

Introduction to App Development for Android

Android Studio

Prof. Dr. Paul Rawiel

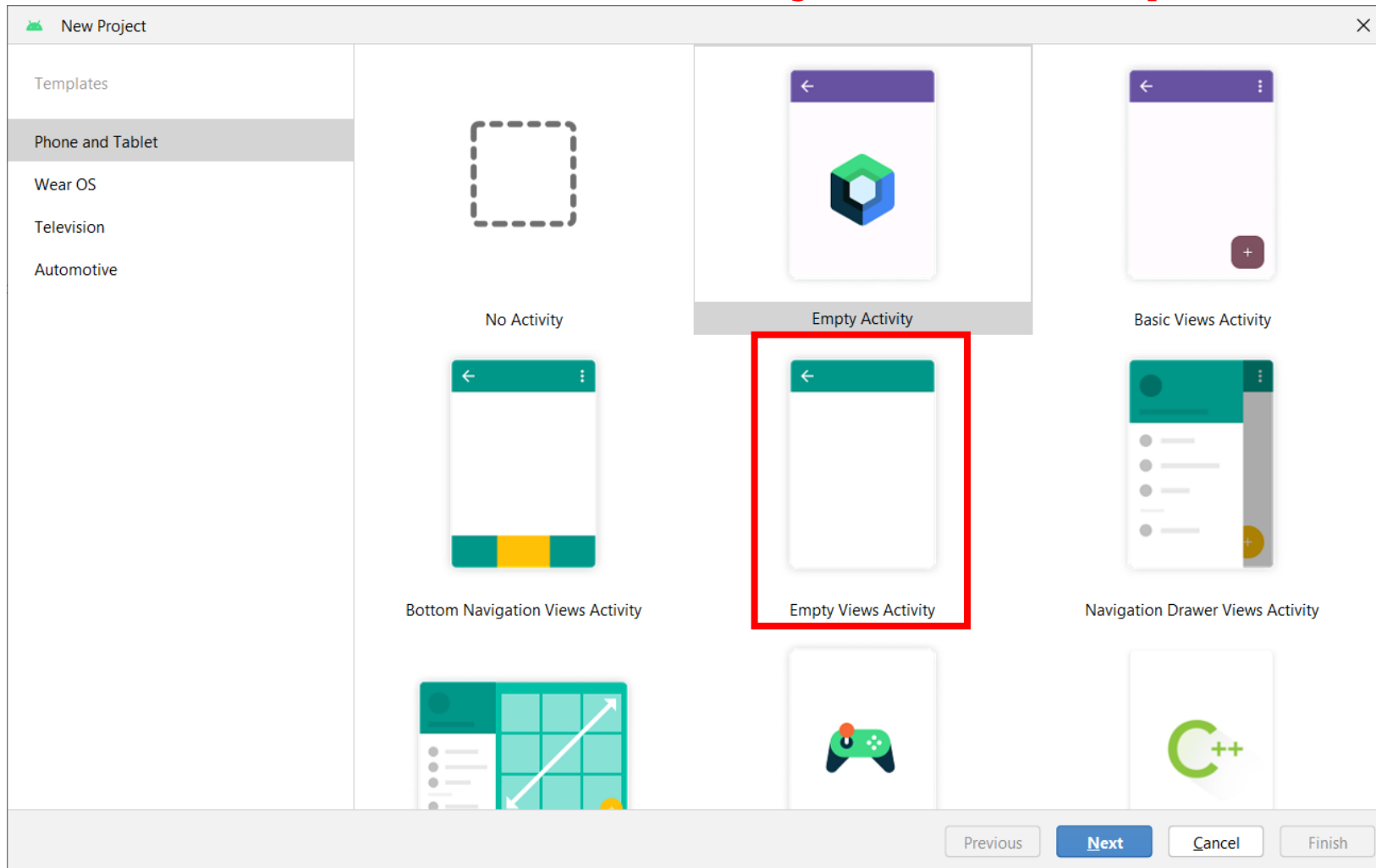
Android Studio

- Standard development environment Android Apps
- Eclipse also possible if the needed plugins are installed
- Android Studio is more comfortable, because everything is already preconfigured for App development

