

Android-Apps programmieren

Professionelle App-Entwicklung
mit Android Studio 4

Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

Ihr mitp-Verlagsteam



Neuerscheinungen, Praxistipps, Gratiskapitel,
Einblicke in den Verlagsalltag –
gibt es alles bei uns auf Instagram und Facebook



[instagram.com/mitp_verlag](https://www.instagram.com/mitp_verlag)



[facebook.com/mitp.verlag](https://www.facebook.com/mitp.verlag)

Eugen Richter

Android-Apps programmieren

**Professionelle App-Entwicklung
mit Android Studio 4**



mitp

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-7475-0217-4

3. Auflage 2021

www.mitp.de

E-Mail: mitp-verlag@sigloch.de

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2021 mitp Verlags GmbH & Co. KG, Frechen

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Janina Bahlmann, Sabine Janatschek

Sprachkorrektur: Claudia Fluor

Covergestaltung: Christian Kalkert

Coverbild: © Diana Richter, www.diana-richter.com

Satz: III-satz, www.drei-satz.de

Inhaltsverzeichnis

1	Einleitung	11
1.1	An wen richtet sich dieses Buch?	11
1.1.1	Voraussetzungen	11
1.2	Technischer Stand	11
1.3	Was finden Sie in diesem Buch?	12
1.4	Was behandelt dieses Buch nicht?	13
1.5	Konventionen	13
1.5.1	XML-Dateien	13
1.5.2	Java-Dateien	14
1.6	Danksagung	14
1.7	Änderungen in der 3. Auflage	15
2	Grundlagen	17
2.1	Entwicklungsumgebung	17
2.1.1	Historie	17
2.1.2	Installation	18
2.1.3	Erster Start von Android Studio	26
2.1.4	Struktur der Entwicklungsumgebung	32
2.1.5	Build-System Gradle	38
3	Anlegen einer neuen App	41
3.1	Projektanlage	41
3.1.1	Ein neues Projekt starten	43
3.2	Ausführen der App im Emulator	46
3.2.1	Neues AVD anlegen	47
3.2.2	Starten der App	52
3.3	App-Bausteine	54
3.3.1	Manifest	55
3.3.2	Activity	56
3.3.3	Fragment	56
3.3.4	Ressourcen	56
3.3.5	Layout	57

3.4	Layout-Erstellung	77
3.4.1	Layout erstellen	78
3.4.2	Ressourcen für die Texte	85
3.4.3	Ressourcen und Spezialisierungen	100
4	Basis App-Logik	115
4.1	Template-Pattern	116
4.2	Logik der Zeiterfassungs-App.	118
4.2.1	Suche der Oberflächenelemente	118
4.2.2	Interaktionen des Benutzers verarbeiten	121
4.2.3	Formatierung der Ausgabe	125
4.3	Fehlersuche (Debuggen)	129
5	Datenbank – SQLite.	133
5.1	Überblick über die Datenbanken unter Android.	133
5.1.1	SQLite	133
5.1.2	Alternativen zu SQLite	135
5.2	Datenbank definieren	137
5.2.1	Entwurf der Datenbank-Struktur.	137
5.2.2	Hilfsmittel für SQLite	140
5.3	Anlegen der Datenbank mit Room.	145
5.3.1	Abhängigkeiten zur Room-Bibliothek hinzufügen	146
5.3.2	Eine Entität hinzufügen	148
5.3.3	Klasse für Datenzugriff anlegen	150
5.4	Auf die Datenbank zugreifen	152
5.4.1	Erstellen der eigenen Application-Klasse	154
5.4.2	Erstellen eines Executors	155
5.4.3	Datenbank als Singleton	158
5.4.4	Optimierung für den Datenbankzugriff.	160
5.5	Überprüfung der Daten.	163
5.6	Automatische Konvertierung mit Room	165
5.6.1	Datenklasse (Entity) anpassen	165
5.6.2	Logik für den Start-Button	169
5.6.3	Logik für den Beenden-Button	171
5.7	Laden und Validieren der Daten.	173
5.7.1	Aufräumen in der Klasse	175
5.7.2	UI-Optimierung	179
5.8	Zusammenfassung	180

6	Navigation	181
6.1	Menü-Ressourcen	181
6.1.1	Menütypen	182
6.1.2	Menü anlegen	183
6.1.3	Menü einbinden	187
6.1.4	Auf Menü-Aktionen reagieren	189
6.2	Navigation unter Android	190
6.2.1	Implizite Intents	190
6.2.2	Explizite Intents	191
6.3	Activity für die Auflistung	193
6.3.1	Erstellen der Layouts	194
6.3.2	Erstellen des Adapters	196
6.3.3	Anbinden der Daten an die Liste	203
6.3.4	Optimierung der Auflistung	207
7	Dialoge	215
7.1	Dialoge nutzen	215
7.1.1	Löschen eines Eintrags aus der Liste	217
7.2	Daten mit Dialogen bearbeiten	225
7.2.1	Activity mit Parametern	226
7.2.2	Bearbeitung der Daten in Dialogen	231
8	Datenbank Erweiterung und Migration	247
8.1	Version der Datenbank als Snapshot speichern	247
8.2	Neue Spalte anlegen und migrieren	250
8.2.1	Erweiterung des Datenobjekts	250
8.2.2	Migration der neuen Datenbankversion	251
8.2.3	Pausenspalte anlegen	252
8.3	Inhalt der neuen Felder in die Datenbank speichern	256
9	Hintergrundprozesse und Berechtigungen	261
9.1	Export der Daten als CSV-Datei	261
9.1.1	Berechtigungen	263
9.1.2	Schreiben der Daten als CSV-Datei	267
9.1.3	Fortschrittsanzeige für den Export	273
9.1.4	IntentService	277
9.2	Internet-Zugriff	285
9.2.1	Internetseiten in der App anzeigen	285

