

O'Reillys Taschenbibliothek

2. Auflage  
Behandelt Android 4.3



# Android Programmierung

*kurz & gut*

O'REILLY®

*Jörg Staudemeyer*

Wenn man beginnt, sich mit der Anwendungsentwicklung unter Android zu beschäftigen, trifft man zunächst auf eine Vielzahl von Begriffen, die spezifisch für Android sind oder die im Android-Umfeld in einer bestimmten Weise verwendet werden, und die einem anfangs das Verständnis der Zusammenhänge erschweren. Um den Einstieg zu erleichtern, sind daher im Folgenden die wichtigsten dieser Begriffe aufgelistet und kurz erläutert.

## ACP

*Android Compatibility Package* – zum Android-SDK gehörende Erweiterungsbibliothek, mit der sich neuere Android-Funktionen auch auf Geräten mit einer älteren Android-Version nutzen lassen.

## Activity

Einer der vier Android-Komponententypen. Wird implementiert durch eine von `android.app.Activity` abgeleitete Klasse, die für die Darstellung einer Benutzeroberfläche und die Behandlung der darin auftretenden Ereignisse zuständig ist.

## ADB

*Android Debug Bridge* – vielseitiges Befehlszeilen-Werkzeug für die Kommunikation mit einem laufenden Emulator (→ AVD) oder einem angeschlossenen Android-Gerät.

## ADT

Android Development Tools – Plugin für die integrierte Entwicklungsumgebung → *Eclipse*

*se*, das zur Entwicklung von Android-Anwendungen benötigt wird.

## AIDL

*Android Interface Definition Language* – Android-spezifische Schnittstellen-Definitionssprache. Wird benötigt, wenn Daten zwischen nebenläufigen Prozessen ausgetauscht werden sollen. In anderen Fällen ist es einfacher, mit gebundenen Services zu arbeiten.

## APK

*Android Package* – im JAR-Format gepacktes Dateiarchiv, das eine komplette, installierbare Android-Anwendung enthält. Wird im Android-Buildprozess vom `apkbuilder`-Tool erzeugt.

## App-Widget

Kleine Anwendungsoberfläche (→ *View*), die innerhalb einer anderen Anwendung dargestellt werden kann. Wird typischerweise für den → *Home-Screen* verwendet.

## AVD

*Android Virtual Device* – Emulator zum Testen von Android-Anwendungen anstelle eines physischen Geräts.

## Beam

*Android Beam* ist eine neue Technologie für den drahtlosen Datenaustausch über kurze Strecken (→ *NFC*).

## Broadcast-Receiver

Einer der vier Android-Komponententypen. Wird implementiert durch eine von `android.content.BroadcastReceiver` abgeleitete

Klasse, die dafür zuständig ist, auf bestimmte Ereignisse zu reagieren, die über *Intents* systemweit verbreitet werden.

### **Content-Provider**

Einer der vier Android-Komponententypen. Wird implementiert durch eine von `android.content.ContentProvider` abgeleitete Klasse, die für dafür zuständig ist, anderen Komponenten Informationen zur Verfügung zu stellen.

### **Daydream**

Interaktiver Bildschirmschoner, der in eine Anwendung integriert werden kann.

### **DDMS**

*Dalvik Debug Monitor Server* – Bestandteil des Android-Systems, das Debug-, Monitoring- und Simulationsdienste zur Verfügung stellt, die für Zwecke der Anwendungsentwicklung genutzt werden können.

### **DEX**

*Dalvik Executable* – Format der von der Dalvik Virtual Machine auszuführenden Java-Maschinencode, wird aus Standard-Java-Klassendateien (*.class*-Dateien) generiert. Die Dateien haben die Endung *.dex*.

### **Dialog**

Fenster in der Anwendungsoberfläche, das den Programmablauf unterbricht, um eine Meldung darzustellen oder eine Benutzereingabe abzufragen.

### **DVM**

*Dalvik Virtual Machine* – virtuelle Maschine für die Ausführung des speziellen, auf die besonderen Anforderungen mobiler Geräte zugeschnittenen → DEX-Bytecodes.

### **Eclipse**

In einem Open-Source-Projekt entwickelte umfangreiche integrierte Entwicklungsumgebung (IDE, *Integrated Development Environment*), die mit Hilfe des Plugins → ADT auch zur Entwicklung von Android-Anwendungen verwendet werden kann.

### **Fragment**

Teil der innerhalb einer → *Activity* implementierten Benutzeroberfläche einer Anwendung.

### **Google Play Services**

Über Google Play herunterladbare App, die anderen Apps zusätzliche Funktionen zur Verfügung stellt, z.B. für den Zugriff auf Google Services.

### **IME**

*Input Method* – Eingabemethode, Komponente für die Texteingabe (z.B. Hardware-Tastatur oder virtuelle Tastatur auf dem Bildschirm).

### **IMF**

*Input Method Framework* – API, die es ermöglicht, die Eingabemethode (→ *IME*) von der Anwendung zu trennen.

### **Instrumentation**

Hilfsmittel für Testklassen, das es erlaubt, in der zu testenden Anwendung bestimmte Ereignisse abzufangen und den jeweiligen Zustand der Anwendung abzufragen.

