

Entwicklung mobiler Applikationen

WS2022/2023

Jan Brodhaecker | Oktober 2022

Agenda

- Organisation: Abstimmung Prüfungsleistung
- Android Studio und das Eco-System
- Android Basics - App Components
 - Activities
 - Layouts

Android Studio

- offizielle freie IDE von Google
- vorgestellt im Ende 2014 auf Google I/O
- basiert auf IntelliJ IDEA Community Edition (von JetBrains)
- ersetzt ADT (Android Developer Tools) für Eclipse (deprecated seit Ende 2015)

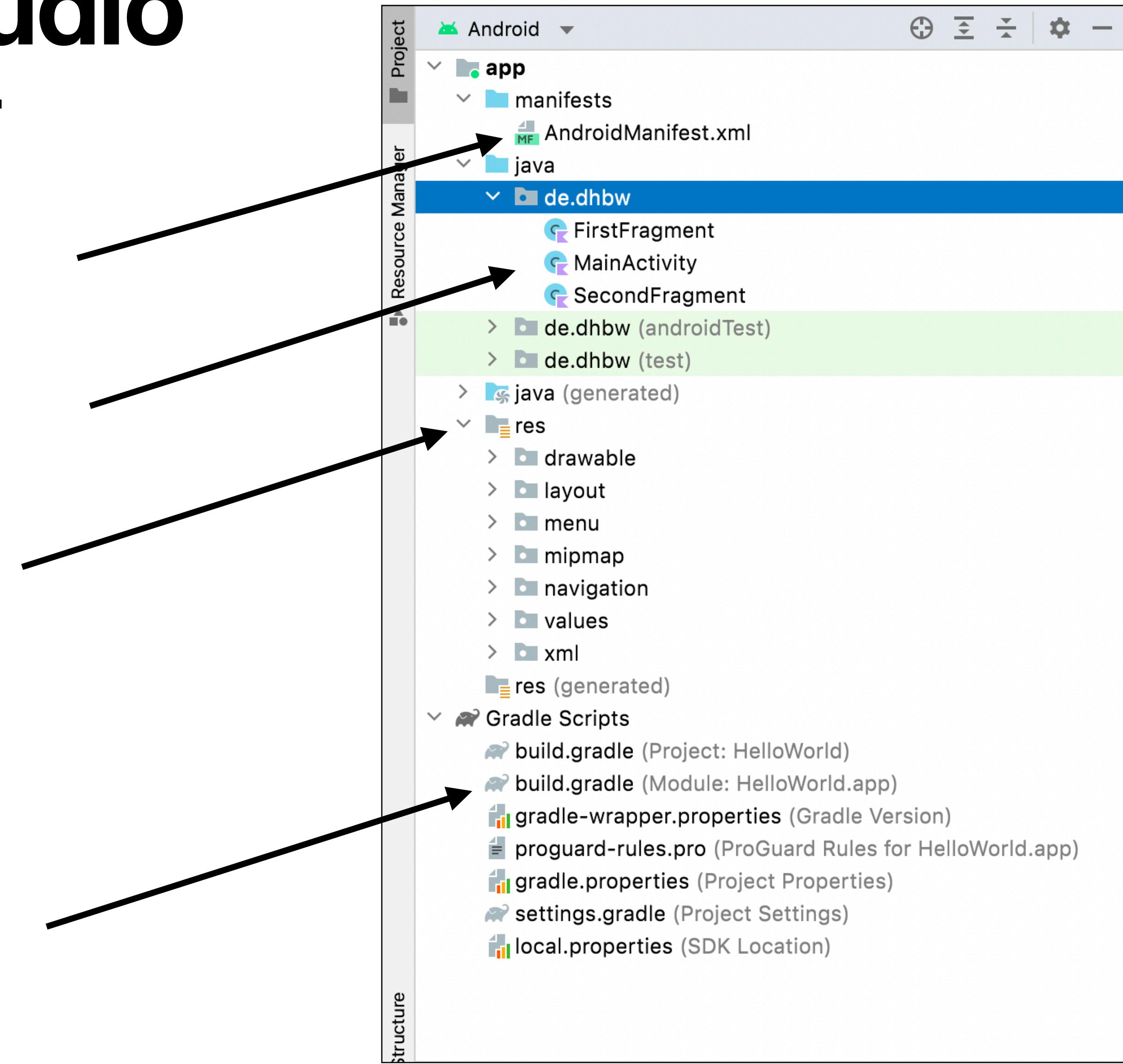
<https://developer.android.com/studio>



Android Studio provides the fastest tools for building apps on every type of Android device.