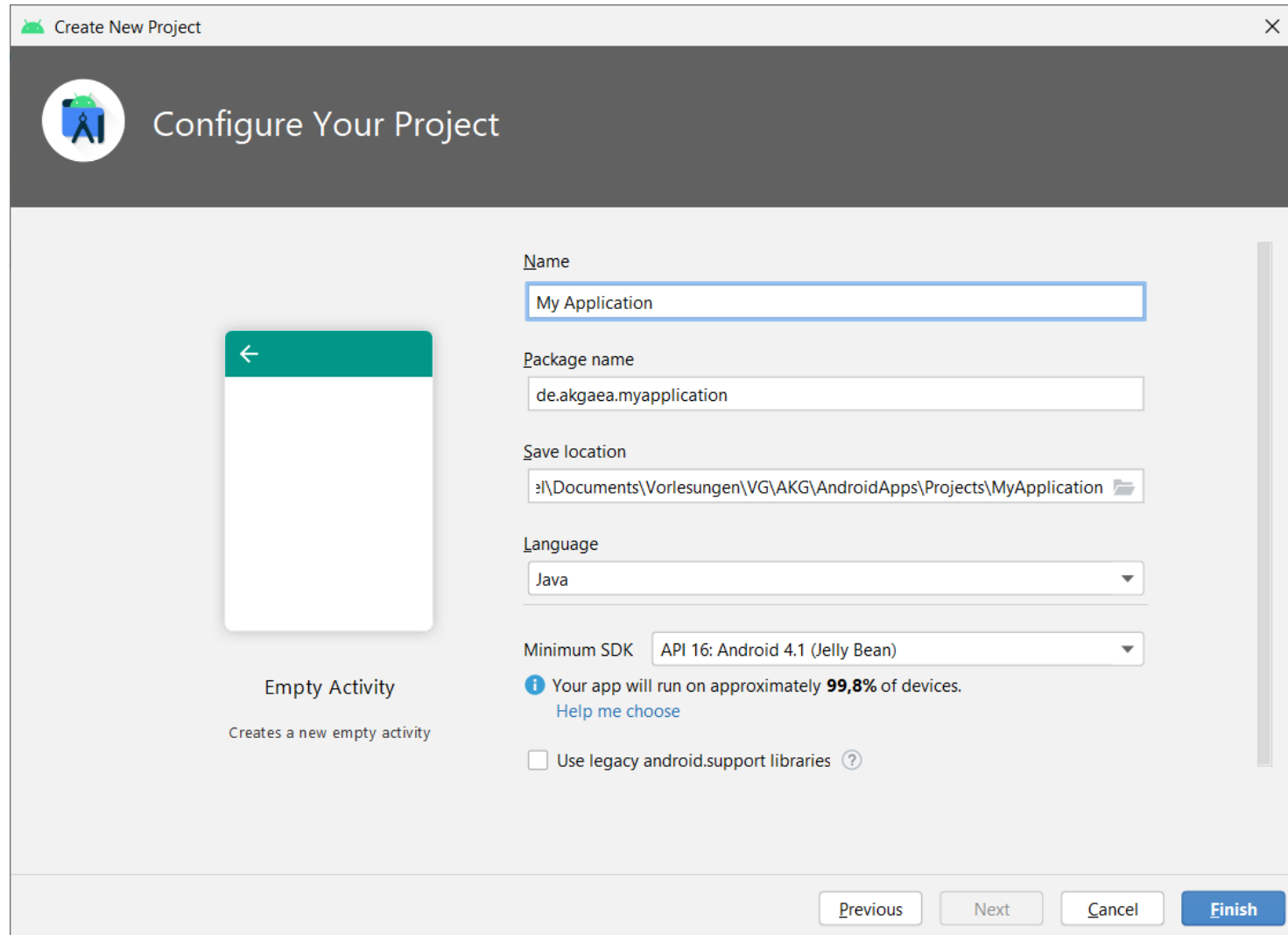
First start of Android Studio

- A screen appears, where you can make a new project
- Selection of a project template
- Several infos about the project
- After clicking the finish button a new directory is created and many files in a fixed directory structure are created
 - Don't change this structure!
 - If it has to be changed for some reason, only apply changes using the Android Studio
 - Never with the Windows Explorer

