

Einführung in die Entwicklungsumgebung Android Studio

Zuletzt bearbeitet von Alexander Bazo

U01 | Grundlagen und Einführung in die Entwicklungsumgebung Android Studio



Einführung in die Anwendungsentwicklung mit Android

U01 | Grundlagen



Voraussetzungen

Für die komfortable Entwicklung von Android-Applikationen werden eine Reihe von Tools und Komponenten benötigt. Zusammen erlauben diese Werkzeuge das Programmieren, Kompilieren, Testen und Installieren von Android-Apps. Ihre erste Aufgabe in diesem Kurs ist es, Ihren privaten Rechner für die Android-Entwicklung einzurichten. Für die Entwicklung einer Android-App benötigen Sie einen Editor der es Ihnen erlaubt JAVA-Code und XML-Dateien zu schreiben. Zusätzlich brauchen Sie Zugriff auf die Klassenbibliotheken und Support-Tools des Android-Systems. Zum Testen Ihrer Anwendungen benötigen Sie ein Android-kompatibles Gerät oder einen sogenannten Emulator. Dieser stellt auf Ihrem Rechner ein virtuelles Android-System bereit. Google bietet eine Reihe von Tools an, die Ihnen die Einrichtung und das Arbeiten erleichtert: Diese offiziellen Tools werden auch wir nutzen. Die offizielle IDE von Google zur Entwicklung von Android-Apps ist AndroidStudio. In der Vergangenheit war das ADT-Plugin für Eclipse (wird aber nicht mehr weiterentwickelt) Standard.

Android Software Development Kit (SDK)

Die Android Developer Tools sind eine Reihe von Tools und können direkt von AndroidStudio aus verwendet werden. Sie erlauben den Zugriff auf die wesentlichen Bestandteile des SDKs. Damit können Sie Benutzerschnittstellen für ihre Applikation erstellen, Debug-Ausgaben anzeigen und Applikationen auf Emulatoren und Android-Geräten installieren und ausführen. Die einzelnen Tools können auch außerhalb von AndroidStudio genutzt werden. Der genaue Pfad auf ihrem Computer hängt von ihrem Betriebssystem und der Art der SDK-Installation ab. Die Tools finden sie in AndroidStudio unter dem Menüpunkt *Tools* und dann *Android* bzw. über die entsprechenden Symbole in der Leiste unter den Menüs.



Abbildung 1: Toolbar des Android Studios mit VDM und SDK Manager

XML-Editor

Einige der zentralen Elemente eines Android-Projektes (Ressourcen, Layouts, Manifest) werden in Form von XML-Dokumenten angelegt. Für Layout-Dateien stellt Ihnen AndroidStudio einen zusätzlichen Editor zur Verfügung: Sie können diese Dateien entweder direkt durch Modifikation des XMLs verändern oder einen graphischen Editor nutzen. Zwischen diesen beiden Ansichten können Sie am unteren Rand des Editor-Fensters wechseln.

API-Versionen

Es existieren verschiedene offizielle Versionen des Android-Systems. Da nicht alle Endgeräte automatisch die aktuellste Version unterstützen, ist es mitunter notwendig eine ältere Variante für die Entwicklung zu nutzen. In der Regel sind die Versionen untereinander rückwärtskompatibel. Über den Android SDK Manager können Sie die verschiedenen Plattformen (Versionen) herunterladen und damit in ihrer Entwicklungsumgebung verfügbar machen. Die Plattformen enthalten die jeweiligen Versionen der Android-Systembibliotheken sowie weitere versionsspezifische Tools. Wenn Sie ein neues Android-Projekt anlegen, können Sie wählen welche der verfügbaren Applikationen Sie nutzen bzw. unterstützen wollen.

Emulatoren

Das Android-SDK verfügt über einen Emulator. Dieses Tool erlaubt Ihnen, auf ihrem Rechner ein Android-Gerät zu virtualisieren. Ein Emulator ist in diesem Fall ein Programm, das auf ihrem Rechner ausgeführt wird, aber in sich ein anderes Betriebssystem und eine andere Hardware virtualisiert (das Android-SDK nennt diese Emulatoren Virtual Devices). Mit dem Android Virtual Device Manager können Sie verschiedene dieser Emulatoren anlegen und dadurch unterschiedliche Geräte (Smartphone/Tablet) mit unterschiedlicher Displaygröße und Android-Version virtualisieren. Wenn Sie ein solches Gerät angelegt haben, können Sie ihre Anwendungen über AndroidStudio auf diesem Emulator installieren und ausführen. Das Virtual Device stellt ihre App dann ähnlich wie auf einem tatsächlichen Android-Gerät dar. Sie können die Anwendung mit der Maus steuern. Ein einfacher Klick simuliert dabei einen Kontakt auf dem Touch-Screen. Das Ziehen des Fingers (Drag) kann über die gedrückte Taste und eine Mausbewegung simuliert werden.

