

Einführung in die Android App Entwicklung

Philipp Deppenwiese

<https://github.com/zaolin/android-app-demo>

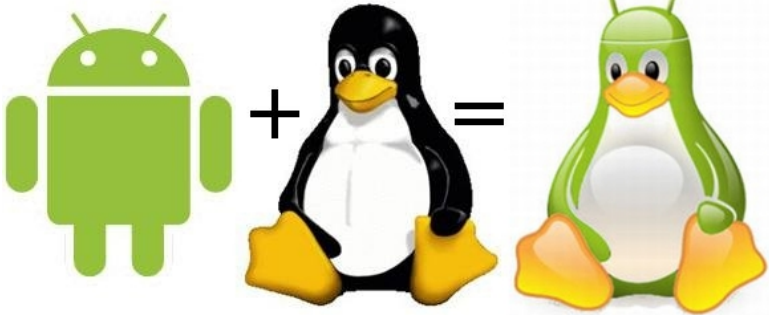
2. Juli 2015

Das Labor e.V.

Bochums Hackerspace

"Wir schaffen mit Technik Faszinierendes und Nützliches."

Android



Worum es in dem Vortrag nicht geht

- Appstore und Zertifizierung
- Versionskontrolle
- IDE Funktionalität
- Einbinden & Entwicklung von nativen Bibliotheken
- Reflections
- Tiefergreifende UI Entwicklung
- Netzwerk
- usw..

Weiterführend

<https://developer.android.com/guide/index.html>

Roter Faden

- API Level
- App Struktur, Manifest & Komponenten
- Framework, Ressourcen & UI
- Zugriffsrechte
- Inter-Prozess-Kommunikation
- Native Unterstützung
- Hardware Einrichten
- Workshop Part 1
- Workshop Part 2
- Workshop Part 3

API Level

API Level

- Das Android Framework welches eine Java Bibliothek darstellt besitzt eine Schnittstellen (API) Versionierung.
- Jede API Version ist von der System version abhängig.
- Android 5.1 == API Level 22
- Android 5.0 == API Level 21
- Android 4.4 == API Level 19

App Struktur, Manifest & Komponenten

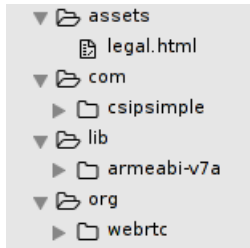
App Struktur

- Jede Android App ist eine Zip Datei !
- Diese enthält unter META-INF die zwei Manifest Dateien und ein PKCS7 Container worin sich Signatur und Zertifikat befinden.



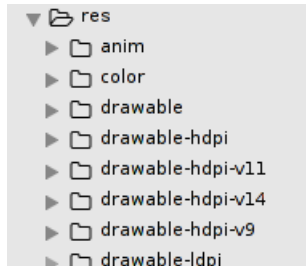
App Struktur

- Unter z.B. org. und com. befinden sich weitere App spezifisch Dateien (Lizenzen, README).
- lib enthält Architektur spezifische native Bibliotheken.
- assets ist ähnlich dem res Ordner enthält jedoch alle möglichen Dateien.



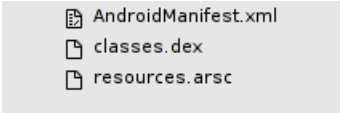
App Struktur

- In dem res Ordner finden sich sämtliche Ressourcen die UI spezifische Elemente wie Strings, Grafiken usw.



App Struktur

- Die AndroidManifest.xml definiert das Android Package Kit.
- Die resources.arsc ist eine Index Datei der Ressourcen.
- Die classes.dex enthält alle kompilierten Klassen als DEX Bytecode.



AndroidManifest.xml
classes.dex
resources.arsc

App Manifest

- Jedes Android Paket hat einen eindeutigen und einzigartigen Paketnamen !
- Das Manifest beschreibt alle Komponenten die zum Android Paket dazugehören.
- Ebenso finden sich hier die Zugriffsrechte die diese App benötigt. Diese werden einem bei der Installation der App angezeigt.
- Ebenfalls findet man zusätzliche Metadaten die zum Android Paket gehören.

