React Native experience and preconceptions of attendees
- Present agenda

2/ Introduction

In this section, we make a brief introduction to React Native. We set up machines for all the people that haven't done that yet. By the end, attendees can create a new project, run it and understand tools available.

- Describe React Native:
 - Philosophy
 - Available platforms
 - Available APIs
 - Available components
 - Bridge
 - Prerequisites
 - Installation review (show installation instructions)
- Create an app with `react-native init` (mention Expo and Create React Native App command)
- Use `react-native run-ios/android` to run the app
- Describe `react-native-packager`
- Demonstrate `Developer Menu`
- Small exercises to get people familiar

3/ Back to JavaScript

In this section, we go through latest ES6/ES7 features that were added, like import / export syntax or fat arrow functions. We also cover basics of React (or just refresh them for people already experienced), like component lifecycle and/or JSX. All these are run and live tested on an app set up in the previous section.

- List all `ES6` and `ES7` features (one per slide, w/o going into details)
- Explain how the code is transpiled
- Explain JSX
- React Component lifecycle
- Open discussion about other features
 - Show how to overload Babel config
- A couple of exercises to make everyone comfortable with React / JavaScript

4/ Core components

In this section, we cover the basic components, like `>` & `>`.

- Describe all core components and their `props`
- A couple of exercises based on default `init` template:
 - Add few buttons and test out handlers (with `bind` preferably)
 - Add a bunch of different `>`s

5/ Styling

In this section, we, describe `StyleSheet` API, how it implements flexbox and how's that different from CSS.

- Describe StyleSheet, what are available values, how it works
- Demonstrate `Flexbox`
- Mention that there are UI kits, but we are not going to use them as it's too advanced for now
- Mention that there are cross-platform styling techniques, like styled-components one can use
- A bunch of exercises to get attendees more familiar with the styling, esp.:
 - Flexbox and its properties - implement column/grid layout as presented on a slide
 - `<Text />>` number of lines

6/ Navigation

In this section, we demonstrate basic concepts of navigation. We also brief attendees into how's JS navigation different from fully native one. We list available alternatives as well what we will use throughout today.

- List available navigation solutions
- Brief readers into the one that was selected
- Make them aware of the API and how to think of the route hierarchy
- A couple of exercises working on current `init` template, e.g convert app to a stack, so we can push a new route

7/ Lists

In this section, we describe lists and why they are so important in React Native. We describe available alternatives and list how's `<ScrollView />` different than `<FlatList />` (and `<SectionList />`) and when to use each.

- List available scroll solutions
- Explain when to use each
- Describe performance optimizations
- Warn about common pitfalls / issues
- As a demo task, one can create a view that has a list of contacts and each of them can be tapped to move to a new screen (with details)

8/ Cross Platform APIs

We will discuss and implement the most used React Native APIs that work cross platform

9/ iOS specific APIs

We will discuss and implement the most used iOS specific React Native APIs

10 / Android specific APIs

We will discuss and implement the most used Android specific React Native APIs

11/ Persistence

In this section, we describe how persistence is done with React Native and how it can be achieved using other technologies.

- Demonstrate persistence using AsyncStorage
- Exercise attendees to persist stuff (literal, more advanced JSON)
- Demonstrate available APIs, like `multiSet` and when it's better to use what
- Mention other tools like `realm`

12/ Animations

In this section, we will examine different approaches to animating elements within the app. Specifically, we will check `LayoutAnimation` API and the better - `Animated`. We will briefly talk about performance concerns as well.

- Introduce LayoutAnimation
- Challenge attendees with simple animations
- Ask if they feel happy with it?
- Introduce `Animated` as a general solution to the problem
- Encourage them to animate few things on screen
- Perf. wise - mention native driver

- Do more tasks!

13/ Working with HTTP, network requests, and accessing restful services

Here we look at using both the fetch API as well as Axios for fetching and sending data, and using the returned data to update our application UI.

14/ Data Architecture

In this section, we cover both MobX and Redux and talk about how and why they are useful in a React Native app

- Introduce Redux
- Set up a basic redux implementation
- Add Thunk middleware for asynchronous actions
- Fetching data and updating our redux store
- Discuss other asynchronous libraries such as Saga and Redux Promise Middleware
- Introduce MobX
- Set up a basic MobX implementation

15/ Application Deployment

In this section, we discuss various settings and configurations that the developer must use and understand to deploy the app to both the Google Play store as well as the Apple App store. We also discuss continuous deployment with CodePush.