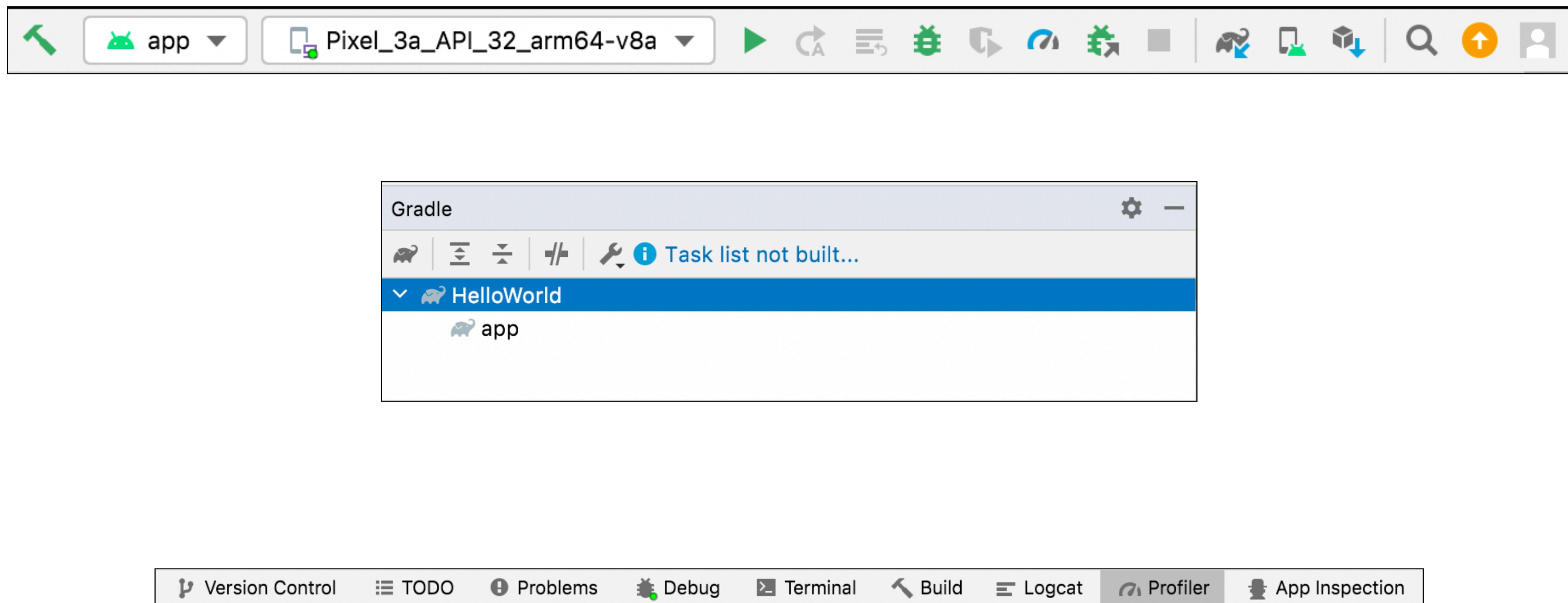
Android Studio

Projekt-Explorer



