



# Android Entwicklung

Google Technology Group

München, 23. Juli 2009

Markus Junginger

greenrobot

# Über mich

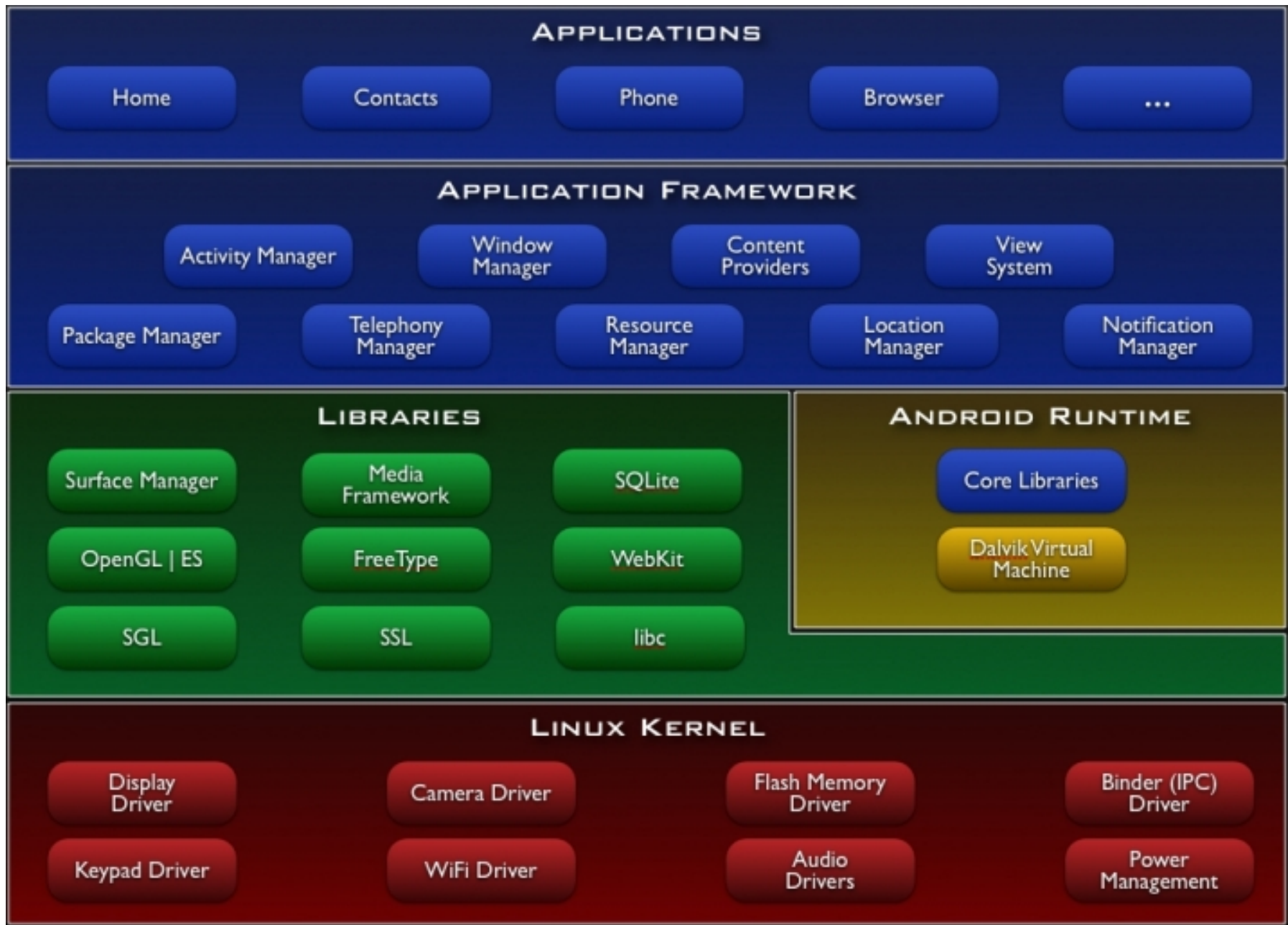
- Java Guy (Java 1.0 – 6, JEE, Web, UI, ...)
- Windows Mobile (C++ & .NET), JME
- Erste Android App Ende 2007
- greenrobot: Android Startup (+iPhone)
- Derzeit Android Projekt bei dt. Telco
- Android Entwicklung, Beratung, Schulung
- markus@greenrobot.de

# Outline

- Einführung
- Activities & Intents
- Hello World
- User Interface (UI)
- Typische Tasks (Entwicklung)
- Erfahrungswerte

# Warum ist Android interessant?

- Was ist Android? Open Source Betriebssystem (Linux) für Mobile
- 20 Android Geräte bis Ende 2009
- 3stellige Wachstumsrate (Studie: 900%)
- (Einzig?) Ernsthafte iPhone Konkurrenz
- Gute Adoption durch Industrie (Preis, ...)
- Konkurrenz → Massenmarkt?



