# 

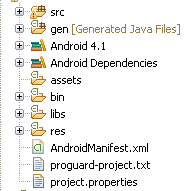
# C:\Users\Anatol\Dropbox\Moodle\Android\Bilder\google_android_logo.jpgTutorial 05: Trennung von Code und Ressourcen

**Wie sind Ressoucen und Java-Code miteinander verbunden?**

**Kennzeichen von Android-Apps:   
Durchgängige Trennung von CODE und RESSOURCEN**

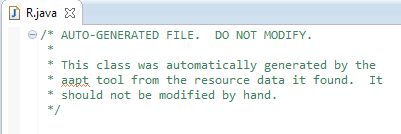
* **Code:** Steuert meine App
* **Ressourcen:**  Beschreiben das Layout, verwendete Texte und genutzte Grafiken, etc.

**Im Android-Projekt durch verschiedene Ordner erreicht:**

* **src**-Ordner/Knoten: Java-Code:
* **res**-Ordner/Knoten: Ressourcen

**Zweiteilung/Arbeitsteilung in Teams möglich:**

* **Fachlogik** / Ablauf des Programms: Java-Entwickler
* **Layout** / Design: Designer (XML-Kenntnisse)

**Wer sorgt für die Zusammenarbeit?**

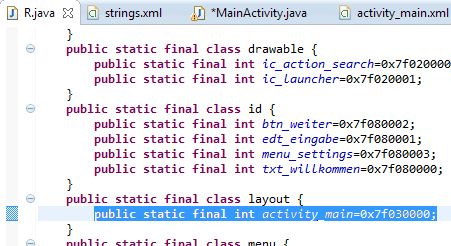
* Die Klasse **R.java** im „gen“-Ordner!
* Vom Android Asset Packaging Tool (aapt) stets **automatisch generiert**
* **Struktur:**
  + eine finale Klasse „R“
  + weitere, innere Klassen mit statischen (Klassen-) *Eigenschaften* vom Typ *int*
  + Namen der inneren Klassen:   
    **Verzeichnisnamen** im **res**-Ordner (layout…)  
    **Werte der Dateien** im **values**-Ordner (string, color)  
    Also: strings.xml beinhaltet <string>-Werte…
  + alle Eigenschaften haben einen *eindeutigen Zahlenwert*, zur eindeutigen Identifikation.

# C:\Users\Anatol\Dropbox\Moodle\Android\Bilder\R.java_02.jpgAblauf:

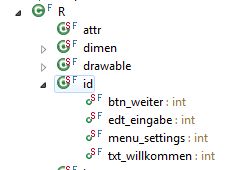
1. In der Start-Activity wird mit dem Befehl

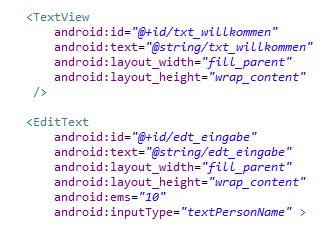
**setContentView(R.layout.activity\_main);**

eine Layout Ressource über die Klasse R.java geladen

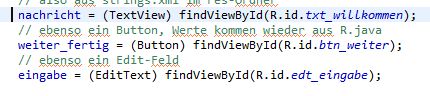
1. In der Klasse **R.java** wird die passende Activity gefunden und der dazu passende id-wert genommen,
2. damit die Layout-Ressource **activity\_main** im Ordner layout aufgerufen und ausgeführt wird: So wird also das passende Layout der Start-Activity geladen.

# Die Klasse id der R.java

Beinhaltet ids von diversen Komponenten, z.B. Buttons, Edit-Feldern, etc…

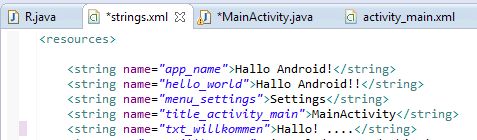
 **Wie wird die id erzeugt?**

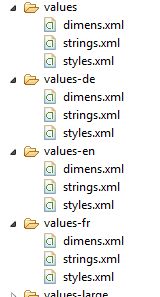
Mit **@+id** wird in einer XML-Datei die Ressource zur Klasse R.java als id aufgenommen: hier ein *TextView* und ein *EditText*

In der passenden Activity kann über den Befehl **findViewById(R.id.txt\_willkommen)** nun mit der Ressource (Ein Textfeld) gearbeitet werden; man erhält also die View auf diese Ressource/id.

# C:\Users\Anatol\Dropbox\Moodle\Android\Bilder\R.java_05.jpgWie können Ressourcen auf andere Ressourcen verweisen?

Wir schauen uns noch eimal die XML-Layout Datei an:

Der Text des TextView steht hier nicht, sondern kommt aus einer weiteren Ressource (strings.xml). mit **@string/txt\_willkommen** greifen wir also auf den Wert des XML-Knotens *string* mit dem Namen *txt\_willkommen* zu.

**Warum so kompliziert?**

Dadurch lassen sich ganz einfach **mehrsprachige Apps** generieren: man legt einfach neue Ordner mit der Bezeichnung **values-en**, **values-fr**, **values-de** mit passenden strings.xml Dateien an und sorgt so dafür, dass ein Android-Gerät mit englischer Sprache auf die passende Datei zurückgreift und so eine englische Beschreibung/Texte erhält.