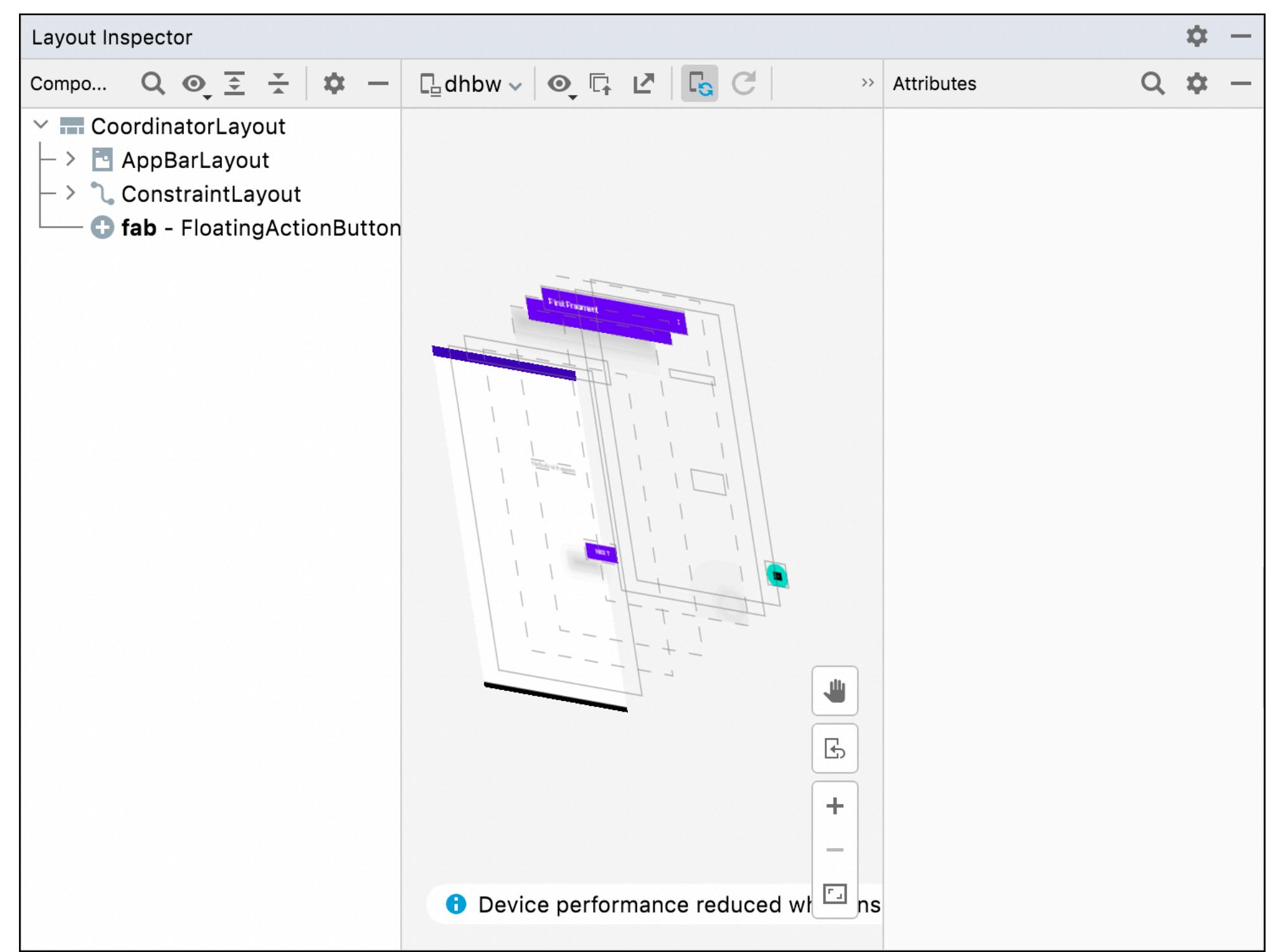
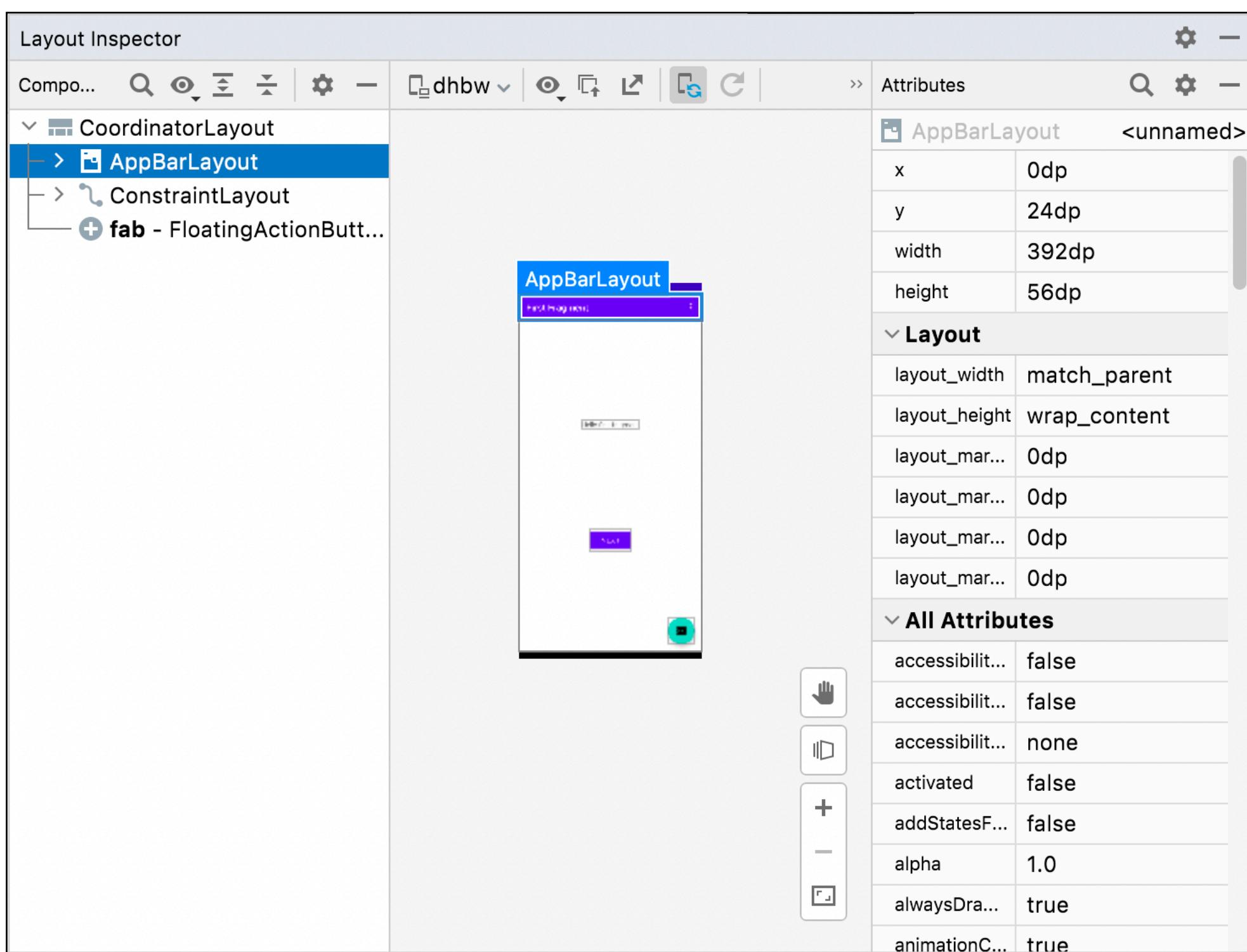
Android Studio