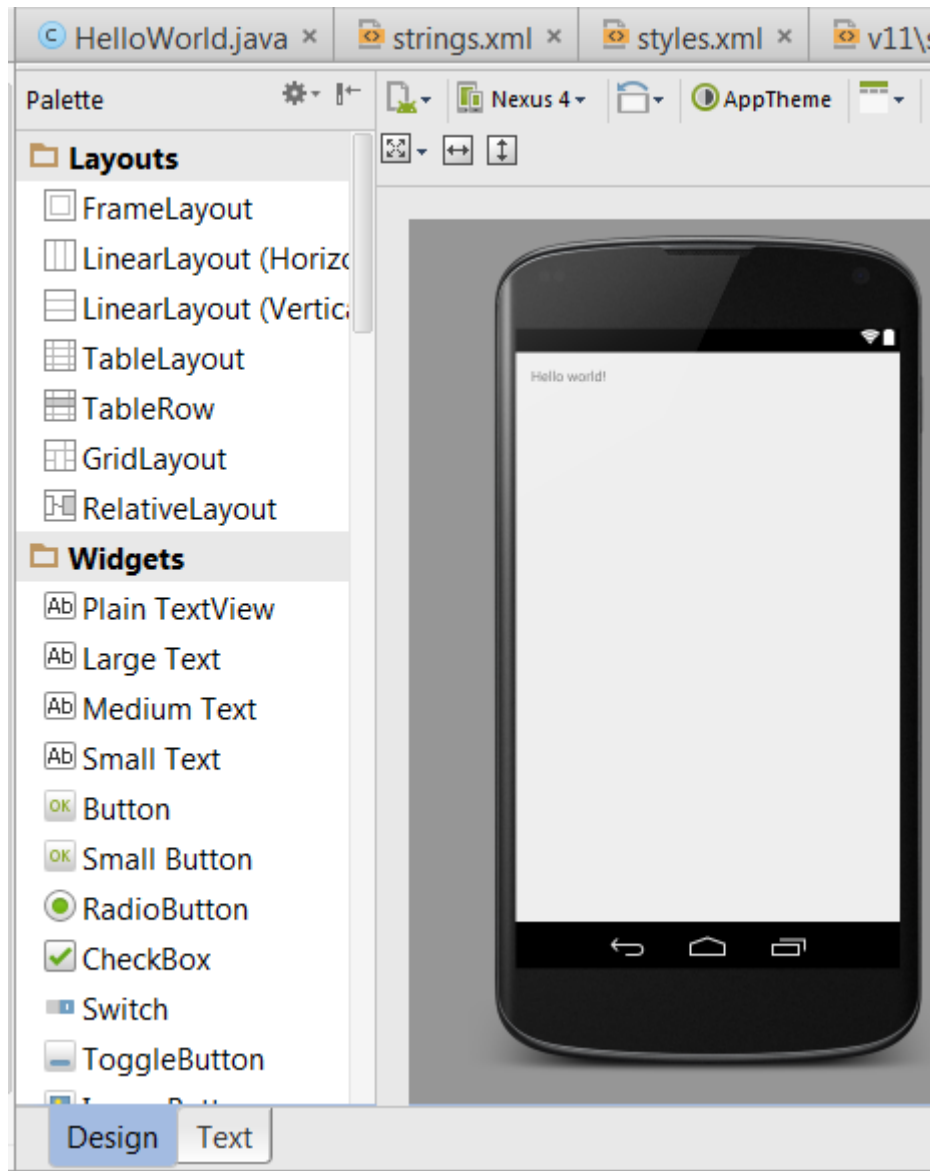


Abbildung 2: Graphischer Layout-Editor des Android Studios

Hinweise: Der Emulator benötigt eine sehr lange Zeit zum Starten (je nach PC 3 bis 15 Minuten) und läuft generell deutlich langsamer als ein reales Gerät. Auch hat der Emulator nicht alle Funktionen eines realen Gerätes, z.B. ist Bluetooth nicht nutzbar.

Android auf dem eigenen Rechner

Eine Installationsanleitung für AndroidStudio und ein ergänzendes Video dazu befindet sich bei der zweiten Vorlesungseinheit.

