Moderne Android App Entwicklung

Java User Group Saarland – 21.07.2016

David Kunzler

Über Mich

- Java-Developer @ DIaLOGIKa
- Nebenberuflich Android Entwickler
- Durchgängig Android Erfahrung seit 2012

- www.devland.de
- https://github.com/dkunzler
- https://www.google.com/profiles/dkunzler

Android Portfolio

- esperandro
  - Open-Source Library
  - 167 Stars
- Master Password for Android
  - Open-Source
  - Gefeatured auf heise.de und in c‘t
  - 4.4 Sterne im Play Store
  - 24.000 Downloads
  - Pro Version (30x raubkopiert ☺)
- Lockscreen Birthdays
  - Closed-Source
- App-Volume
  - unveröffentlicht

Was machen wir heute?


Thomas Darimont
@thomasdarimont
Folge ich
Random app idea "Code Safari" - show a block of code and ask whether it's elegant or not :)
Übersetzung anzeigen
RETWEET
1
GEFÄLLT
6
13:43 - 30. Juni 2016
1
6

Build

Build Tools (Cutting Edge)

- Android Studio 2.2 Preview
  - Instant Run (teilweise buggy)
  - Java 8 Support (mit Jack)
  - Layout Designer mit Constraint Layout
- Jack/Jill Compiler
- Preview Gradle Plugin

Jack (nicht production ready)

- Neuer Toolchain in einem Tool
  - Packaging
  - Shrinking
  - Obfuscation
  - Multidex
- Überspringt .class Generierung
- Kleinere APK Dateien
  - Erst ab gewisse Anzahl Klassen/Dateien
  - Jack Runtime in der APK benötigt
- Java 8 Features
- (Noch?) kein Support für projectlombok und evlt. andere Annotation Prozessoren

Nützliche Libraries

HTTP(S): Retrofit

- Interface-basiert
- Objekt-Serialisierung
- Get, Put, Post, Delete
- Synchrone und asynchrone Verwendung möglich

Dependency Injection

- Dagger 2
  - Dependency Graph komplett zur Compilezeit
  - Lohnt erst für größere Projekte
  - Aufwändig zu konfigurieren (kein Plug & Play)
  - Möglichkeit Dependencies in Unit Tests auszutauschen
- Keine ernsthafte weitere Alternative

SharedPreferences: esperandro

- Typsichere Preferences
- Aktionen zur Erleichterung des Handlings
- Defaults
- Möglichkeit der logischen Trennung durch Benamung

View Injection: Butterknife

- Ohne Casts Views injecten
- ViewHolder Pattern erleichtern
- Diverse Events direkt an Methoden binden
  - OnClick
  - OnFocus
  - OnCheckedChange
  - …

Datenbank

SQLite ORM Mapper

SugarORM

- Schnell aufgesetzt
- Leichtgewichtig
- Parent Objekte für App und Entities
- Schema Updates über plain SQL Skripte
- Query-Methoden am Entity

OrmLite

- Mächtig
- Erfordert Einarbeitung
- Reflection & Annotations
- Schema Updates im Code
- Query-Methoden an DAOs
- Tipp: Besser mit DI nutzen

Realm

- Neue Cross Platform Datenbank
- Unabhängig von Plattform SQLite
- Schneller als SQLite
- Fluent API
  - RealmResults<Dog> puppies = realm.where(Dog.class).lessThan("age", 2).findAll();
- Encryption Support
- Schema Updates im Code

UI

Moderne UI

- Google Support Libraries verwenden
  - Cards
  - Recyclerview
  - Toolbar
  - DrawerMenu
  - Snackbar
  - FloatingActionButton
- Animationen / Fading
- www.google.com/design studieren

Sonstige Hinweise

Best Practices

- Nicht den UI Thread blockieren
  - AsynTask oder Handler verwenden
- Ressourcenschonend programmieren
  - Wenige Threads
  - Wenig Reflection (langsam unter Android)
  - Wenige Services
  - BroadcastReceiver verwenden
- Rad nicht neu erfinden
  - Square
  - Github
  - Android Arsenal https://android-arsenal.com/
- Build aktuell halten
  - Lieber viele kleinere Schritte mit Android Studio und Build Tools als wenige große
  - Support Libs aktuell halten
  - Inzwischen relativ stabil und problemlos
- <Insert favorite Java Best Practice here>

Nützliche Apps

- Keyline Pushing
- Vysor
- SQLite Debugger (root)
- CatLog
- Demo Apps verschiedener UI Libs

Danke

Fragen/Diskussionen?

Resources

- https://sites.google.com/a/android.com/tools/tech-docs/jackandjill
- https://developer.android.com/topic/libraries/support-library/index.html
- https://dkunzler.github.io/esperandro/
- https://square.github.io/
- https://jakewharton.github.io/butterknife/
- https://google.github.io/dagger/
- https://material.google.com/
- https://realm.io/