### **Intent**

Instanz der Klasse `android.content.Intent`, die dazu dient, anwendungsübergreifend Funktionen aufzurufen und Informationen zu übergeben.

### **Komponente**

Java-Klasse, die als Grundbaustein in jeder Android-Anwendung enthalten ist. Es gibt vier Arten von Komponenten: *Activity*, *Service*, *Content-Provider* und *Broadcast-Receiver*.

### **Layout**

In einer XML-Datei beschriebener Aufbau einer Anwendungsoberfläche.

### **LBS**

*Location Based Service* – Anwendung, die auf der aktuellen geographischen Position des Benutzers beruht.

### **Lint**

Werkzeug zur statischen Code-Analyse. Es hilft, anhand von Quellprogrammen Programmierfehler und Optimierungsmöglichkeiten im Programm zu finden.

### **Loader**

In Android 3.0 (API Level 11) eingeführter Mechanismus zum asynchronen Laden von Daten über das Netzwerk in eine *Activity* oder ein *Fragment*.

## LogCat

Werkzeug, mit dem sich vom Entwicklungssrechner aus Protokollausgaben im Android-System sammeln und auswerten lassen.

## Manifest

„Ladungsverzeichnis“ – XML-Datei im Wurzelverzeichnis der Android-Anwendung, in der alle Komponenten sowie sonstige, für die Installation und den Betrieb wichtige Eigenschaften der Anwendung deklariert sind.

## MTP

*Media Transfer Protocol* – Standard für den Austausch von Mediendaten. Wird von Android für die Kommunikation mit Kameras verwendet.

## NDEF

*NFC Data Exchange Format* – Format für den Austausch von Informationen über → *NFC*.

## NDK

*Native Development Kit* – Ergänzungspaket zum → *SDK*, mit dessen Hilfe Android-Anwendungen erstellt werden können, die in C oder C++ programmierten Maschinencode enthalten.

## NFC

*Near Field Communication* (Nahbereichskommunikation) – Technologie für den berührungslosen Datenaustausch auf kurze Distanz, die beispielsweise für Zugangskontrollen, Zahlungen, Tickets usw. eingesetzt werden kann.

## OpenGL

*Open Graphics Library* – Standardbibliothek für hochperformante zwei- und dreidimensionale Grafikausgaben.

## Parcel

„Päckchen“ – Instanz der Klasse `android.os.Parcel`, die für den Austausch serialisierter Objekte zwischen Prozessen im Android-System verwendet wird.

## ProGuard

Android-Werkzeug, mit dem man Programmcode komprimieren, optimieren und ein Reverse-Engineering erschweren kann.

## PTP

*Picture Transfer Protocol* – Standard für den Austausch von Bilddaten. Wird von Android

für die Kommunikation mit Kameras verwendet.

## QEMU

*Quick Emulator* – Open-Source-Projekt zur Entwicklung eines Prozessor-Emulators; dient als Basis für das → *AVD*.

## RenderScript

Android-API für die Ausführung von hochperformanten Berechnungen, die in C programmiert sind.

## SDK

Software Development Kit – komplette Entwicklungsumgebung für die Erstellung von Android-Anwendungen auf Java-Basis. Enthält alle benötigten Hilfsmittel als Befehlszeilenwerkzeuge; für die Verwendung mit → *Eclipse* werden zusätzlich die → *ADT* benötigt.

## Service

Einer der vier Android-Komponententypen. Von `android.app.Service` abgeleitete Klasse, die über keine eigene Benutzeroberfläche verfügt und entweder asynchron im Hintergrund läuft (ungebundener Service) oder anderen Komponenten synchrone Funktionen zur Verfügung stellt (gebundener Service).

## SQLite

Plattformunabhängige C-Bibliothek für eingebettete SQL-Datenbanken, die durch eine Java-API gekapselt, Android-Anwendungen zur Verfügung gestellt wird.

## Studio

Auf IntelliJ IDEA basierende, in Kooperation mit der Firma JetBrains entstandene graphische Entwicklungsumgebung für Android.

## Style

In einer XML-Datei definierte Sammlung von Stil-Eigenschaften für die einheitliche Formatierung und Darstellung von → *Views* und Fenstern.

## Theme

Entspricht einem → *Style*, wird aber auf eine gesamte → *Activity* oder eine gesamte Anwendung angewendet.

## Toast

In einem Popup-Fenster dargestellte Meldung, die keinen Fokus erhält und nach einer

festgelegten Zeit von selbst wieder verschwindet.

#### **View**

„Ansicht“ – Bezeichnung für ein Element einer Android-Anwendungsoberfläche. Alle Views einer Anwendungsoberfläche sind – ausgehend von einem Wurzel-View – baumartig ineinander verschachtelt.

#### **Wallpaper**

„Tapete“ – in der Regel animierte Hintergrundgrafik für eine Anwendungsoberfläche.

#### **Zygote**

Interner Prozess im Android-System, der mittels *fork* sehr schnell vorinitialisierte Prozesse für neu gestartete Anwendungen erzeugen kann.

#### **9-Patch-Drawable**

Aus neun Feldern bestehende skalierbare Grafik, bei der die vier Eckfelder in ihrer Größe unverändert bleiben.