Task-Bar / Gradle / Debugging



Android Studio

Layout-Inspector



Android Studio

Device File Explorer

Device File Explorer

Emulator Pixel_3a_API_32_arm64-v8a Android 12, API 32

Name	Permissions	Date	Size
> acct	drwxr-xr-x	2021-12-20 16:07	4 KB
> apex	drwxr-xr-x	2022-10-10 20:03	1,1 KB
> bin	lrw-r--r--	2021-12-20 16:53	11 B
> cache	drwxrwx---	2021-12-20 16:07	4 KB
> config	drwxr-xr-x	1970-01-01 01:00	0 B
> d	lrw-r--r--	2021-12-20 16:53	17 B
> data	drwxrwx--x	2022-08-08 21:39	4 KB
> data_mirror	drwx-----	2022-10-10 20:03	120 B
> debug_ramdisk	drwxr-xr-x	2021-12-20 16:07	4 KB
> dev	drwxr-xr-x	2022-10-10 20:03	2,7 KB
> etc	lrw-r--r--	2021-12-20 16:53	11 B
> linkerconfig	drwxr-xr-x	2022-10-10 20:03	240 B
> lost+found	drwx-----	2021-12-20 16:53	16 KB
> metadata	drwxr-xr-x	2022-06-06 18:37	4 KB
> mnt	drwxr-xr-x	2022-10-10 20:03	320 B
> odm	drwxr-xr-x	2021-12-20 16:07	4 KB
> odm_dlkm	drwxr-xr-x	2021-12-20 16:07	4 KB
> oem	drwxr-xr-x	2021-12-20 16:07	4 KB
> postinstall	drwxr-xr-x	2021-12-20 16:07	4 KB
> proc	dr-xr-xr-x	2022-10-10 20:03	0 B
> product	drwxr-xr-x	2021-12-20 16:51	4 KB
> second_stage_resources	drwxr-xr-x	2021-12-20 16:07	4 KB

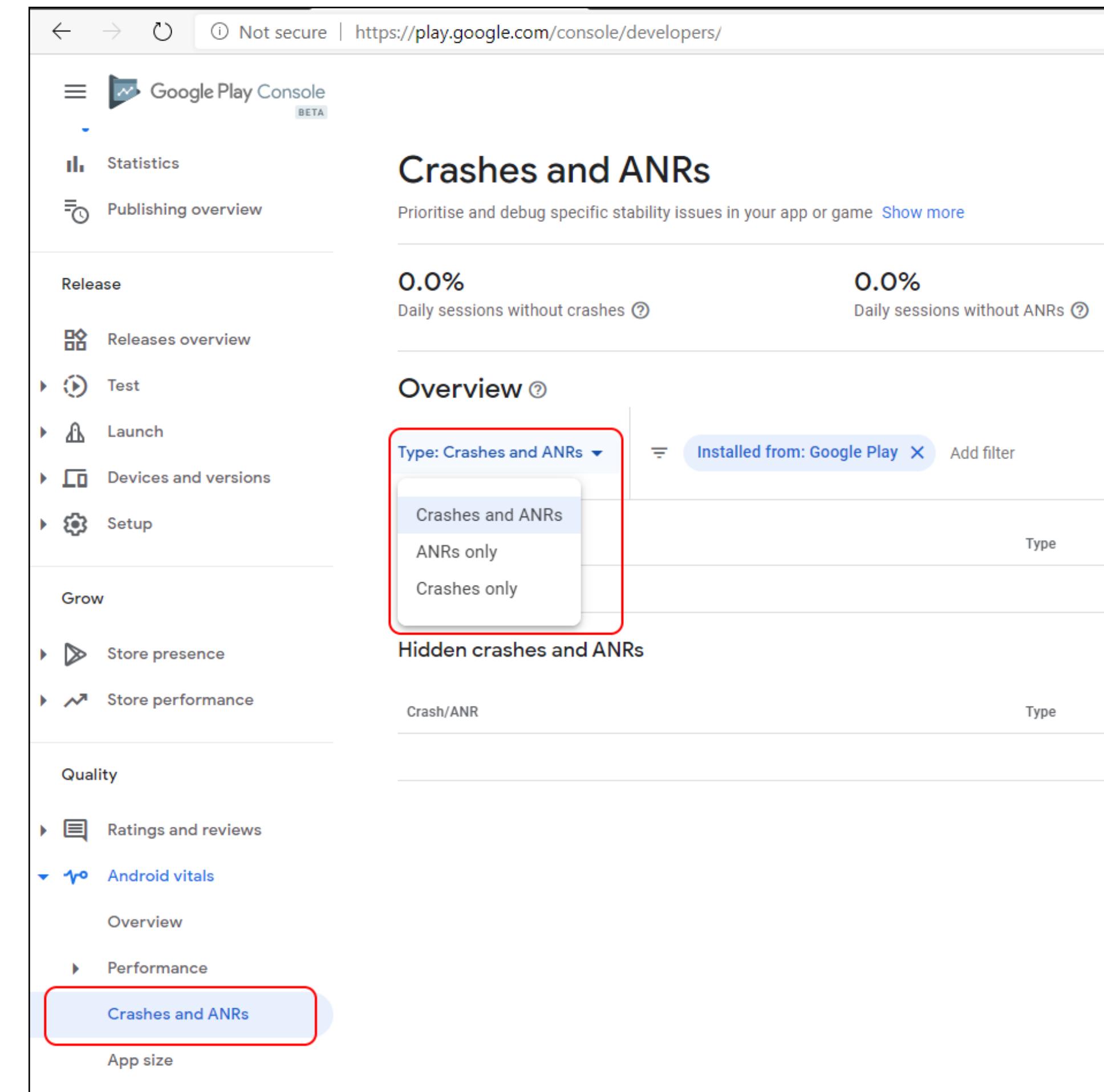
Android Studio

Exkurs: Logging

- Logs helfen dabei Fehler nachzuvollziehen, auch wenn die App bereits ausgeliefert wurde