9.2.2	Zugriff auf REST-Services	287
9.2.3	Download der Daten aus dem Internet im Hintergrund.	289
9.2.4	JSON-Daten mit Bordmitteln auslesen.	293
9.2.5	JSON-Daten mit gSON auslesen.	295
9.2.6	Generieren einer HTML-Seite.	298
9.2.7	Darstellen in einer Liste.	301
9.2.8	OkHttp als Http-Client.	307
10	App-Optimierungen	309
10.1	Storage Access Framework (SAF).	309
10.1.1	Anlegen einer neuen Datei mit SAF.	310
10.1.2	Exporter erweitern	313
10.2	Android-Binding	317
10.2.1	Projekt für Binding bereit machen	318
10.2.2	Arbeiten mit Bindings	321
10.3	Veröffentlichen der fertigen App	340
10.3.1	App-Icon erstellen.	341
10.3.2	Signierung mit Zertifikat.	344
10.3.3	Veröffentlichung.	348
11	Automatisierte Tests	349
11.1	MonkeyRunner	349
11.2	Unit-Tests	351
11.2.1	Testen des Ladens aus dem gespeicherten Zustand	352
11.2.2	Testen mit Mocks	356
11.3	Android-Tests.	362
11.4	Oberflächen-Tests	366
11.4.1	Optimierung des Espresso-Codes	370
12	Schlusswort und Ausblick.	375
12.1	Beste Anlaufstellen für die erste Suche	375
12.2	Themen, die in diesem Buch (noch) nicht behandelt wurden	376
12.2.1	Kotlin	376
12.2.2	Bluetooth	377
12.2.3	Android Architecture Patterns.	377
12.2.4	Android Wear/Android TV/Android Auto/Android IoT.	377
12.2.5	Monetarisierung	377
12.3	Verbesserungsvorschläge/Fehler	378

	Anhang	379
A.1	Glossar	379
A.2	Installation von HAXM	382
	A.2.1 Voraussetzungen	383
	A.2.2 Installation	384
A.3	Smartphone oder Tablet als Entwickler-Gerät einrichten	386
A.4	Vorhandenen Quellcode in Android Studio öffnen	388
A.5	Tastatur-Kürzel	392
	A.5.1 Suche Aktion	392
	Stichwortverzeichnis	393

Einleitung

Im mobilen Segment hat Android mittlerweile einen Marktanteil in Deutschland von über 73% (Quelle: Statista 2020). An dieser großen Verbreitung möchten Sie sicherlich teilhaben, ob als Hobby-Programmierer oder in einem Unternehmen. Mit diesem Buch finden Sie einen praktischen Einstieg in die Android-Programmierung.

1.1 An wen richtet sich dieses Buch?

Dieses Buch ist für alle geschrieben, die in die Android-Entwicklung einsteigen möchten, richtet sich aber auch an App-Programmierer, die alte Kenntnisse auffrischen wollen.

1.1.1 Voraussetzungen

Die wichtigste Voraussetzung für die Lektüre sind grundlegende Kenntnisse in einer objektorientierten Programmiersprache, im Idealfall in Java. Aber auch ohne Java-Kenntnisse (zum Beispiel als C#-Entwickler) finden Sie sich sehr schnell zurecht, wenn objektorientierte Programmierung für Sie kein Fremdwort ist.

Weiterhin ist ein gutes Verständnis des Dateiformats »XML« notwendig, da Android die Layouts und einige andere Ressourcen in XML beschreibt.

Die letzte Voraussetzung ist ein grundlegendes Verständnis von relationalen Datenbanken. Keine Angst, Sie müssen in diesem Buch keine »CREATE«-Anweisung aus dem Kopf schreiben.

1.2 Technischer Stand

Dieses Buch wurde grundlegend für die Android-Studio-Version 4.0 geschrieben, die zum Druckzeitpunkt aktuell ist. Wenn Sie eine neuere Version verwenden, werden eventuell Teile der Benutzeroberfläche anders aussehen als die Screenshots im Buch. Das grundlegende Vorgehen sollte sich aber nicht ändern. Der

Quellcode für die App aus dem Buch wird kompatibel zu der aktuellen Android-Studio-Versionen gehalten.

Aktualisierungen

Sollten neue Android-Studio-Versionen grundlegende Änderungen enthalten, so dass die Inhalte im Buch nicht mehr nachvollzogen werden können, werden korrigierte Kapitel auf der Projektseite veröffentlicht. Sollten Sie eine solche Änderung feststellen, schreiben Sie bitte an android-buch@webducer.de, damit ich die Korrekturen vornehmen und für alle verfügbar machen kann.



wdur1.de/ab3-projekt

Quellcode

Den Quellcode für die Aufgaben im Buch finden Sie bei dem Anbieter Bitbucket. Der Code ist in Kapitel unterteilt und führt Sie von einer Aufgabe zur nächsten – vom Ausgangscode bis zur Lösung der gestellten Aufgaben.



wdur1.de/ab3-code

1.3 Was finden Sie in diesem Buch?

Das Buch ist in Form eines praktischen Workshops aufgebaut. Es startet mit der Installation der Entwicklungsumgebung auf den wichtigsten Betriebssystemen und vermittelt anschließend durch die Entwicklung einer praxistauglichen App zahlreiche Fertigkeiten für die Android-Entwicklung:

- UI-Design mit XML (Designer und Code)
- Logik in Java
- Zugriff auf Datenbanken
- Zugriff auf Internet-Ressourcen (APIs)
- Hintergrundprozesse
- Dialoge
- Binding
- und viele andere

Am Ende entsteht eine fertige App, die in einem Store oder auf der eigenen Homepage veröffentlicht werden kann. Dabei werden alle Schritte durch den Quellcode zu dem jeweiligen Fortschritt unterstützt, den Sie auch auf der Bitbucket-Seite des Projekts zum Download finden: <https://wdur1.de/ab3-code>

1.4 Was behandelt dieses Buch nicht?

Auf der Google I/O wurde die Programmiersprache Kotlin als offiziell unterstützte Sprache angekündigt. Dieses Buch behandelt Kotlin aus folgenden Gründen in der aktuellen Auflage (noch) nicht:

