

Android Expert Development

•Android Design

- Android design key principles
- Design recommendations for great applications

•Using Model View Presenter in Android

- The Model View Presenter pattern
- Using build flavors to increase testability
- Using dependency injection to increase testability

•Fragments deep dive

- Dynamic fragment replacement done right
- Using headless fragments

•Gradle deep dive

- Building product flavors
- Defining custom tasks

•Using XML view databinding

•Dependency injection with Dagger 2

- What is dependency injection
- Usage of Dagger 2 in Android
- apt compile hooks

•Efficient network communication

- Using image processing libraries
- Using OkHttp for HTTP requests
- Using Retrofit for REST clients
- Efficient Json parsing with Gson

•Building reactive Android applications with RxJava

- Overview of RxJava
- Using RxJava in Android
- Using RxJava as event system
- Combining RxJava with Retrofit

•Efficient list and grid handling

- Optimizing lists and grid handling with RecyclerView
- Image handling with image libraries like Picasso

•Custom and Compound Views and the Canvas API

- Custom Views
- Compound Views
- Canvas API
- Persisting View data
- Single touch
- Multi touch
- Gesture detection

•Background processing deep dive

- Asynchronous processing deep dive
- Headless Fragments
- Loader
- Android platform service
- Declaring own services
- Service and activity communication
- Outlook: AIDL and interprocess communication

•Tips and Tricks

- Solving common design problems
- Supporting several releases

•Outlook: Cloud connectivity

- Connecting to the Google Cloud
- Using Firebase as backend