# Vier Entwicklungsarten

- Android SDK: Entwicklung mit Java
- Webseiten (Chrome/WebKit)
- Cross Platform Apps
- Native Entwicklung
  - NDK in Verbindung mit Android SDK
  - Eigene Android Firmware

# Vier Entwicklungsarten

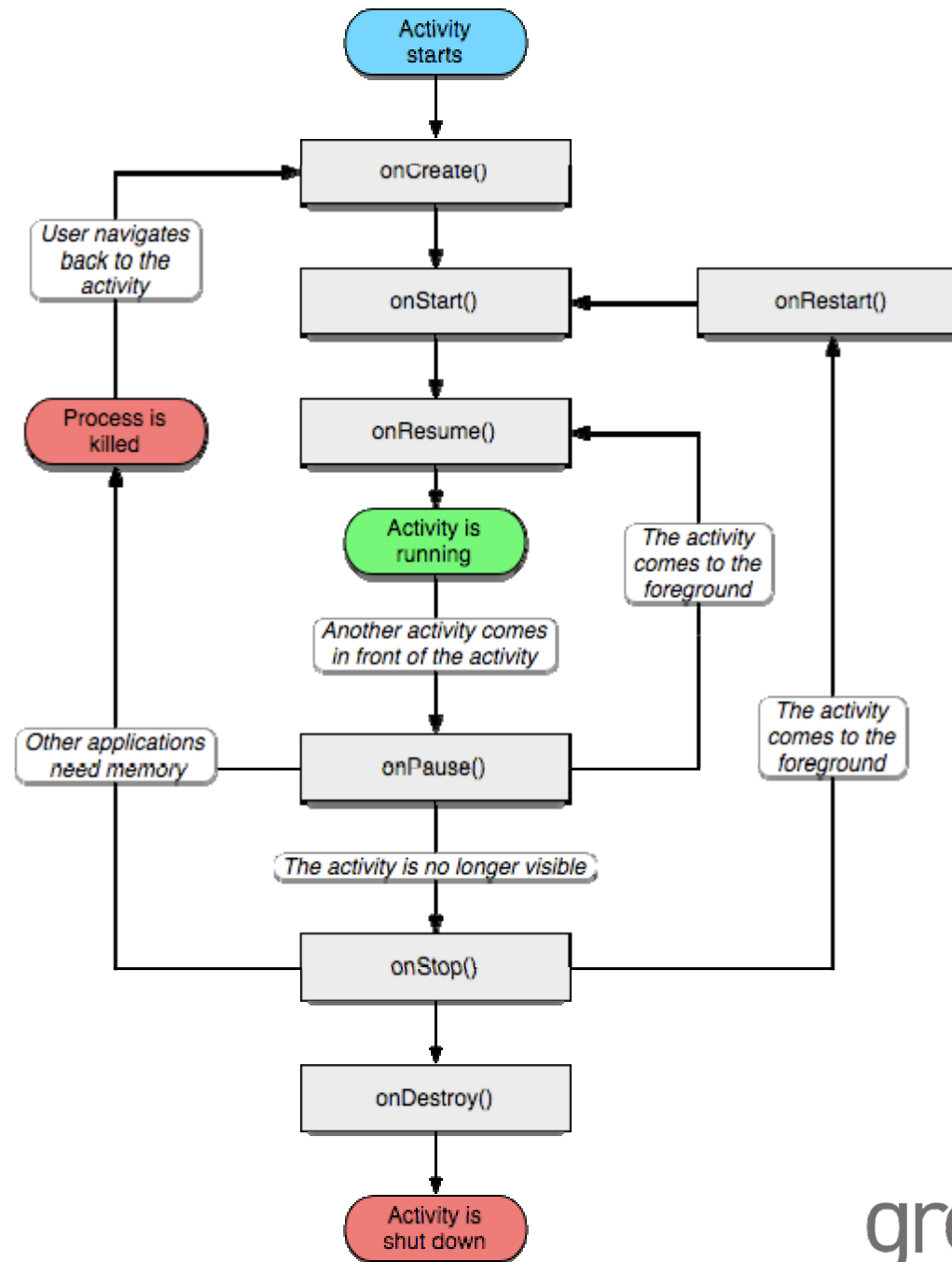
- **Android SDK: Entwicklung mit Java**
- Webseiten (Chrome/WebKit)
- Cross Platform Apps
- Native Entwicklung
  - NDK in Verbindung mit Android SDK
  - Eigene Android Firmware

