Big Nerd Ranch

Advanced Android

Training Syllabus

Course Overview

This 5-day class is intended to help intro-level Android developers get the skills needed to work on production-ready applications. They will learn industry best practices and libraries that will enable them to create well designed code that is easy to maintain. They will also learn how to test the different parts of their applications, as well as using performance testing tools to identify potential issues before their users.

Who Should Take This Course

- Current Android developers with 6+ months of development experience on the platform. If you are comfortable with creating new projects and building a few screens but want to learn the practices and tools the industry uses for applications then this class is a good fit for you.
- Students should be familiar with standard Android components and how they work together. Activities and Fragments, and their lifecycles, layout inflation, Services, and background threading should be familiar topics to potential students.

Syllabus

Model Architecture

- Configure your application for dependency injection using Dagger 2 so your application components can get the dependencies they need without unnecessary coupling.
- Explore the usage of functional-reactive programming with RxJava and RxAndroid.

- Add Data Binding to your application to help implement the Model-View-ViewModel architecture pattern.
- Understand View Binding and how it relates to Data Binding.
- Reap the benefits of dependency injection by easily testing your model layer objects.

Advanced Networking

- Explore the Retrofit library and learn how it can reduce the network boilerplate in your applications.
- Learn how to authorize your application with a back-end server and make authenticated requests using OAuth
- Implement common patterns to deal with network exceptions and background threading.
- Test your networking code using both unit and integration tests for a full test suite.

Android Views

- Create a fully custom list item view to gain as much performance as possible from your UI.
- Support accessibility in your custom views.
- Apply a notification style to change how your data is displayed to the user depending on the type of data.
- Test the UI of your application by using the Espresso framework to write your integration tests.

Performance Testing

- Utilize the built-in testing tools to identify view performance issues that can lead to dropped frames and suboptimal user experience.
- Identify memory leaks in your application as well as conditions that can lead to high battery drain and unhappy users.

Creating Builds

- Configure your application to produce different build flavors and identify how these can be used.
- Prepare your application for release in the Play store by configuring your application, signing it, and learning about the release process.