<https://developer.android.com/topic/performance/vitals/anr>

- Crash-Logs / Android Not Responding-Fehler (ANRs) können über die Google Play Konsole eingesehen werden

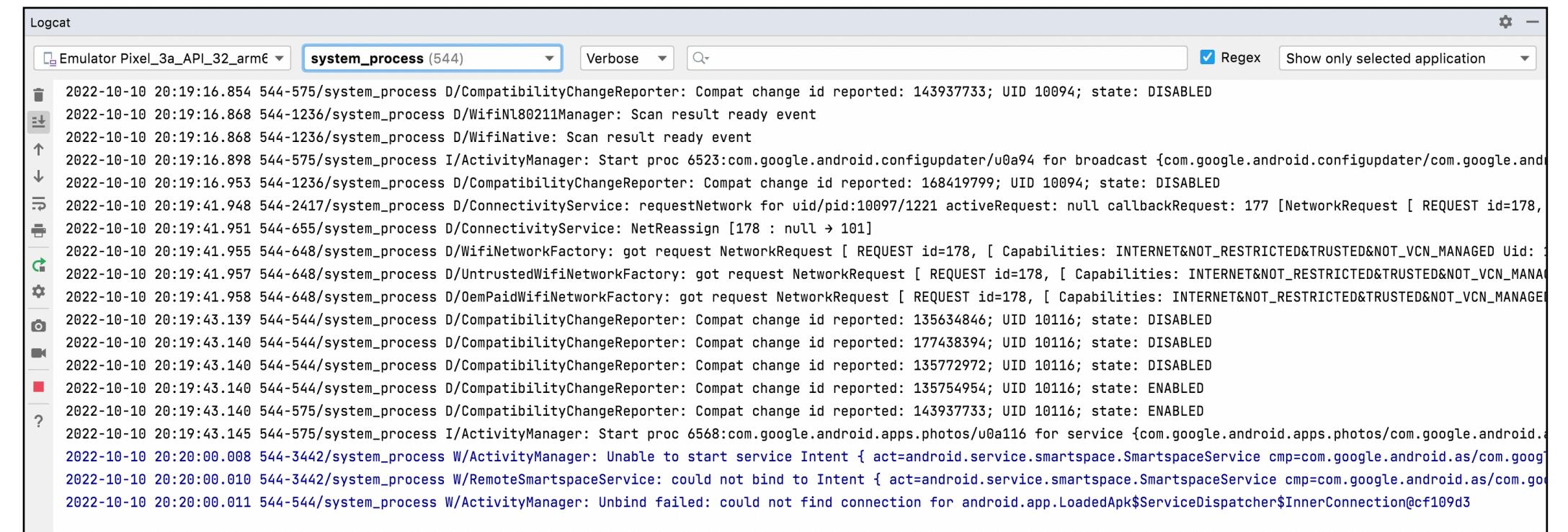


<https://stackoverflow.com/a/63612270>

Android Studio

Exkurs: Logging

- Logcat hilft (App-)Logs zu lesen/filtern
- Einsatz verschiedener Log-Level sinnvoll
 - verbose, debug, info, warning, error
- als TAG eignet sich bspw. der Name der jeweiligen Klasse



The screenshot shows the Android Studio Logcat tool. The dropdown menu at the top is set to 'Emulator Pixel_3a_API_32_arm6'. The current log entry is highlighted in blue and shows a compatibility change report. The log window displays numerous entries from the system process, with timestamps ranging from 2022-10-10 20:19:16 to 2022-10-10 20:20:00. The log entries include various system components like CompatibilityChangeReporter, WiFiNative, ActivityManager, ConnectivityService, and WiFiNetworkFactory, reporting events such as scan results, network requests, and service starts.



```

1  override fun onPause() {
2      super.onPause()
3      Log.d(MainActivity::class.simpleName, "onPause() was called.")
4  }
5

```

The screenshot shows a snippet of Java code within an Android application. It defines an overridden `onPause()` method. Inside the method, there is a call to `Log.d` with the tag `MainActivity::class.simpleName` and the message `"onPause() was called."`. Above the code, there are three small colored circles (red, yellow, green) which are part of the Android Studio interface for navigating between tabs or files.

Android Studio

Exkurs: Gradle

- Open-Source basierend auf Groovy JVM
- Build-Automatisierung / Tool
- Dependency-Management
- Android Studio Build System basiert auf Gradle
- Gradle vs. Maven (Flexibilität, Performance, User-Experience)
<https://gradle.org/maven-vs-gradle/>



<https://tinyurl.com/4fsytc5u>

Android Studio

Exkurs: Gradle

- In Gradle besteht ein Build aus ...
 - ... einem oder mehreren Projekten (*Projects*)
 - war/jar/zip Dateien erzeugen
 - ... ein Projekt besteht aus einer oder mehreren Aufgaben (*Tasks*)
 - „*single piece of Work*“
 - weitere Infos: <https://www.baeldung.com/gradle>

```
1 // build.gradle
2
3 task hello {
4     doLast {
5         println 'Hello World!'
6     }
7 }
8
9 // >> gradle -q hello
```