- Die Sprache ist relativ neu und nur wenige Entwickler kennen diese, insbesondere außerhalb des Android-Universums. Programmierung für ein neues mobiles System und zusätzlich eine neue Programmiersprache zu lernen, wäre nicht anfängerfreundlich.
- An den meisten Universitäten und Berufsschulen wird Java als »Ausbildungssprache« unterrichtet. Damit ist auch der Einstieg in die Android-Entwicklung für die zukünftigen Entwickler in Java einfacher.
- Die meisten Beispiele im Netz (aber auch die offiziellen von Google) für Android zeigen den Java-Code. Aus diesem Grund ist es für Android-Einsteiger besser, mit Java zu starten. Ich hoffe, dass sich Kotlin mit der Zeit mehr durchsetzt, da die Sprache im Vergleich zu Java wirklich deutlichen Mehrwert bietet.
- Kennt man Android mit Java, ist der Umstieg zu Android mit Kotlin relativ einfach. Es gibt sehr gute Bücher über Kotlin, die es ermöglichen, die deutlichen Vorteile der Sprache zu erlernen und einzusetzen (z.B.: »Kotlin – Einstieg und Praxis« von Karl Szwilius – ISBN: 9783958458536).

1.5 Konventionen

1.5.1 XML-Dateien

- Dateiname: Kleinschreibung mit Worttrennung durch »_«.
Beispiel: `activity_main.xml`
- Ressource-Namen (für IDs, Strings, Abstände usw.):
»CamelCase«-Schreibweise.
Beispiel: `<string name="LabelName">Name:</string>` oder
`<TextView android:id="@+id/FirstLabel" />`

1.5.2 Java-Dateien

- Dateiname/Klassenname: »CamelCase«-Schreibweise.
Beispiel: DbHep1er
- Nicht private Konstanten: »UPPER Case«-Schreibweise Worttrennung durch »_«.
Beispiel: `public final static String ID_KEY = "TimeDataIdKey";`
- Private Konstanten: »UPPER Case«-Schreibweise mit »_«-Zeichen als Präfix und Worttrennung durch »_«.
Beispiel: `private final static int _LOADER_ID = 150;`
- Private Klassenvariablen: »pascalCase«-Schreibweise mit »_«-Zeichen als Präfix.
Beispiel: `private DateFormat _dateTimeFormatter = null;`
- Methodennamen: »pascalCase«-Schreibweise.
Beispiel: `private void saveEndDateTime() { ... }`
- Parameternamen und lokale Variablen: »pascalCase«-Schreibweise.
Beispiel:

```
private void setStartDate(Calendar startDate) {  
    ...  
    String startDateString = _dateTimeFormatter.format(startDate.getTime());  
    ...  
}
```

1.6 Danksagung

Dieses Buch würde nicht da sein, wenn mich nicht einige Menschen unterstützen würden.

In erster Linie gilt mein größter Dank meiner Familie, die oft genug auf mich verzichten muss, damit ich an dem Buch arbeiten kann.

Weiterhin gilt mein Dank meiner Lektorin, Frau Bahlmann, die das Buch mit Ideen für eine bessere Strukturierung und einen optimalen Sprachgebrauch zugänglicher gemacht hat. Auch ein Dank an meine erste Lektorin, Frau Janatschek, die die Geduld hatte, zwei Jahre auf die erste Veröffentlichung zu warten, und meinen technischen Schreibstil in einen leserlichen verwandelte.

Ein weiterer Dank gilt der Münchner Volkshochschule, in der ich seit 2009 unterrichte und mein Können bei der Arbeit mit Lernwilligen verfeinern darf.

Ohne die Leser wäre dieses Buch sinnlos. An dieser Stelle möchte ich mich bei allen Lesern bedanken, die Feedback zu diesem Buch (über Amazon-Rezensionen oder direkt per E-Mail an android-buch@webducer.de) an mich geschrieben haben. Ich habe in dieser dritten Auflage das Feedback umgesetzt und freue mich auf weiteres.

1.7 Änderungen in der 3. Auflage

Im Vergleich zur zweiten Auflage wurde Folgendes geändert:

- Aktualisierung der Installationen unter allen Betriebssystemen auf Android Studio 4.0
- Aktualisierung aller Screenshots
- **Neu:** Nutzung von ROOM für den Zugriff auf die Datenbank
- **Entfernt:** Content Provider
- **Neu:** Nutzung von AndroidX-Bibliotheken
- **Neu:** Links zu Quellen als QR-Code
- Bessere Gliederung
- Verweise auf den Quellcode

Grundlagen

In diesem Kapitel befassen wir uns mit den Grundlagen für die Android-Programmierung. Dazu werden Sie eine Entwicklungsumgebung aufsetzen und die Grundbausteine der Android-Programmierung kennenlernen.

2.1 Entwicklungsumgebung

Bei der Programmierung ist es wichtig, nicht nur die Programmiersprache zu kennen, sondern auch die Werkzeuge. In diesem Abschnitt werden wir daher die Entwicklungsumgebung installieren und einrichten.

Android Studio ist seit 2014 die offizielle Entwicklungsumgebung für Android und basiert auf der nicht nur unter Java-Entwicklern sehr beliebten IDE (englische Abkürzung für **I**ntegrated **D**evelopment **E**nvironment) von JetBrains IntelliJ IDEA. Diese Entwicklungsumgebung für auf »Java Virtual Machine« (kurz JVM) basierende Sprachen unterstützt den Programmierer bei der Entwicklung.

Neben IntelliJ IDEA liefert die Firma JetBrains noch weitere Entwicklungsumgebungen für andere Sprachen wie WebStorm für die Web-Entwicklung, PHP-Storm für PHP-Entwicklung, RubyMine für Ruby on Rails, ReSharper für .Net-Entwickler und mehr.

2.1.1 Historie

Die Android Entwicklung begann mit der Entwicklungsumgebung Eclipse und dem Plugin »Android Developer Tools«.