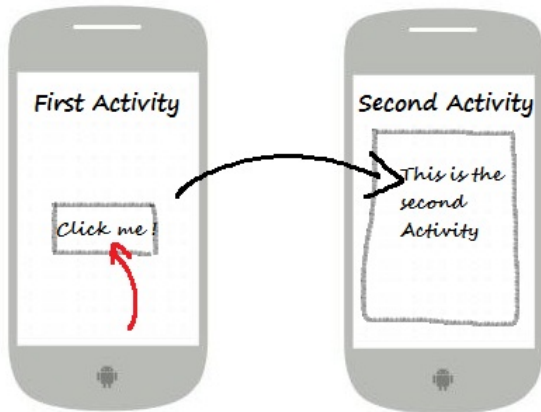
App Manifest

```
<?xml version='1.0' encoding='utf-8' standalone='no'?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:installLocation="auto" package="com.csipsimple">
  <supports-screens android:anyDensity="true" android:largeScreens="true" android:normalScreens="true" android:smallScreens="true" android:xlargeScreens="true"/>
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission android:name="android.permission.RECORD_AUDIO"/>
  <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
  <uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS"/>
  <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
  <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
  <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
  <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
  <uses-permission android:name="android.permission.READ_CONTACTS"/>
  <uses-permission android:name="android.permission.WRITE_CONTACTS"/>
  <uses-permission android:name="android.permission.CALL_PHONE"/>
  <uses-permission android:name="android.permission.WAKE_LOCK"/>
  <uses-permission android:name="android.permission.DISABLE_KEYGUARD"/>
  <uses-permission android:name="android.permission.VIBRATE"/>
  <uses-permission android:name="android.permission.BLUETOOTH"/>
  <uses-permission android:name="android.permission.READ_LOGS"/>
  <uses-permission android:name="android.permission.WRITE_CALL_LOG"/>
  <uses-permission android:name="android.permission.USE_SIP"/>
  <uses-permission android:name="android.permission.CONFIGURE_SIP"/>
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
  <uses-permission android:name="android.permission.CAMERA"/>
  <uses-permission android:name="android.permission.BROADCAST_STICKY"/>
  <uses-permission android:name="android.permission.READ_PROFILE"/>
  <uses-permission android:name="android.permission.WRITE_CALL_LOG"/>
  <permission android:description="@string/permdesc_useSip" android:label="@string/permlab_useSip" android:name="android.permission.USE_SIP" android:permissionGroup="android.permission-group.COST_MONEY" android:protectionLevel="dangerous"/>
  <permission android:description="@string/permdesc_configureSip" android:label="@string/permlab_configureSip" android:name="android.permission.CONFIGURE_SIP" android:permissionGroup="android.permission-group.COST_MONEY" android:protectionLevel="dangerous"/>
  <uses-feature android:name="android.hardware.wifi" android:required="false"/>
  <uses-feature android:name="android.hardware.telephony" android:required="false"/>
  <uses-feature android:name="android.hardware.microphone" android:required="true"/>
  <uses-feature android:name="android.hardware.touchscreen" android:required="false"/>
  <uses-feature android:name="android.hardware.bluetooth" android:required="false"/>
  <uses-feature android:name="android.hardware.screen.portrait" android:required="false"/>
  <uses-feature android:name="android.hardware.camera" android:required="false"/>
  <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
  <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
  <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
  <application android:allowBackup="true" android:backupAgent="com.csipsimple.backup.SipBackupAgent" android:hardwareAccelerated="true" android:icon="@drawable/ic_launcher_phone" android:label="@string/app_name">
    <meta-data android:name="app_type" android:value="release"/>
    <meta-data android:name="app_channel" android:value="armeabi-v7a"/>
    <meta-data android:name="com.google.android.backup.api_key" android:value="AEDPqrEAAAIL5x8KlvG-YbN4vJ4PN5y2JH6yQZ49yKbrKJAw"/>
    <meta-data android:name="com.sec.android.support.multiwindow" android:value="true"/>
    <activity android:label="@string/app_name" android:launchMode="singleTask" android:name="com.csipsimple.ui.SipHome" android:theme="@style/DarkTheme" android:uiOptions="splitActionBarWhenNarrow">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>
```

App Komponenten

- In der Android Welt existieren mehrere Komponenten die Inhalt einer Applikation sein können.
- Um eine Komponente verwenden zu können muss von dieser geerbt werden.
- Mehrere Komponenten können Teil einer Applikation sein.