Android Studio

Exkurs: Gradle - Dependency Tree



The screenshot shows a terminal window with the following text:

```
1 // without modules
2 >> gradle dependencies
3
4 // for android
5 >> gradle app:dependencies
```

Android Studio

Exkurs: Gradle - Dependency hinzufügen

https://docs.gradle.org/current/userguide/single_versions.html

<https://android-arsenal.com/>



```
1 // build.gradle
2
3 ...
4
5 dependencies {
6
7     ...
8
9     implementation 'com.android.volley:volley:1.2.1'
10    implementation 'com.google.android.material:material:1.6.1'
11    // implementation 'com.google.android.material:material:+'
12    // implementation 'com.google.android.material:material:1.+'
13
14 }
```

Alternativ

- File
- Project
- Structure
- Module: App
- Dependencies

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to... [Sync Now](#) [Ignore these changes](#)

Android Studio

Exkurs vom Exkurs: SemVer

- Semantic Versioning
 - <https://semver.org/>

<https://semver.org/#semantic-versioning-200>

Semantic Versioning 2.0.0

Summary

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes
2. MINOR version when you add functionality in a backwards compatible manner
3. PATCH version when you make backwards compatible bug fixes

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

Android Studio

Artefakte

- Artefakte erzeugen
 - Build > Build Bundle(s) / APK(s) > Build APK(s) oder Build Bundle(s)
 - Build > Generate Signed Bundle / APK ...
 - signiert durch *keystore-file* <https://developer.android.com/training/articles/keystore>
- Artefakte
 - APK (Android-Package)
 - Android App Bundle https://developer.android.com/guide/app-bundle?utm_source=android-studio
 - für upload in Google Play Store
 - kleineres optimiertes APK

Android Studio

Während der Entwicklung

- Falls es mal Probleme gibt ...
 - Build > Clean Project
 - File > Invalidate Caches ... > Invalidate and Restart
- Im Terminal - Android Debug Bridge (adb)
 - >> adb devices
 - >> adb kill-server / adb start-server

Android Basics

- App Components
 - Activities
 - Services
 - Broadcast Receivers
 - Content Providers
- Intents (explizit/implizit)
- Manifest Datei

<https://developer.android.com/guide/components/fundamentals>

Android Basics

Activities

- `main()` - Methode einer Android Applikation
 - Startpunkt einer App ist immer die „`MainActivity`“ (ggf. anderer Name)
 - ermöglicht lose Kopplung
- einzelne Bausteine für Benutzer-Interaktion/ Benutzer-Flow, die atomar verwendet werden können (bspw. E-Mail Client)
- zeigt ein User-Interface, eine Activity per Screen (Preferences vs. Take-Picture)
- werden in der `Mainfest.xml` registriert
- Lifecycle-Events sind zu beachten

Android Basics

Activities



```
1 <manifest ... >
2 ...
3   <application ... >
4     ...
5       <activity android:name=".MainActivity" />
6     ...
7   </application>
8 ...
9 </manifest>
```

Android Basics

Activities - Intent Filter

```
1 <manifest ... >
2 ...
3   <application ... >
4     <activity android:name="com.example.project.ComposeEmailActivity">
5       <intent-filter>
6         <action android:name="android.intent.action.SEND" />
7         <data android:type="*/*" />
8         <category android:name="android.intent.category.DEFAULT" />
9       </intent-filter>
10      </activity>
11    </application>
12 </manifest>
```

Android Basics

Activities - Intent Filter / App - Start

```
1 <activity
2     android:name=".MainActivity"
3     android:exported="true"
4     android:label="@string/app_name"
5     android:theme="@style/Theme.HelloWorld.NoActionBar">
6     <intent-filter>
7         <action android:name="android.intent.action.MAIN" />
8         <category android:name="android.intent.category.LAUNCHER" />
9     </intent-filter>
10 </activity>
```

Android Basics

Activities - Activity starten

```
1 val sendIntent = Intent().apply {  
2     action = Intent.ACTION_SEND  
3     type = "text/plain"  
4     putExtra(Intent.EXTRA_TEXT, textMessage)  
5 }  
6  
7 startActivity(sendIntent)
```

Android Basics

Activities - Permissions

- Activity mit Hilfe einer Permission schützen

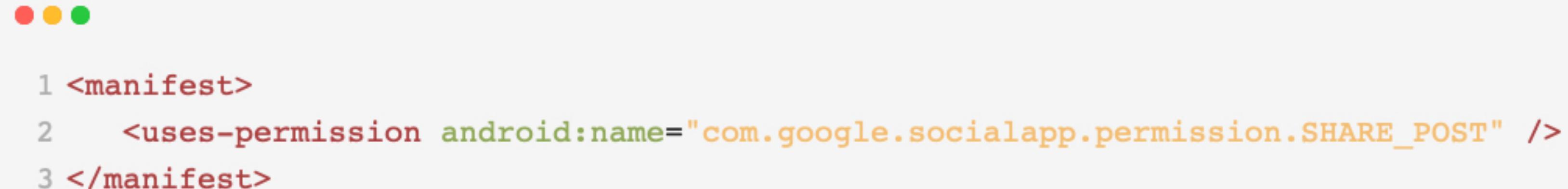
```
● ● ●  
1 <manifest>  
2  
3   <activity android:name="...."  
4     android:permission="com.google.socialapp.permission.SHARE_POST" />  
5  
6 ...
```

<https://developer.android.com/guide/components/activities/intro-activities>

Android Basics

Activities - Permissions

- andere App muss bestimmte Permission verwenden
 - (weitere Details bzgl. Permissions folgen ...)