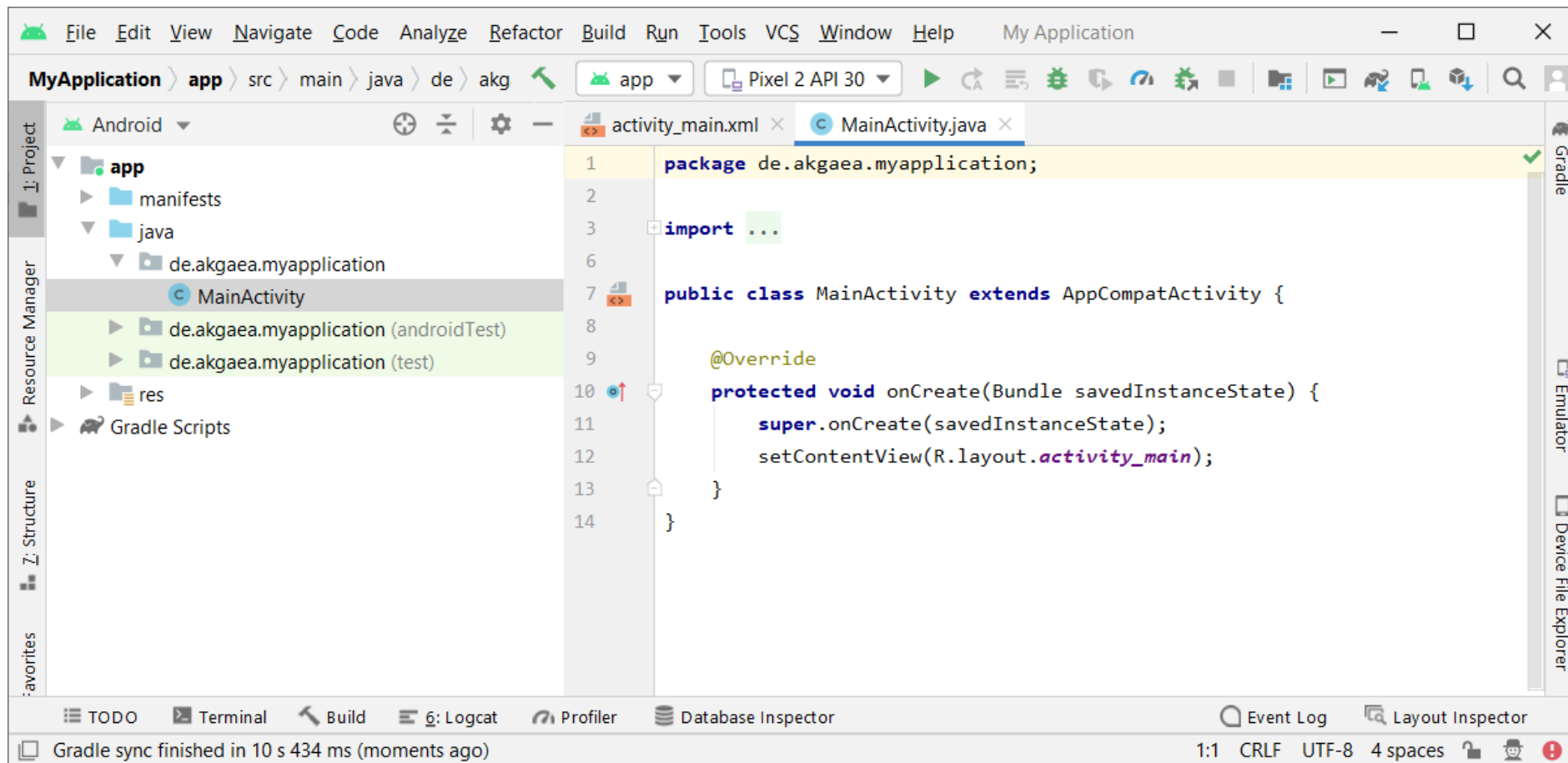
Android Studio Project templates



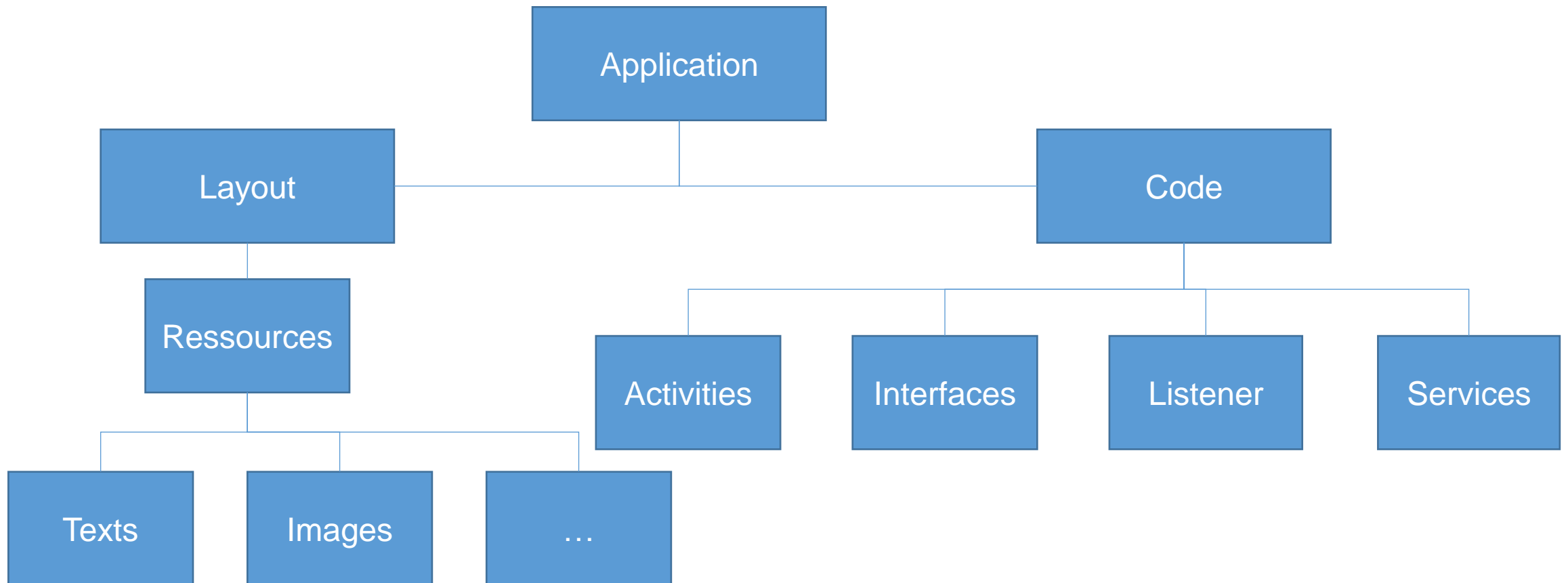
Android Studio – several Infos



Android Studio



Components of an App



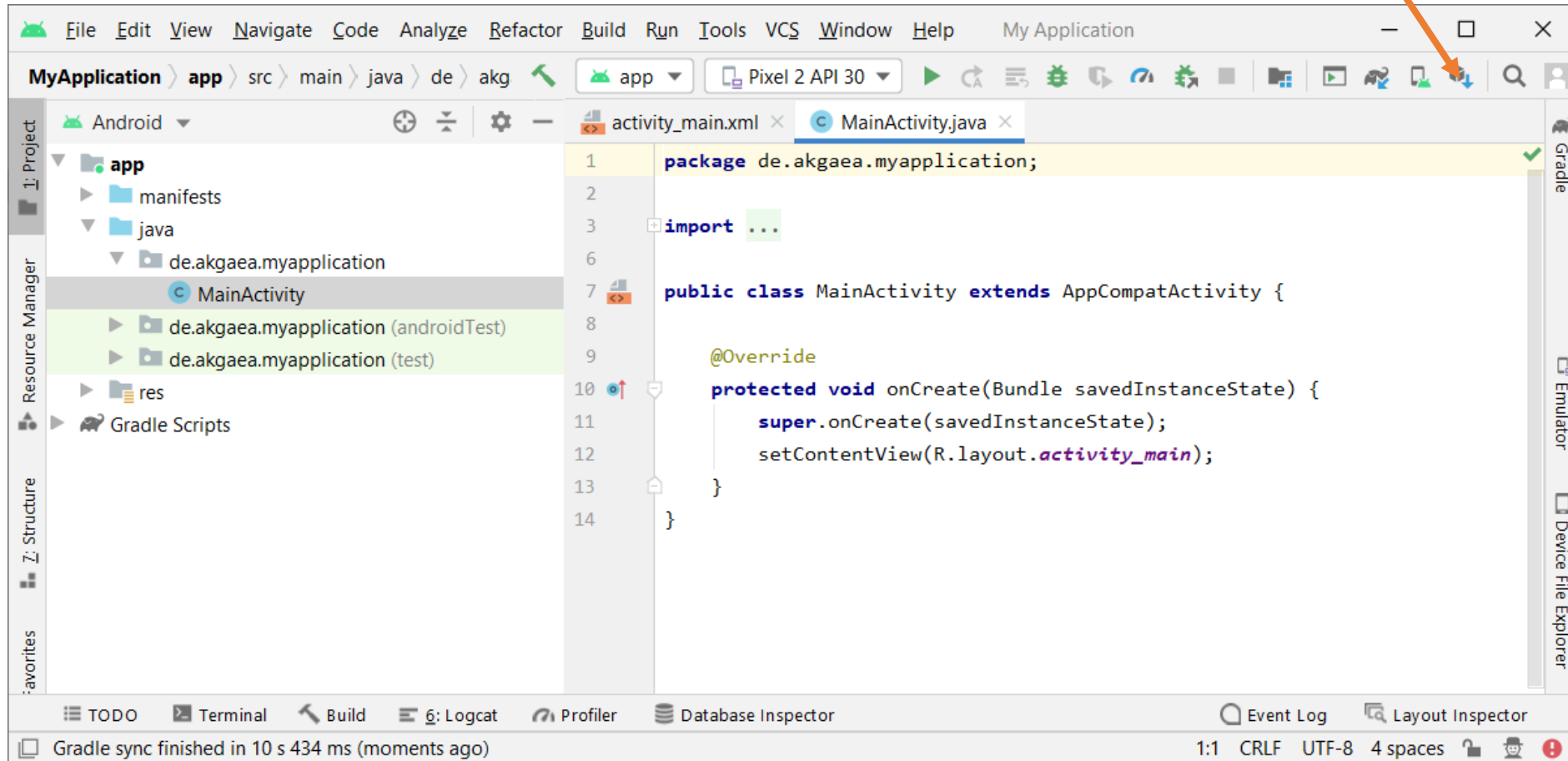
First running App



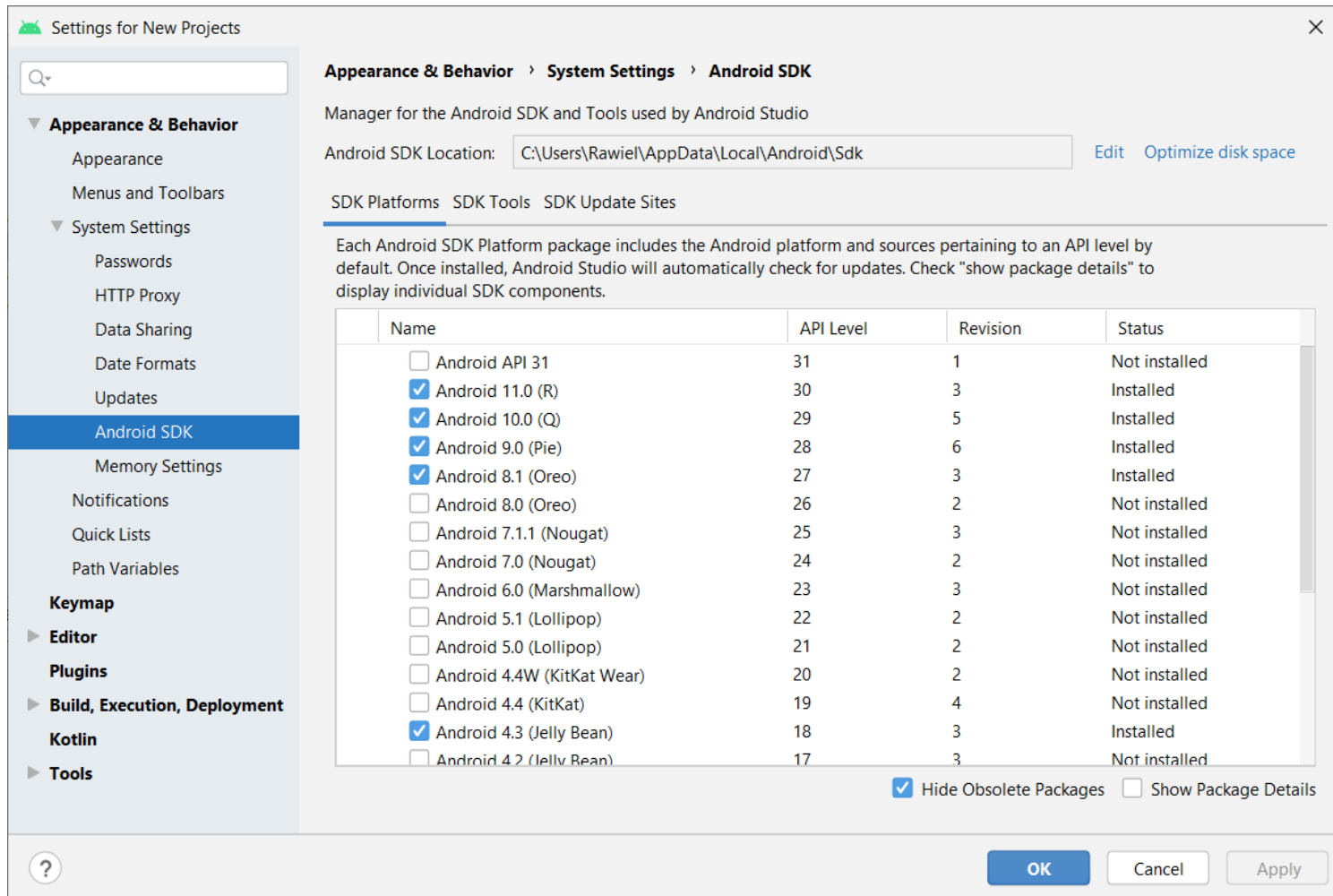
- This is how the created App
- Looks like
- It doesn't do anything
- Just shows a text:
 - Hello World
- Where does that come from?
- How can we make the App do something – how can we communicate with the App?

Primeiro ...

SDK Manager



Der SDK Manager



The screenshot shows the 'Settings for New Projects' dialog box in Android Studio, specifically the 'Android SDK' section under 'System Settings'. The 'Android SDK Location' is set to 'C:\Users\Rawiel\AppData\Local\Android\Sdk'. Below this, there are tabs for 'SDK Platforms', 'SDK Tools', and 'SDK Update Sites'. The 'SDK Platforms' tab is active, displaying a table of installed and available SDK versions. The table has columns for Name, API Level, Revision, and Status. The 'Hide Obsolete Packages' checkbox is checked, and the 'Show Package Details' checkbox is unchecked. At the bottom, there are 'OK', 'Cancel', and 'Apply' buttons.

Settings for New Projects

Appearance & Behavior > System Settings > Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\Rawiel\AppData\Local\Android\Sdk [Edit](#) [Optimize disk space](#)