Android Studio wurde auf der Google I/O 2013 zum ersten Mal präsentiert und als Alpha-Version veröffentlicht. Ein Jahr später, auf der Google I/O 2014, wurde es zum Beta-Status erhoben. Bereits im Dezember 2014 erschien dann die erste finale Version 1.0.

Seit Dezember 2014 ist Android Studio die offizielle Entwicklungsumgebung für die Android-Entwicklung. Es hat Eclipse mit den Android Developer Tools abgelöst. Neuere Funktionen erscheinen seitdem nur noch für Android Studio. Android Devel-

oper Tools werden zwar durch die Eclipse Foundation weiterentwickelt, aber nicht mehr von Google selbst.

Zusätzlich zu der Umstellung der Entwicklungsumgebung von Eclipse auf Android Studio wurde auch das Build-System von Ant bei Eclipse auf Gradle unter Android Studio geändert.

Eine kurze Einführung in das Gradle-Build-System finden Sie weiter unten im Abschnitt 2.1.5.

Android Studio – Open Source

Android Studio ist ein Open-Source-Projekt. Der Quellcode kann unter wdurl.de/ab3-as-projekt heruntergeladen werden und Sie können eigene Android-Studio-Versionen damit bauen.

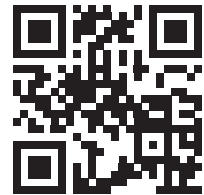


wdurl.de/ab3-as-projekt

2.1.2 Installation

Nun geht es ans Werk. Die aktuelle Version von Android Studio erhalten Sie von der offiziellen Android-Entwicklerseite (developer.android.com/studio/).

Im Normalfall erkennt der Browser das aktuelle Betriebssystem und schlägt die passende Datei für den Download vor. Wenn dies nicht der Fall ist oder Sie die Installationsdatei für einen anderen Rechner benötigen, finden Sie diese auf der oben genannten Seite unter »Download Options«.



wdurl.de/ab3-as

Android Studio wird für die gängigen Betriebssysteme Windows, macOS und Linux angeboten.

Für Windows stehen mehrere Versionen für den Download zur Verfügung:

- Android Studio mit Installer (Standard 64 Bit)
- Android Studio ohne Installer als Zip-Archiv (64 Bit)
- Android Studio ohne Installer als Zip-Archiv (32 Bit)

Nutzen Sie den Installer. Andere Installationsmethoden sind nur in Spezialfällen erforderlich.

Das SDK (Software Development Kit) wird erst beim ersten Start von Android Studio nachgeladen.

Hardware-Anforderungen

Die aktuellen Hardware-Anforderungen finden Sie unter :

developer.android.com/studio/index.html#Requirements.



wdurl.de/ab3-as-req

Als Hardware-Voraussetzungen sind ein moderner Rechner mit mindestens 4 GB RAM, 2 GB freiem Speicherplatz und 1280 x 800-Auflösung notwendig. Mit dieser Minimalanforderung wird die App-Entwicklung aber nicht wirklich Spaß machen. Für flüssige Android-Entwicklung empfehle ich einen modernen Rechner mit mindestens folgender Ausstattung:

- Core i5/i7 (ab 6. Generation) mit aktivierter Virtualisierungstechnik VT-x, EM64T und XD oder vergleichbarem AMD Prozessor
- 8 GB RAM (für Android Virtual Devices-AVDs)
- SSD statt Festplatte (rotierende Scheiben/HDD)
- 24“-Bildschirm mit Full-HD-Auflösung (1920 x 1080)

Die Virtualisierungstechnik ist notwendig, um schnellere Emulatoren installieren zu können (Intel HAXM, Hyper-V, Genymotion und weitere). Die Standard-Emulatoren (ohne Virtualisierungstechnik) sind leider sehr langsam und machen bei der Entwicklung keinen Spaß. Genauerer können Sie in Anhang A.2 »Installation von HAXM« nachlesen.

Installation unter Windows

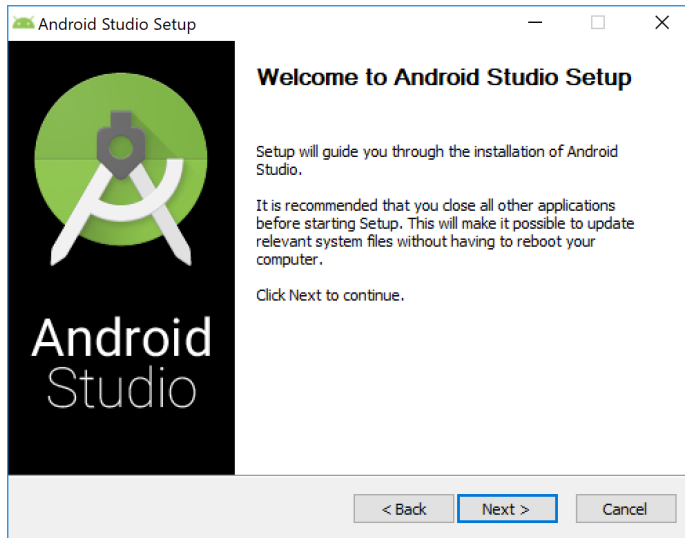


Abb. 2.1: Assistent für die Android-Studio-Installation

Nach dem Download, der fast 1 GB groß ist, kann die Installation starten. Die Installation liefert auch eine Java-Laufzeitumgebung, sodass keine JDK-Installation mehr notwendig ist.