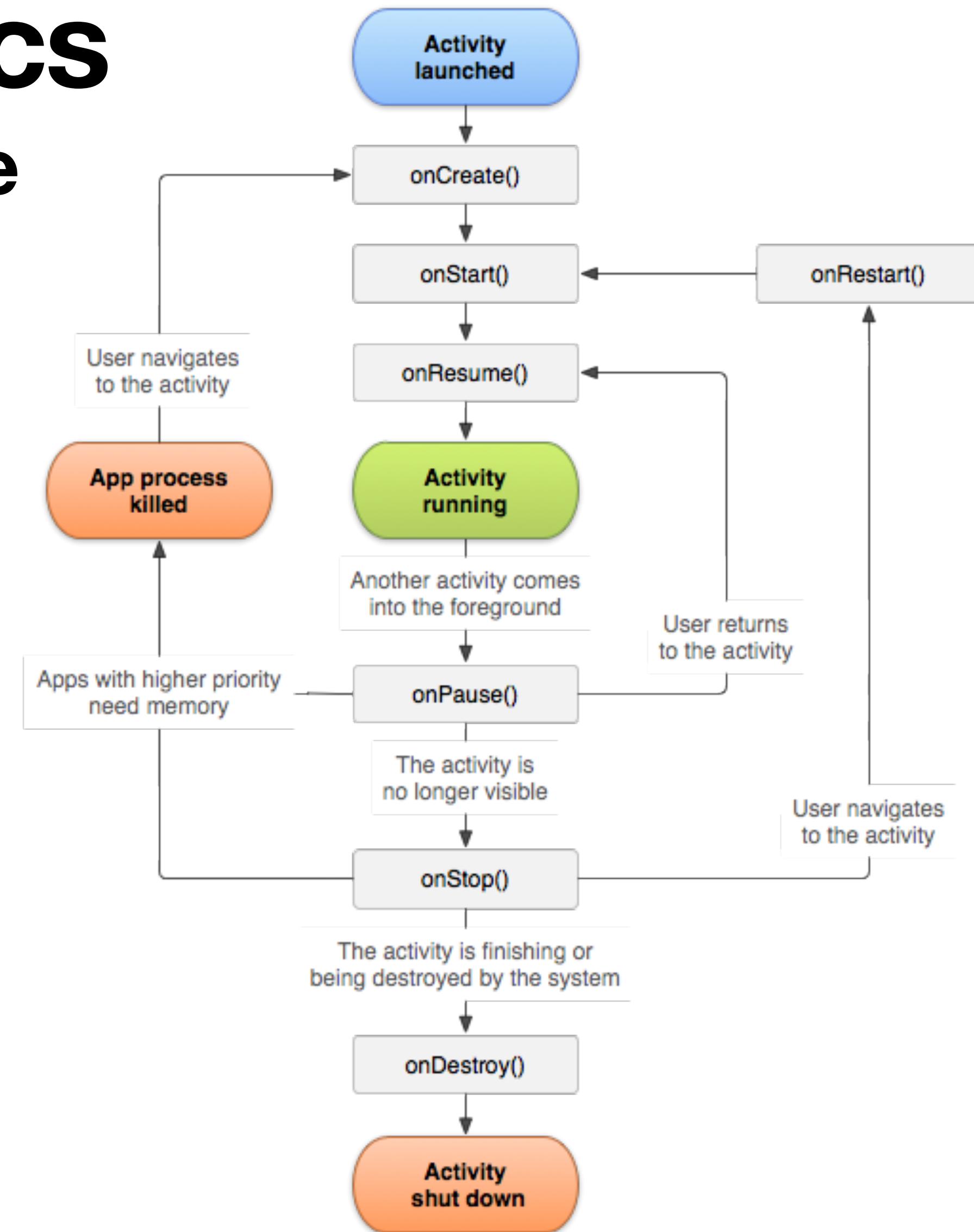


```
1 <manifest>
2   <uses-permission android:name="com.google.socialapp.permission.SHARE_POST" />
3 </manifest>
```

<https://developer.android.com/guide/components/activities/intro-activities>

Android Basics

Activities - Lifecycle



<https://developer.android.com/guide/components/activities/activity-lifecycle>

Android Basics

Activities - Layouts

- `onCreate()` - grundlegendes Setup für die Activity (u.a User-Interface)
- Layouts können im ...
 - ... XML-Format definiert werden
 - ... programmatisch erzeugt werden