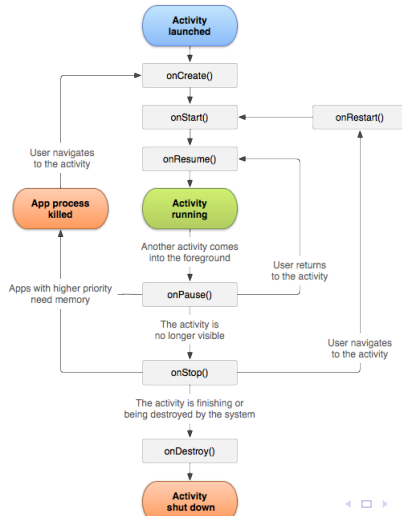
Activity



Activity

- Eine Klasse die einen Screen beschreibt in dem ein Benutzer Interaktionen tätigen kann.
- Diese Klasse besitzt einen Lifecycle wie alle Komponenten und kann durch Intents aufgerufen werden.
- Beim Start einer Android Applikation wird eine Hauptaktivität gestartet welche den ersten Screen anzeigt von dem aus weitere Activities gestartet werden können.

Activity



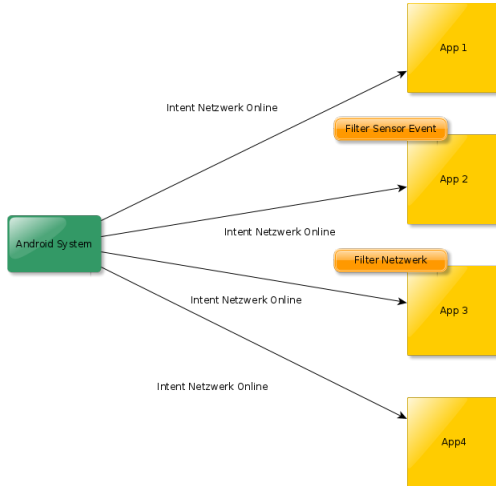
Service



Service

- Eine Service ist ein Hintergrunddienst der auch nach schlafen legen der Activity weiterhin aktiv bleibt.
- Services werden meistens via Interfaces aufgerufen die sie beim sogenannten ServiceManager registrieren können.
- Dadurch wird es möglich eine lokale oder auch Inter-Prozess-Kommunikation zwischen App oder Activity zu erstellen.
- Nebenbei Android selbst bietet eine Menge System Services an !

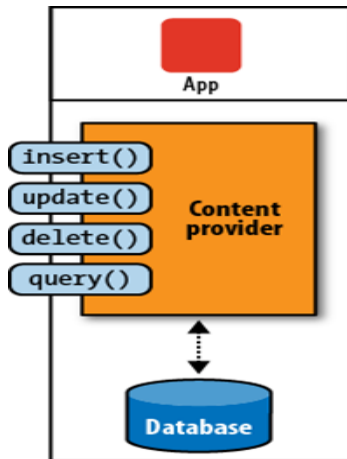
BroadcastReceiver



BroadcastReceiver

- Diese Klasse funktioniert Event basiert und nutzt den Intent IPC Mechanismus !
- Es können sämtliche System und App Intents gelesen werden.
- Jeder Receiver kann ein Filter definieren indem er nur bestimmte Intent Typen zulässt.

ContentProvider



ContentProvider

- Ist eine Art von Service welcher via Datenbanken Informationen abspeichert.
- Nutzen SQLite als Datenbank Backend.

Framework, Ressourcen & UI

Android Framework

- API level spezifisch.
- Eine Android Bibliothek die sämtliche Funktionalitäten anbietet.
- Ist auch Teil des SDK damit man Android Apps entwickeln kann.
- Diese Bibliothek ist extrem groß und eine offizielle Dokumentation ist im World Wide Web zu finden.

Android Ressourcen

- Android Ressourcen bestehen aus layout, strings, bilder usw..
- Ressourcen werden im code durch ID's geladen und können somit Objekten zugewiesen werden.
- Als primäre benötigt man ein Layout welches unter res/layout zu finden ist.
- Das einfachste Layout ist das lineare welches wir auch in den Workshops nutzen werden.