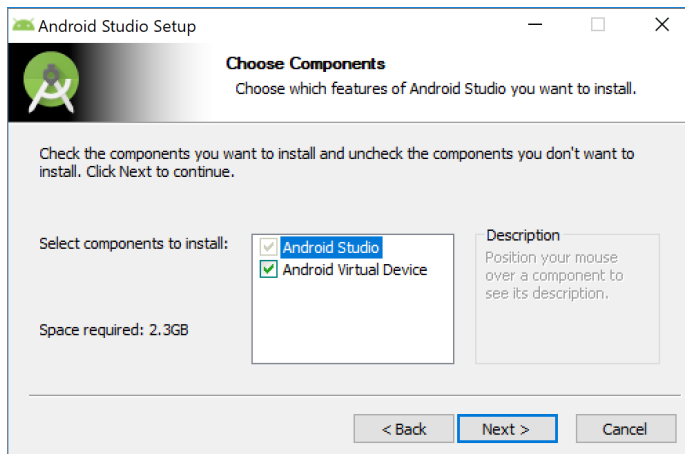


Abb. 2.2: Auswahl der Komponenten bei der Installation

Wenn Sie die Standardeinstellungen bei der Installation beibehalten, installiert das Setup neben Android Studio auch ein Android Virtual Device (kurz AVD). Sie

werden noch weitere AVDs im Laufe des Buchs installieren und deren Funktionen kennenlernen.

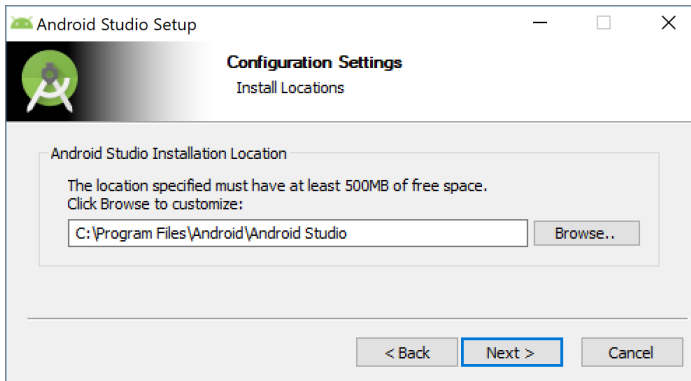


Abb. 2.3: Auswahl des Installationsverzeichnis

Mit dem nächsten Schritt (Abbildung 2.3) können Sie den Pfad für die Installation anpassen, falls Sie einen anderen Speicherort auswählen möchten.

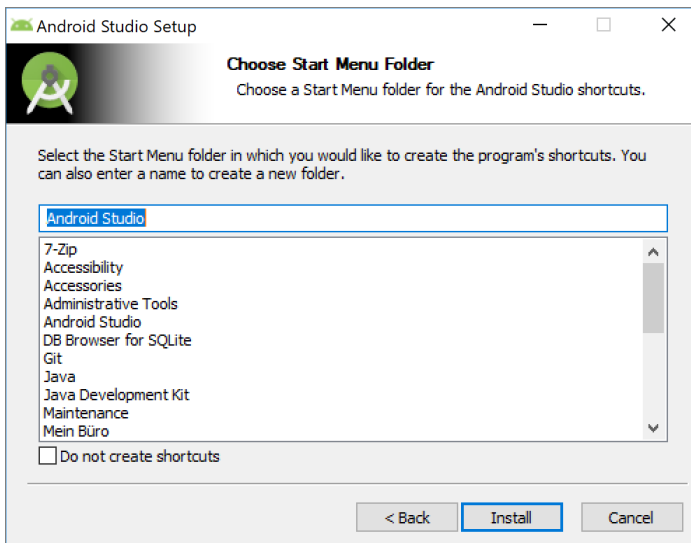


Abb. 2.4: Auswahl des Ordners im Startmenü

In dem vorletzten Schritt (Abbildung 2.4) können Sie einen Ordner für das Startmenü ändern.

Nun startet die Installation, die abhängig vom System einige Minuten in Anspruch nehmen wird (Abbildung 2.5).

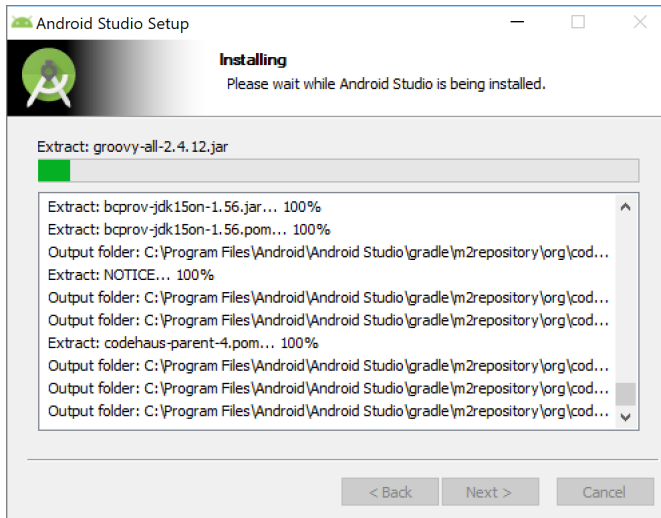


Abb. 2.5: Fortschritt der Installation

Nach der erfolgreichen Installation (Abbildung 2.6) haben Sie die Möglichkeit, Android Studio direkt zu starten. Das sollten Sie auch tun, da die Installation noch nicht abgeschlossen ist. Neben Android Studio muss auch Android SDK (Software Development Kit) installiert werden. Abhängig von den ausgewählten Komponenten werden noch mal ein bis zwei GB an Daten auf Ihren Rechner heruntergeladen.

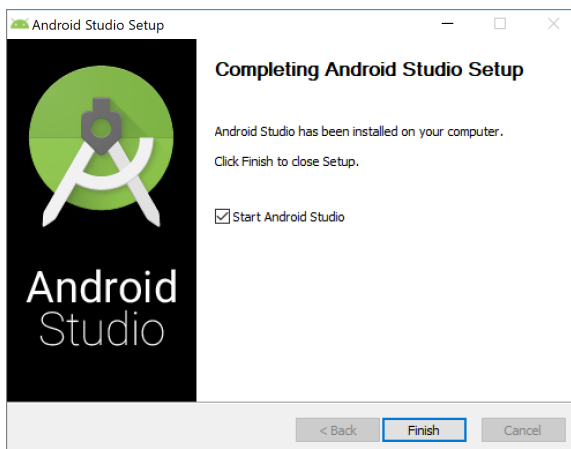


Abb. 2.6: Erfolgreiche Installation von Android Studio

Abhängig von der Windows-Version und der Einstellung der Firewall kann es beim Start zu einer Anfrage kommen, ob Android Studio sich mit dem Internet verbinden darf (für eigene Updates und für Updates von Android SDK). Um auf dem aktuellen Stand zu bleiben, sollten Sie natürlich den Zugriff gewähren.