# Entwicklung mit Android SDK

- Java 5 (Scala, JRuby)
- Java 5 SE APIs teilweise vorhanden (io, nio, lang, util, math, etc.)
- Proprietäre Android APIs (android.\*)
- Java Bytecode wird in DEX umgewandelt
- Dalvik Virtual Machine
- IDE: Eclipse (oder von Hand)
- Device Emulator

# Activities

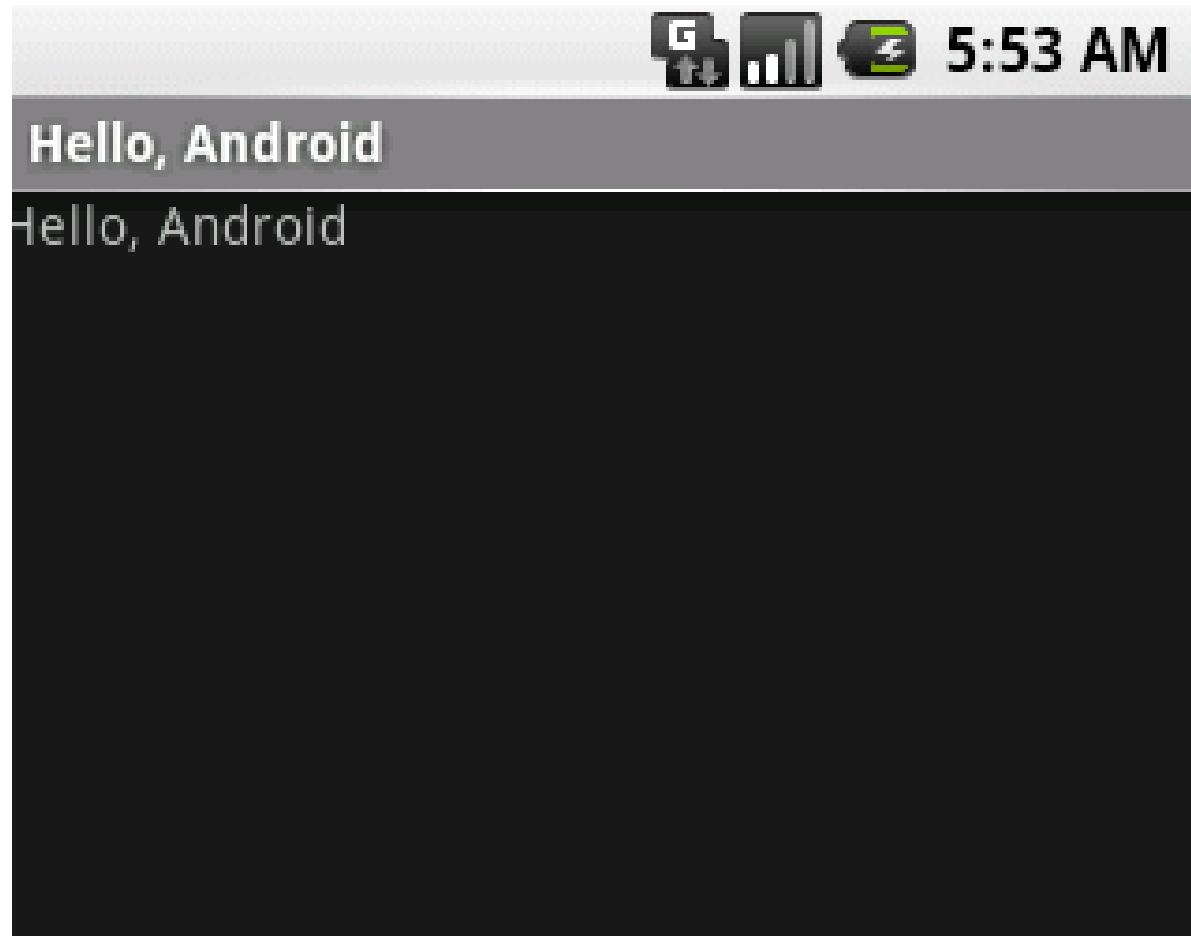
- Einstiegspunkt in Android Programm
- Aktiver Programmteil
- Eine Activity setzt (mind.) ein View (GUI)
- Activities unterliegen Life Cycle
- onResume und onPause (Persistenz!)
- Android OS kann Activities killen
- Alternative: Background Services