Android UI



Android UI

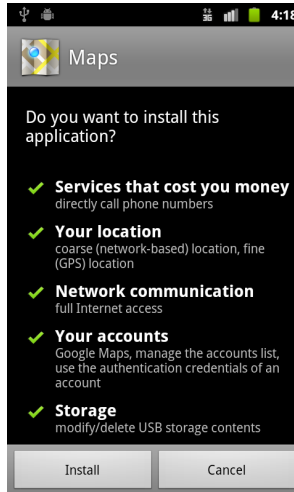
- Eine Layout besteht aus mehreren Views und auch View Gruppen.
- Dadurch ist es möglich eine generische Anordnung herzuleiten.
- Es gibt drei Typen von Layouts: Linear, Relative, List und Grid.

Zugriffsrechte

Android Zugriffsrechte

- Android verfügt im Application Layer über ein Capability based Access Control.
- Das heisst jede App die Zugriffsrechte z.B. auf das Netzwerk benötigt muss diese in der AndroidManifest.xml definieren.
- Dadurch wird der Nutzer bei der Installation einer Applikation nach den Rechten gefragt.
- Zugriffsrechte können auch selbst durch die App neu definiert werden die Interfaces beschränken.

Android Zugriffsrechte



Android Zugriffsrechte

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.DISABLE_KEYGUARD" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.READ_LOGS" />
<uses-permission android:name="android.permission.WRITE_CALL_LOG" />
```

IPC

Basis

- IPC basiert meist auf Binder IPC.
- Es existiert ein ServiceManager welche alle Android Services vorhält.
- Highlevel Netzwerkimplementierung kann auch für IPC benutzt werden.

Intents

- Die meist genutzte Art zwischen Applikationen zu kommunizieren.
- Kann nicht nur Informationen sondern auch Daten transportieren.
- Die Gegenstelle benötigt einen BroadcastReceiver und IntentFilter.

Services

- Services bieten ein Interface via Binder an (AIDL).
- Dadurch kann man Methoden in einer Serviceinstanz aufrufen.
- Services können sich selbst durch die Android Zugriffskontrolle schützen.

LocalSockets

- Highlevel Netzwerkimplementierung um Sockets nutzen zu können.
- Apps können via Client-Server Beziehung kommunizieren.

Native Unterstützung

C & C++ Bibliotheken

- Java kann via Java Native Interface C und C++ Bibliotheken laden.
- Diese müssen mit einer nativen Toolchain auch NDK genannt für die Plattform kompiliert werden.
- Dadurch wird es Funktionen aus Low level Bibliotheken zu nutzen.

Ende

Hardware Einrichten

Treiber

- Unter Windows oder OSX muss man evtl. noch die sogenannten ADB Treiber installieren.
- Diese sollte man auf der Gerätehersteller Webseite finden können.

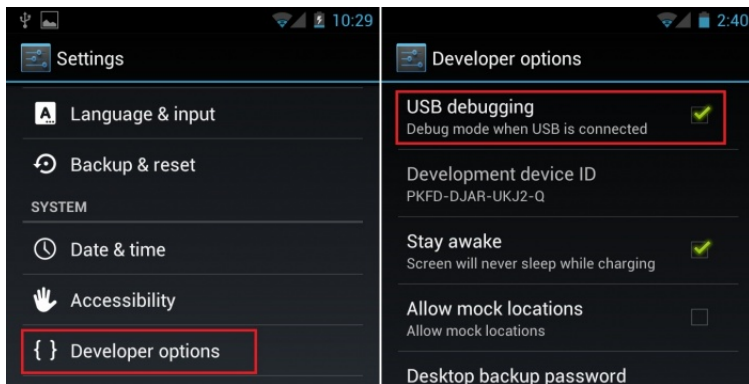
Debugging Modus

- Aktivieren des Debugging Modus ermöglicht es live und realistisch Apps debuggen.
- App kann auf dem Gerät direkt gestartet werden.
- Deutlich schneller als der Emulator und stellt eine optimale Testumgebung dar.
- Zugriff der Nutzung wird durch Bestätigung gewährt.

Debugging Modus



Debugging Modus



Workshop Part 1

Workshop Part 2

Workshop Part 3

Quellen

www.groovypost.com

www.companionlink.com

www.xda-developers.com

www.tutorialspoint.com

www.pcmag.com

developer.android.com