Den Rest der Ersteinrichtung beschreibe ich nach der macOS- und Linux-Installation in Abschnitt 2.1.3, da diese Schritte unter allen Systemen gleich aussehen.

Installation unter OS X



Abb. 2.7: Installation unter macOS

Wie üblich unter macOS, müssen Sie nur Android Studio aus dem geöffneten Package in den Programmordner verschieben (oder in einen Ordner Ihrer Wahl, Abbildung 2.7). Im Paket befindet sich nur Android Studio. Das Android SDK (Software Development Kit) muss beim ersten Start installiert werden. Zusätzlich zu Android Studio fallen noch einmal 1 bis 2 GB an zusätzlichen Daten an.

Nach Abschluss der Installation und beim Start erscheint als Erstes ein Sicherheitsdialog, da Android Studio nicht über Apple Store installiert wurde. Bitte bestätigen Sie diesen Dialog (es muss nur einmal gemacht werden).

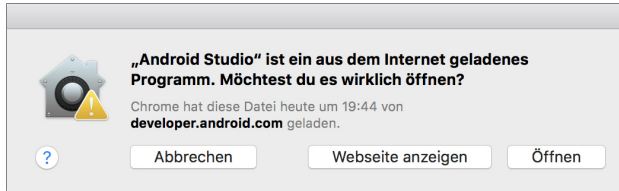


Abb. 2.8: Sicherheitsabfrage beim ersten Start

Die Ersteinrichtung beschreibe ich nach der Linux-Installation im Abschnitt 2.1.3, da diese Schritte unter allen Systemen gleich aussehen.

Installation unter Linux (Ubuntu)

Unter Ubuntu haben Sie zwei Installationsmöglichkeiten:

1. Installation über den integrierten Store. Suchen Sie im Store einfach nach »Android Studio« und installieren Sie es (Abbildung 2.9). Es wird auch eine Verknüpfung im Startmenü automatisch angelegt.
2. Manuelle Installation, die hier beschrieben ist (funktioniert unter allen Linux-Distributionen).

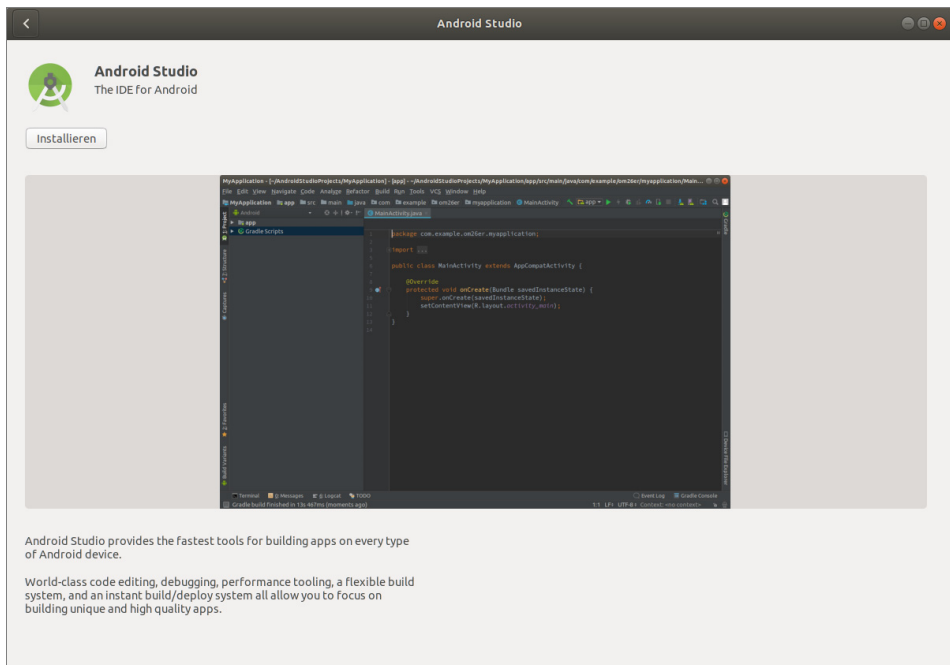


Abb. 2.9: Android Studio in Ubuntu App Store

Unter Linux muss die heruntergeladene komprimierte Datei in einen Ordner entpackt werden. Zum Starten führen Sie im Terminal die Datei `studio.sh` aus dem `bin`-Ordner aus.

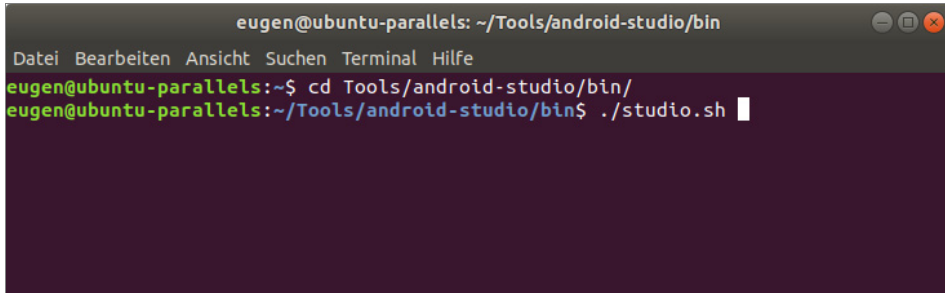


Abb. 2.10: Start von Android Studio über Terminal

Die genauen Anweisungen zum Starten von Android Studio sind auch in der Datei `Install-Linux-tar.txt` beschrieben. Die Datei liegt in dem Installationsverzeichnis.