# Intents

- Eine Absicht etwas zu tun ohne zu definieren wie es getan werden soll
- Vernetzung von Apps über lose Kopplung
- Angelehnt an URLs (Beispiel: „tel:123“)
- Auch für das Aktivieren eines neuen Zustand einer App: Starten einer Activity

# Hello World

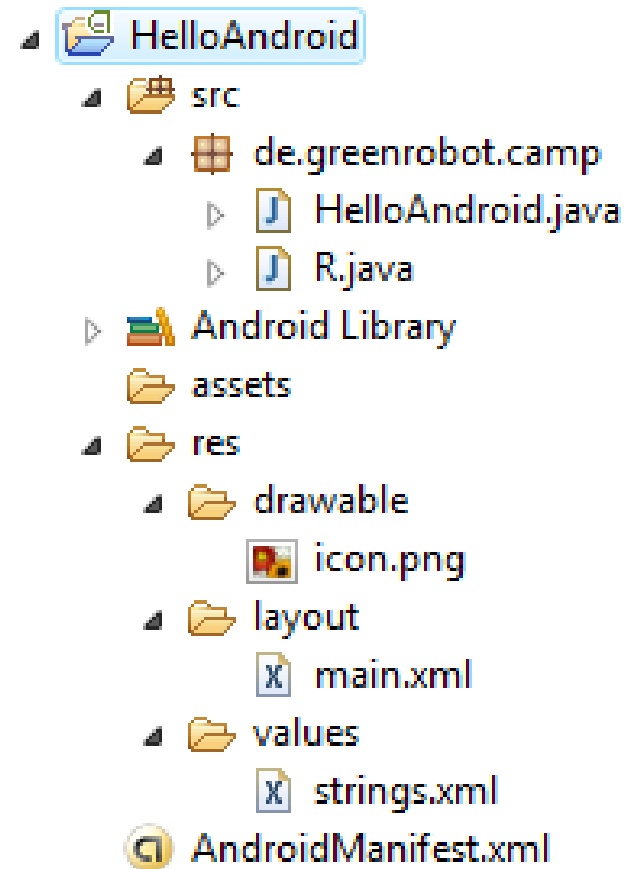


# Hello World

```
public class HelloAndroid extends
    Activity {
    @Override
    public void onCreate(Bundle state) {
        super.onCreate(state);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```

# Android Projekt anlegen

- IDE erstellt Gerüst
- R.java ist generiert
- res Verzeichnis
- AndroidManifest.xml



# UI Prinzipien: Klassisch...

- Als Ressourcen oder programmatisch
- UI Komponenten: Basisklasse View
- Layouts (Linear, Grid, Table, Relative, ...)
- UI Thread: Änderungen an der GUI
- Events (OnClick, ...)
- Visueller GUI Editor in Eclipse
- Low-Level Canvas API

The screenshot shows the Android Studio IDE with the following components:

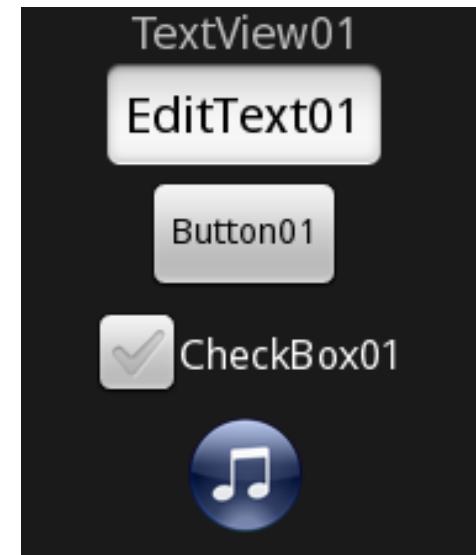
- Top Bar:** Tabs for 'HelloAndroid Manifest', 'main.xml', and 'strings.xml'. A configuration bar with fields for MCC, MNC, Lang, Region, Orient, Density, Touch, Keybrd, Input, and Nav. A 'Create...' button is also present.
- Main Editor:** A preview window showing 'Hello World!' on a black background.
- Outline Pane:** A tree view showing a 'LinearLayout' containing a 'TextView'.
- Properties Pane:** A table listing properties for the selected 'TextView'.

Property	Value
Shadow color	
Shadow dx	
Shadow dy	
Shadow radius	
Single line	
Sound effects enabled	
Tag	
Text	@string/hello
Text appearance	
Text color	
Text color highlight	
Text color hint	
Text color link	
Text scale X	
Text size	
Text style	
Typeface	
Visibility	
Width	

greenrobot

# Typische Views

- TextView
- EditText
- Button
- Checkbox
- ImageView
- WebView (Achtung: JavaScript aktivieren)
- MapView (benötigt Registrierung/Key)



# Beispiel ListView

- Adapter (MVC) hält die Daten der Liste vor
- ListActivity vereinfacht Handhabung
  - setListAdapter z.B. mit ArrayAdapter aufrufen
- Eigene Adapter von BaseAdapter ableiten
  - Object **getItem**(int position)
  - View **getView**(int position, View convertView, ViewGroup parent)

# Asynchrone Verarbeitung & UI

- UI Thread darf niemals blockiert werden
- Nach 5s Android System Dialog
- Netzwerkzugriffe, etc. in eigenem Thread
- ProgressDialog, Progress Indicator
- UI Update aber wieder im UI Thread