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

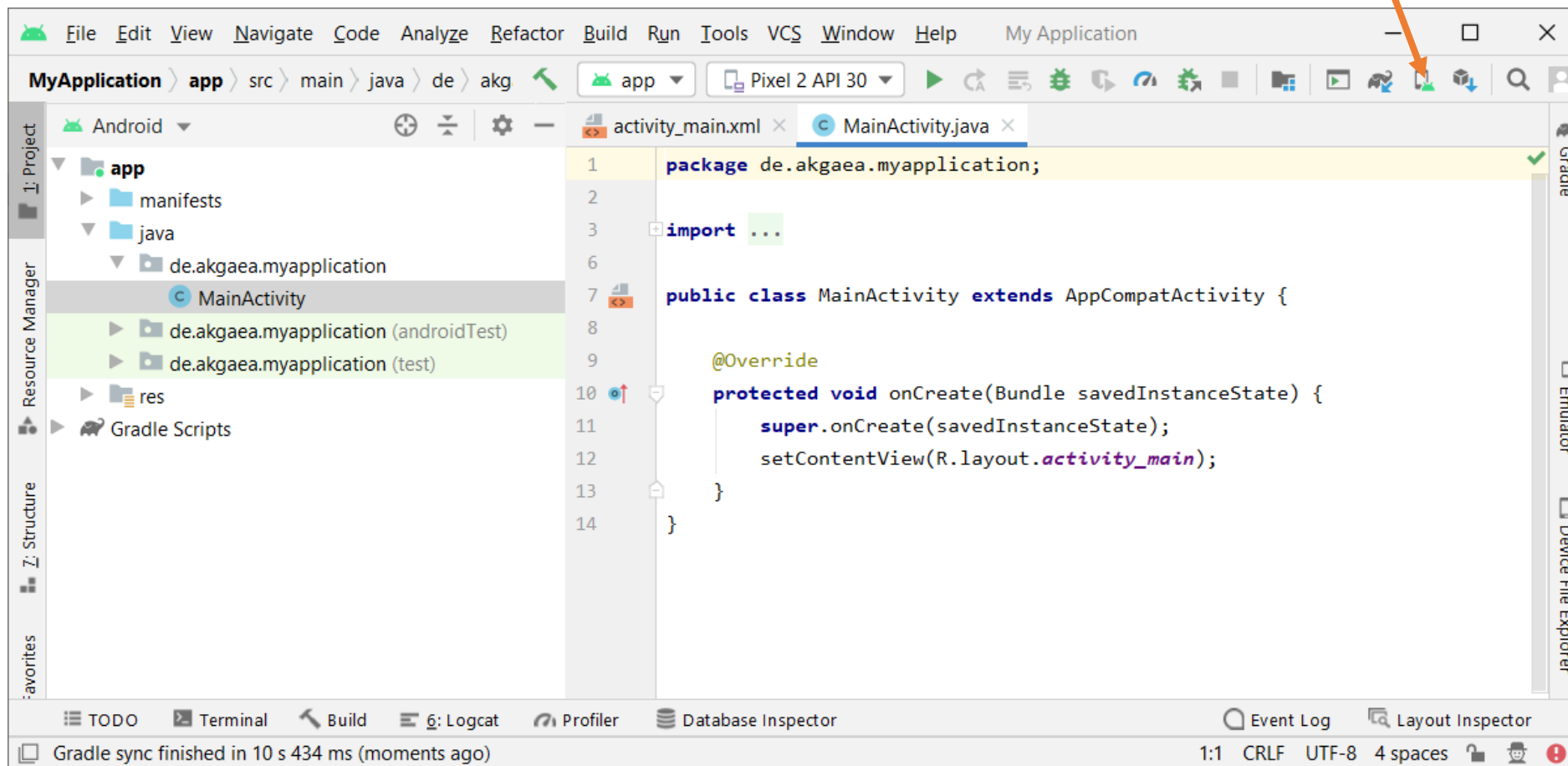
Name	API Level	Revision	Status
<input type="checkbox"/> Android API 31	31	1	Not installed
<input checked="" type="checkbox"/> Android 11.0 (R)	30	3	Installed
<input checked="" type="checkbox"/> Android 10.0 (Q)	29	5	Installed
<input checked="" type="checkbox"/> Android 9.0 (Pie)	28	6	Installed
<input checked="" type="checkbox"/> Android 8.1 (Oreo)	27	3	Installed
<input type="checkbox"/> Android 8.0 (Oreo)	26	2	Not installed
<input type="checkbox"/> Android 7.1.1 (Nougat)	25	3	Not installed
<input type="checkbox"/> Android 7.0 (Nougat)	24	2	Not installed
<input type="checkbox"/> Android 6.0 (Marshmallow)	23	3	Not installed
<input type="checkbox"/> Android 5.1 (Lollipop)	22	2	Not installed
<input type="checkbox"/> Android 5.0 (Lollipop)	21	2	Not installed
<input type="checkbox"/> Android 4.4W (KitKat Wear)	20	2	Not installed
<input type="checkbox"/> Android 4.4 (KitKat)	19	4	Not installed
<input checked="" type="checkbox"/> Android 4.3 (Jelly Bean)	18	3	Installed
<input type="checkbox"/> Android 4.2 (Jelly Bean)	17	3	Not installed