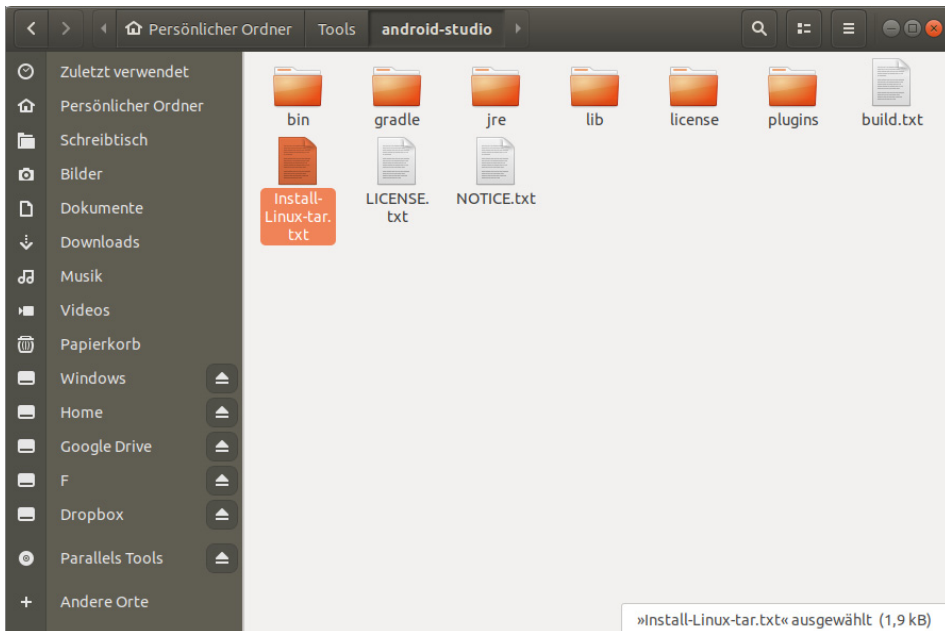


Abb. 2.11: Installationsanweisungen im Installationsordner

Der erste Start (und Einrichtung des SDK) ist unter allen Betriebssystemen fast identisch.

2.1.3 Erster Start von Android Studio

Beim ersten Start erscheint die Meldung zur Übernahme der Einstellungen einer vorherigen Installation (falls vorhanden).

Vorherige Einstellungen übernehmen

Diese Meldung erscheint immer bei größeren Updates, da die Einstellungen in versionsabhängigen Ordnern gespeichert werden (so kann man parallel auch mehrere Versionen von Android Studio haben), zum Beispiel beim Wechsel von Version 3.6.2 auf Version 4.0.0.

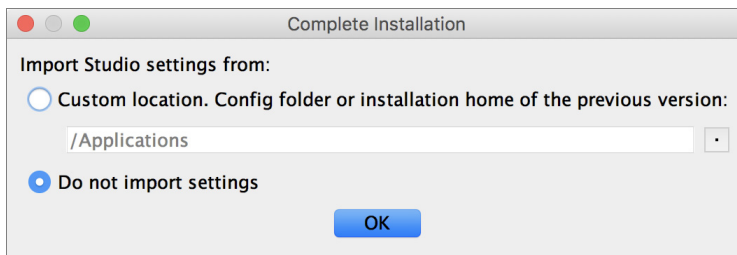


Abb. 2.12: Übernahme der Einstellungen beim ersten Start

Nach dem Dialog zur Übernahme der Einstellungen erscheint der Assistent für die Ersteinrichtung (Abbildung 2.13). Dieser installiert das Android Software Development Kit (kurz Android SDK) und, falls ausgewählt, auch das erste Android Virtual Device (kurz AVD).

Wenn Sie im nächsten Schritt die Installationsmethode »Custom« auswählen (Abbildung 2.14), können Sie die Installation an einigen Stellen anpassen (unter anderem die RAM-Größe für HAXM, wohin SDK installiert wird usw.).

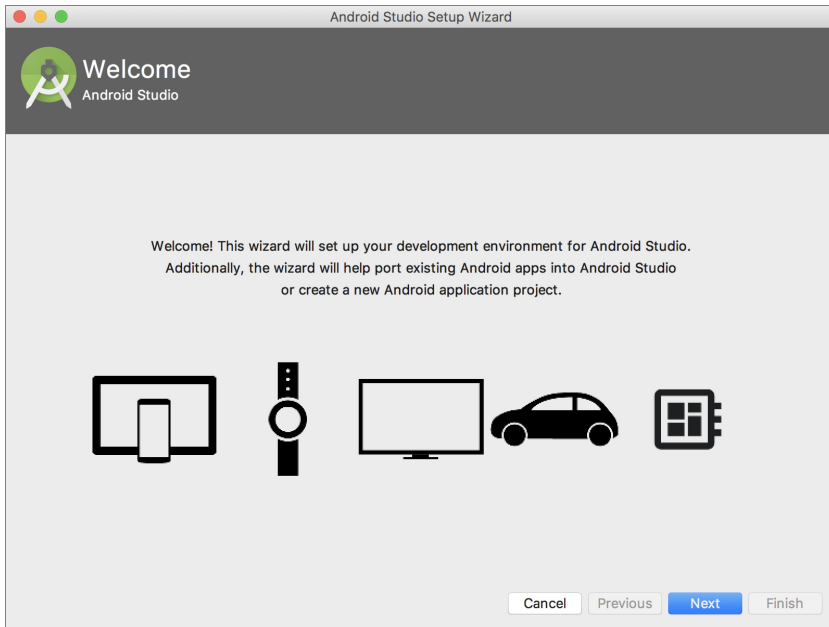


Abb. 2.13: Assistent für den Erststart (hier macOS)