Die erste App: Ein neues Projekt anlegen

Klicken sie auf dem Willkommen-Screen von AndroidStudio auf *Start a new AndroidStudio project* bzw. auf in der Menüleiste auf *File* und dann *New Project...*, falls derzeit gerade ein Projekt geöffnet ist. Im folgenden Schritt werden grundlegende Daten für die App abgefragt:

- **Application Name:** Dieser Name erscheint auf dem Gerät des Benutzers im Launcher und sollte ihre App beschreiben.
- **Company Domain:** Die Domain der Entwicklerfirma in umgekehrter Reihenfolge. Aus `google.com` wird etwa `com.google`
- **Package Name:** Dieser Bezeichner soll Ihre App eindeutig identifizieren und ist von zentraler Bedeutung für ihre App selbst, aber auch für das mögliche Zusammenspiel mit anderen Applikationen. Er wird standardmäßig automatisch aus dem Namen der Anwendung und der Firmendomain erstellt, aber per Klick auf *edit* ist eine nachträgliche Veränderung möglich. Der *Package Name* wird in Form einer vereinfachten URI angegeben (Beispielsweise: `de.ur.mi.android.helloandroid`). Verbreitet ist hier das Schema der *Reversed URL*, also der umgedrehten Webadresse mit einer abschließenden Ergänzung für die jeweilige Applikation.

Im nächsten Schritt erfolgt nun die Frage nach den Android APIs, die ihre App unterstützen soll. Für diesen Kurs ist nur die **Phone and Tablet**-Klasse relevant. Wählen sie in der Liste *API 16* aus. Belassen sie die Wahl für die erste *Activity* der App im folgenden Schritt bei der Vorauswahl. Auch im nächsten Schritt können sie die Vorgaben beibehalten. Nach einem Klick auf *Finish* und einer Wartezeit ist das Projekt mitsamt einer App namens **app** angelegt.

Mit *File* und dann *New Module...* können dem Projekt weitere Apps hinzugefügt werden.

In der Dateiliste auf der linken Seite finden sie unter **app, java, <paketname>, MainActivity.java** ihre erste *Activity*. Eine *Activity* bündelt die Logik einer bestimmten, sichtbaren Komponente ihrer App. Das Aussehen dieser Komponente wird über eine Layout-Datei definiert. Diese Layouts finden sich als XML-Datei unter **res/layout** und werden innerhalb des *Activity*-Codes referenziert.

Ausführen des angelegten Projekts

Sie können die erstellte Anwendung direkt ausführen in dem sie Ihr Projekt über den grünen *Run*-Button in der Menüleiste (etwa in der Mitte) starten. Die Anwendung wird dann entweder auf einem angeschlossenen Android-Gerät oder einem *Virtual Device* ausgeführt. Sollten weder ein Gerät angeschossen noch ein Emulator verfügbar sein wird *Android Studio* sie auffordern, ein neues *Virtual Device* anzulegen.

Ihre App wird nun auf dem Emulator oder dem angeschlossenen Gerät installiert und ausgeführt. Wenn Sie den Emulator nutzen, kann dessen Boot-Vorgang einige Minuten in Anspruch nehmen. Sie sollten anschließend ihre App und dort den Text *Hello world!* sehen.

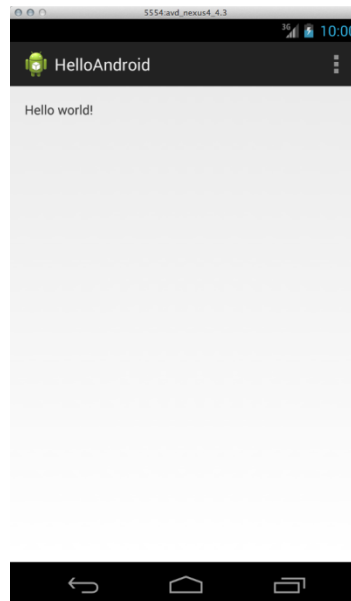


Abbildung 3: Screenshot der ersten App

Erste Modifikationen

Schauen Sie sich nun die Datei `activity_main.xml` an. Hier finden Sie die Definition für das Textfeld, das den *Hello world!*-Text beinhaltet. Der Text selber wird über das XML-Attribut `text` gesetzt. In diesem Beispiel wird der anzuzeigende Text nicht direkt angegeben sondern durch einen Verweis dargestellt. Ressourcen, auch Text, können in gesonderten Dateien zentral gespeichert werden und dann von verschiedenen Stellen referenziert werden. Der hier verwendete Text- Baustein `hello_world` wird in der Datei `res/values/strings.xml` definiert. Öffnen Sie diese Datei, versuchen Sie den definierten Text zu ändern und führen Sie ihre Applikation erneut aus.

Inhalt von `activity_main.xml`

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world"/>

</RelativeLayout>
```