Hide Obsolete Packages Show Package Details

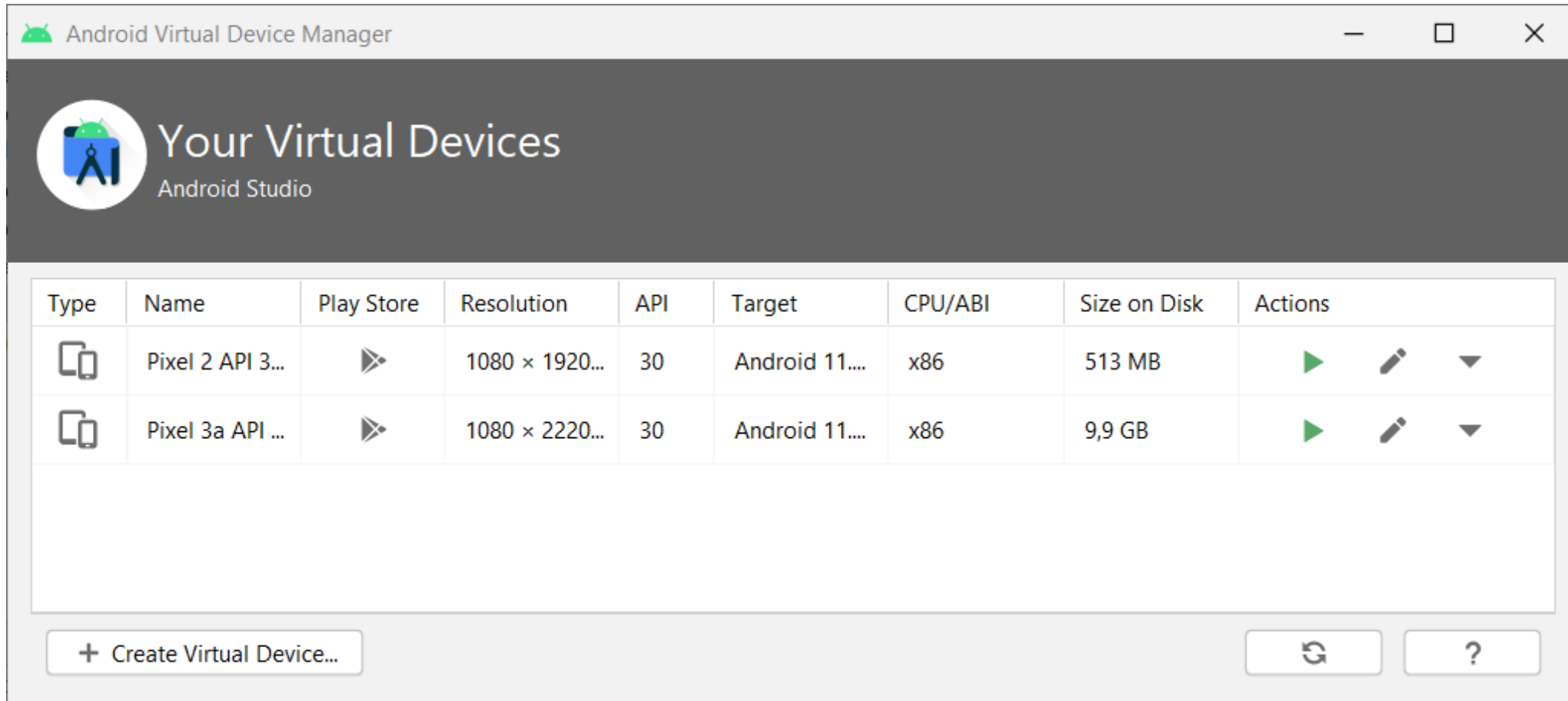
OK Cancel Apply

And then...

AVD Manager



Der AVD Manager



The AVD Manager

- Create a new Android Virtual Device (AVD)
- Select a hardware profile
- Download of an Android version (System Image) for this hardware profile
 - Takes a lot of disk space, don't install too many
 - One is enough
 - For this workshop we don't need that if everybody has an android smartphone

Der AVD Manager

Virtual Device Configuration


System Image

Select a system image

Recommended x86 Images Other Images

Release Name	API Level	ABI	Target
R	30	x86	Android 11.0 (Google)
Q Download	29	x86	Android 10.0 (Google)
Pie Download	28	x86	Android 9.0 (Google)
Oreo Download	27	x86	Android 8.1 (Google)
Oreo Download	26	x86	Android 8.0 (Google)
Nougat Download	25	x86	Android 7.1.1 (Google)

R



API Level
30

Android
11.0

Google Inc.