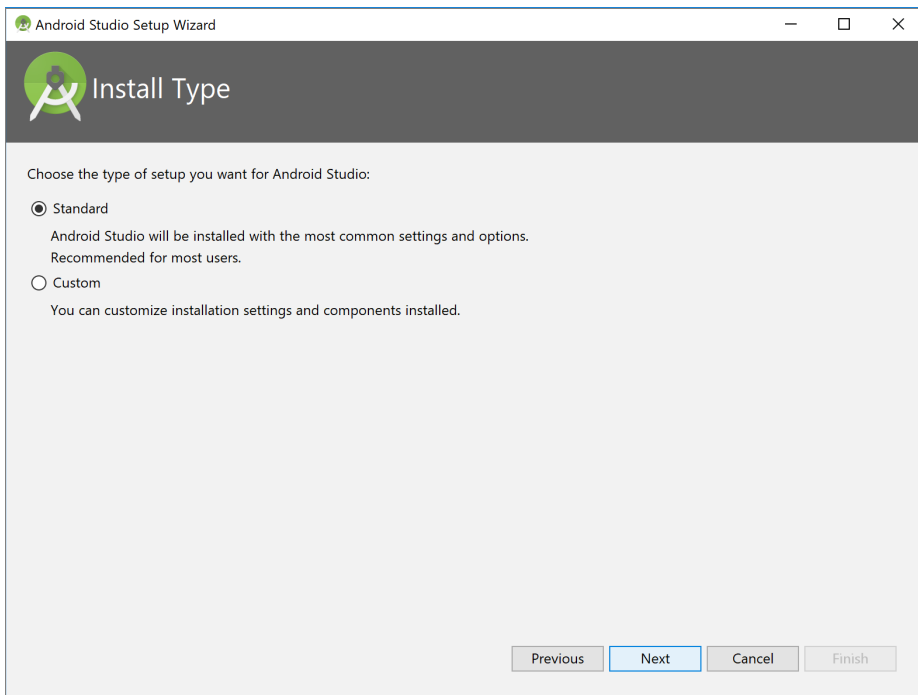


Abb. 2.14: Auswahl des Installationstyps (hier unter Windows)

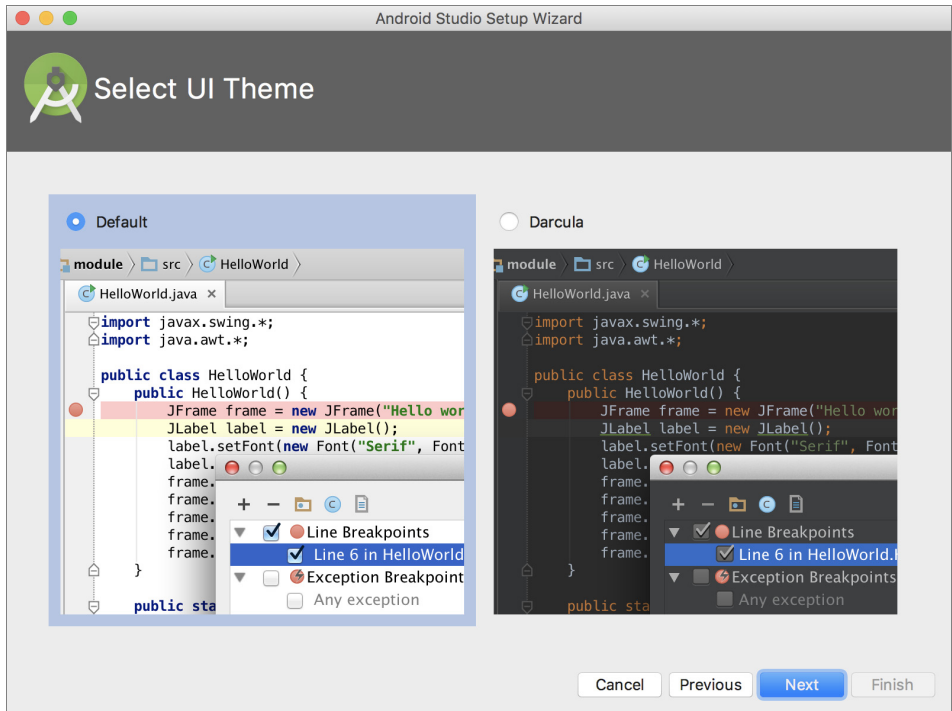


Abb. 2.15: Auswahl des Aussehens (hier unter macOS)

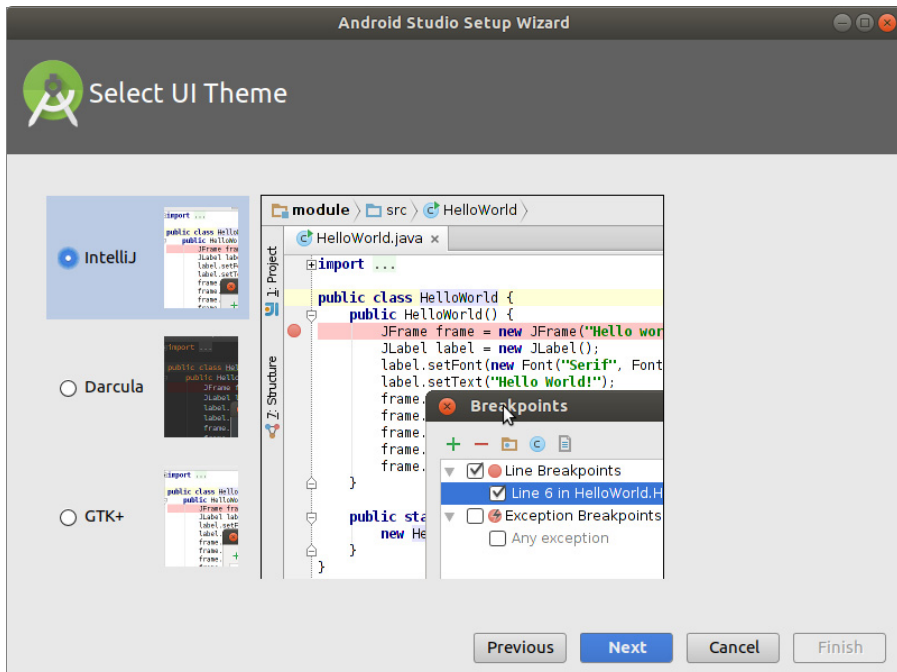


Abb. 2.16: Auswahl des Aussehens (hier Ubuntu)