Android Basics

Activities - Layouts

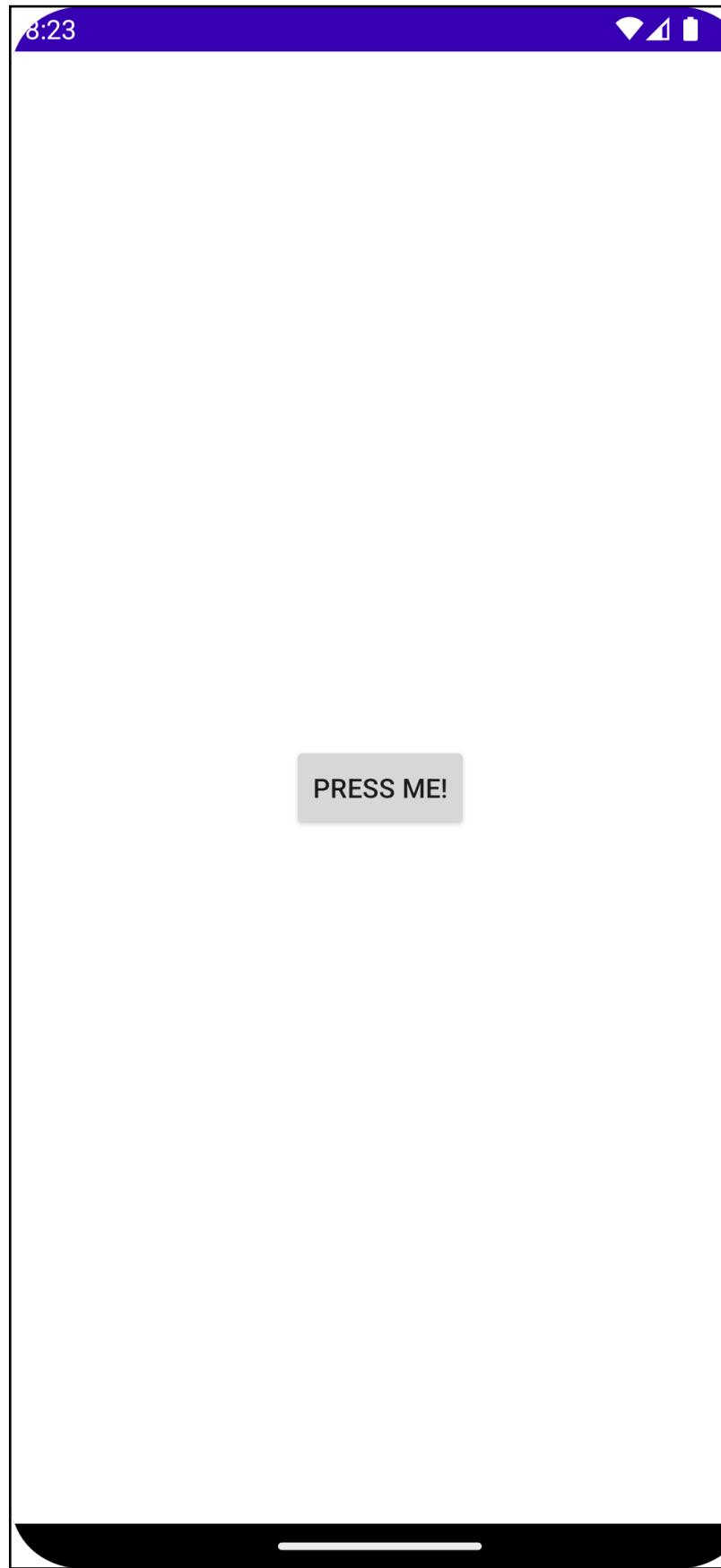


```
1 override fun onCreate(savedInstanceState: Bundle?) {  
2     super.onCreate(savedInstanceState)  
3     setContentView(R.layout.activity_main)  
4  
5 }
```

```
1 ...  
2  
3 <LinearLayout  
4     android:layout_width="match_parent"  
5     android:layout_height="match_parent"  
6     android:gravity="center">  
7  
8     <Button  
9         android:layout_width="wrap_content"  
10        android:layout_height="wrap_content"  
11        android:text="Press me!" />  
12    </LinearLayout>  
13  
14 ...
```

Android Basics

Activities - Layouts



```
● ● ●  
  
1 override fun onCreate(savedInstanceState: Bundle?) {  
2     super.onCreate(savedInstanceState)  
3  
4     val linearLayout = LinearLayout(this)  
5     linearLayout.layoutParams = ConstraintLayout.LayoutParams(  
6         ConstraintLayout.LayoutParams.MATCH_PARENT, // Width of TextView  
7         ConstraintLayout.LayoutParams.MATCH_PARENT)  
8  
9     val button = Button(this).apply {  
10        text = "Press me!"  
11    }  
12  
13    linearLayout.gravity = Gravity.CENTER  
14  
15    linearLayout.addView(button)  
16    setContentView(linearLayout)  
17  
18    ...  
19}
```

Android Basics

Activities - Layouts

- Zugriff auf Elemente per Referenz-ID
(ein-eindeutig pro Layout)

```

●●●
1 ...
2
3 private var counter: Int = 0
4
5 override fun onCreate(savedInstanceState: Bundle?) {
6     super.onCreate(savedInstanceState)
7     setContentView(R.layout.activity_main)
8
9     val textViewCounter = findViewById<TextView>(R.id.text_view_counter)
10    findViewById<Button>(R.id.button_press_me).setOnClickListener {
11        textViewCounter.text = counter++.toString()
12    }
13
14 ...

```

```

●●●
1 ...
2
3 <LinearLayout
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical"
7     android:gravity="center">
8
9     <TextView
10        android:id="@+id/text_view_counter"
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13        android:textSize="28dp"
14        android:text="0"/>
15
16     <Button
17        android:id="@+id/button_press_me"
18        android:layout_width="wrap_content"
19        android:layout_height="wrap_content"
20        android:text="Press me! " />
21 </LinearLayout>
22
23 ...

```