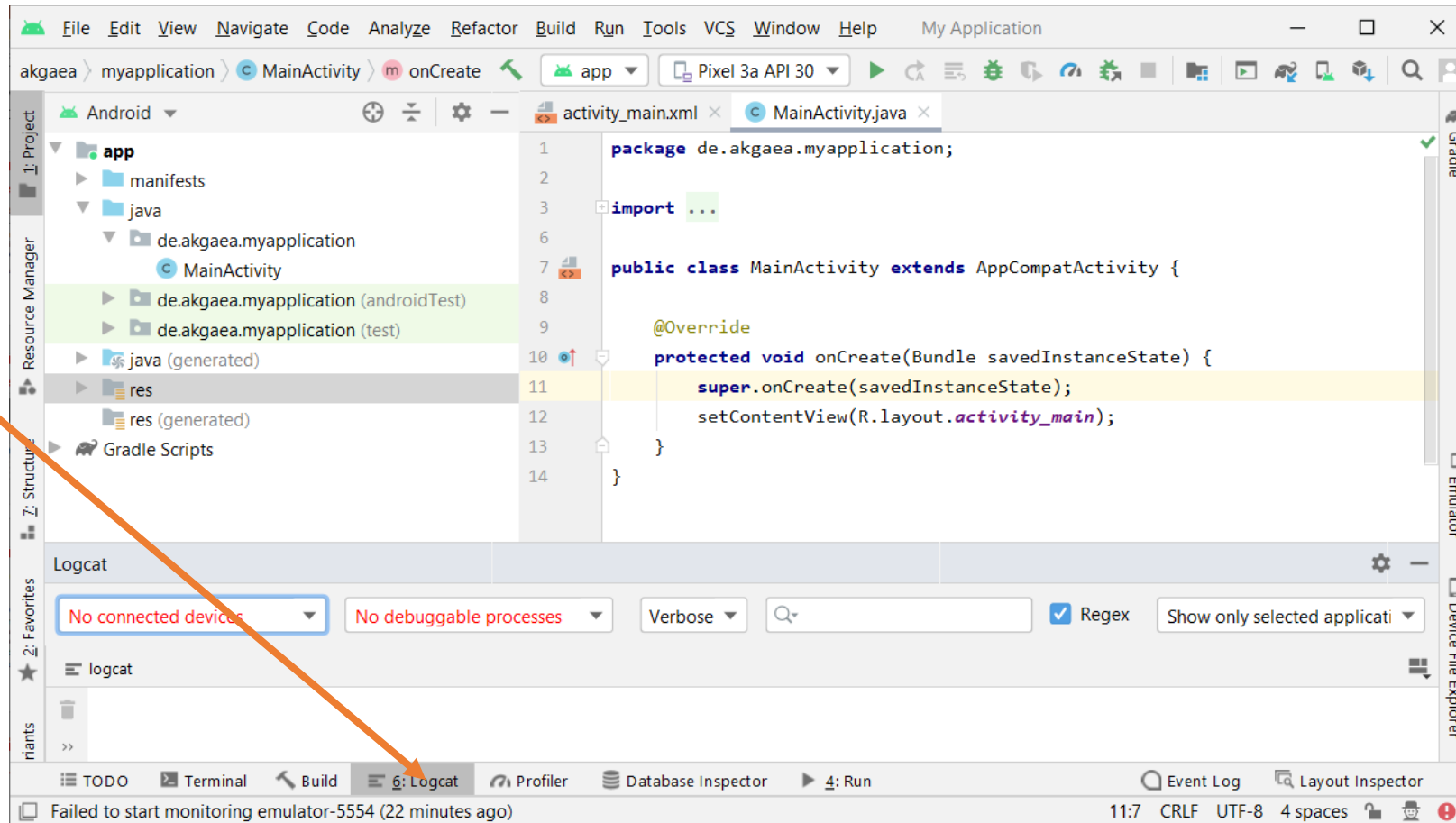
? Previous Next Cancel Finish

Testing of an App

- On the Virtual Device
 - On the real smartphone
 - The necessary settings in the developer options have to be set
 - Connect the smartphone via USB cable to the computer
- it will be detected by the Android Studio and can be chosen as the destiny for the App

Help to find errors

Logcat



The Logcat window

- Here everything that happens on the phone is shown, after you started an App out of Android Studio
- Many entries with different Log Level
 - E = Error
 - W = Warning
 - I = Info
 - D = Debug
 - V = Verbose (extensive)

The Logcat Window

- Create own messages
- The `Log` class
 - Static methods to add lines to the protocol shown in the logcat window
 - Example: `Log.e („My Tag“, „error description“);`
→ Produces a message of the error type
 - `Log.d` produces a message of the debug type
- The first string serves to better find your message
- The second string can be any text, that will be shown as message

Debugging

- Android Studio comes with a powerful debugger
- For debugging you have to set a breakpoint where the execution of the code will be stopped
- Start the App in Debug Mode
 - Icon the icon with the green bug
- The App can then be executed command by command

First running App



- This is how the created App
- Looks like
- It doesn't do anything
- Just shows a text:
 - Hello World
- Where does that come from?
- How can we make the App do something – how can we communicate with the App?

What are the options?

- the Smartphone can
 - listen (Microfon)
 - talk (Lautsprecher)
 - see (Kamera)
 - Determine where it is
 - Measure several things
- Communication through
 - Input of text
 - Buttons
 - Touchscreen
- ...

Some important programming concepts

- classes and objects
 - constructors
 - Import classes
- Data administration
 - Lists
 - Loops
- inheritance
- Interfaces
- Listener