  - Verschiedene Möglichkeiten:  
Activity.runOnUiThread, View.post, Handler  
Android 1.5: AsyncTask

# Client-/Server Kommunikation

- Apache HttpClient 4
- Kein SOAP out of the box → kSOAP
- REST mit XML
  - XML: SAX, DOM, Pull aber kein XPath
- REST mit JSON
  - JSON integriert
  - Einfach zu parsen

# Persistenz

- Files für XML, Objekte, ...
- Preferences mit Key/Value Paaren  
→ Vorteil: einfache UI Erstellung
- SQLite
- Über Applikations-/Prozessgrenzen:  
Content Provider

# Performance

- Hohe Performance für normale Apps
- Optimierungen nötig für:
  - Background Services (be nice)
  - Spiele, Multimedia (flüssiger Ablauf)
- GC legt VM für 100-200ms lahm
- Regel #1: GC vermeiden. Keine neuen Objekte in zentralen Schleifen anlegen
- Versteckte Anlage von Objekten

# Android Market

- Google
- Alternative App Stores existieren
- Registrierung notwendig, einmalig \$25
- Signieren mit Zertifikat (self-signed)
- Kein Review Prozess, sofort im Market
- Bezahlung über Google Checkout
- 70% der Einnahmen an Entwickler

# Pluspunkte

- Schneller Einstieg für Java Entwickler
- Sicherheit (Permissions, Prozesse, User)
- All-in: Eclipse-basierte IDE
- Loose Coupling zwischen Apps (Intents)
- Einfache Integration WebKit, Maps
- UI Customization mit Themes
- GPS, Kamera, Beschleunigungssensor

# Minuspunkte

- Nach schnellem Einstieg komplexe Details
- Dalvik VM erinnert an Java 1.0/1.1 VM
- Homescreen Widgets sehr eingeschränkt
- Fehlende Event Unterstützung UI Editor
- Ressourcen (R) vs. mehrere Projekte
- Unit Testing ist kompliziert
- Android Market Bugs

# Weitere Ressourcen

- [Android.com](https://android.com)
- SDK: Dev Guide mit guten How-tos, Referenz und API Demos mit Source
- [Android Developer Blog](#)
- [android-hilfe.de](https://android-hilfe.de) Forum
- [XING Gruppe](#)

# Mitmachen!

- Android User Group München
  - [sites.google.com/site/androidmunich/](http://sites.google.com/site/androidmunich/)
  - [groups.google.de/group/android-muenchen](http://groups.google.de/group/android-muenchen)
- Droidcon: 3. & 4. November in Berlin
  - Android Barcamp & Konferenz
  - Call for Paper bis 1.9.
  - [droidcon.com](http://droidcon.com)

# Vielen Dank! Q&A

[groups.google.de/group/  
android-muenchen](https://groups.google.de/group/android-muenchen)

Markus Junginger

[markus@greenrobot.de](mailto:markus@greenrobot.de)

<http://greenrobot.de>

<http://jars.de>

Twitter: [greenrobot\\_de](https://twitter.com/greenrobot_de)



greenrobot