

Übung 2 - Native App-Entwicklung: Android

Nilan Marktanner, nilan.marktanner@gmail.com

25. Januar 2017

In diesem Übungsblatt werden wir die grundlegenden Konzepte von Android entdecken. Sie lernen die Struktur und wichtige Dateien eines Android Projekts kennen und untersuchen das Verhalten von Activities.

Dieses Übungsblatt wird während des ersten Kurstermins bearbeitet.

1 Hello Android World

1.1 Vorbereitungen

- Stellen Sie sicher, dass Sie **Übung 1** vollständig bearbeitet haben. Lassen Sie sich bei Problemen von Kommilitonen oder dem Kursleiter helfen.
- Arbeiten Sie mit dem vorbereiteten *Hello Android World* Projekt im `exercices2`-Ordner. Öffnen Sie dazu den Ordner `HelloAndroidWorld` als Projekt in Android Studio.
- Untersuchen Sie die Projektstruktur.
- Führen Sie die App im Emulator oder auf dem Smartphone aus. Welche Elemente sind in der App erkennbar? Welcher Text wird angezeigt?

1.2 Variablen von der App Logik trennen

- Untersuchen Sie die Datei `MainActivity.java`. Wie kommt der angezeigte Text zustande?
- Klicken Sie auf `default_user_name` während Sie die Strg-Taste gedrückt halten oder öffnen Sie direkt die Datei `strings.xml` im Ordner `res/values/`.
- Passen Sie den String `default_user_name` nach Belieben an.
- Führen Sie die App erneut aus.

1.3 Das Android Manifest

- Untersuchen Sie die Datei *AndroidManifest.xml*.

1.4 Layout

- Untersuchen Sie die Datei *MainActivity.java*. Wie wird der Inhalt der `MainActivity` gesetzt?
- Klicken Sie auf `activity_main` während Sie die Strg-Taste gedrückt halten oder öffnen Sie direkt die Datei `activity_main.xml` im Ordner `res/layout/`. Aus welchen Elementen besteht diese Layoutdatei?
- Modifizieren Sie die Layoutdatei so, dass das `TextView`-Element vertikal zentriert ist. Der `Button` wird wieder automatisch unter das `TextView`-Element gesetzt. **Wiederholung:** Der Android Developer Bereich enthält nützliche Dokumentation zu vielen Android Konzepten. Nehmen Sie für diesen Aufgabenteil zum Beispiel folgendes Material zur Hilfe: [Layout Resource Guide](#) und [Relative Layout Guide](#).

1.5 Eine weitere Activity erstellen

Es wird eine neue Activity erstellt, die es dem Nutzer ermöglicht, seinen Namen zu ändern.

- Fügen sie eine weitere Activity *ChangeUserActivity.java* zum Projekt hinzu. Diese Activity soll ein `LinearLayout` mit einem `EditText`-Element und einem `Button` enthalten. Vergessen Sie nicht, die Activity im Manifest einzutragen!
- Fügen Sie dem `EditText`-Element das Attribut `android:hint` mit dem Wert "New Name..." hinzu. Der `Button` soll den Text "OK" anzeigen. Er soll unter dem `EditText`-Element platziert sein. Als Ergebnis der `ChangeUserActivity` soll der Inhalt des `EditText`-Elementes zurückgegeben werden, wenn der Nutzer dies mit dem OK-Button bestätigt. Ist das `EditText`-Element bei Betätigung des Buttons leer, so soll eine Nachricht mit dem Text "Please use a valid user name!" erscheinen.
- Fügen Sie der `MainActivity` einen weiteren `Button` mit dem Text "Change name..." hinzu, der unter dem `TextView`-Element und rechts vom bereits bestehenden `Button` platziert sein soll. Bei Betätigung dieses Buttons soll die `ChangeUserActivity` gestartet werden.
- Implementieren Sie die `onActivityResult`-Methode der `MainActivity` und reagieren Sie auf ein Ergebnis der `ChangeUserActivity`, indem Sie den Username ändern und das `TextView`-Element aktualisieren.

1.6 Zusatzaufgabe: SharedPreferences

- Im Moment wird der gespeicherte Name beim Beenden der MainActivity zurückgesetzt. Setzen Sie sich mit [SharedPreferences](#) auseinander und verwenden Sie diese, um den Nutzernamen über die Beendigung der Activity hinaus zu speichern.