

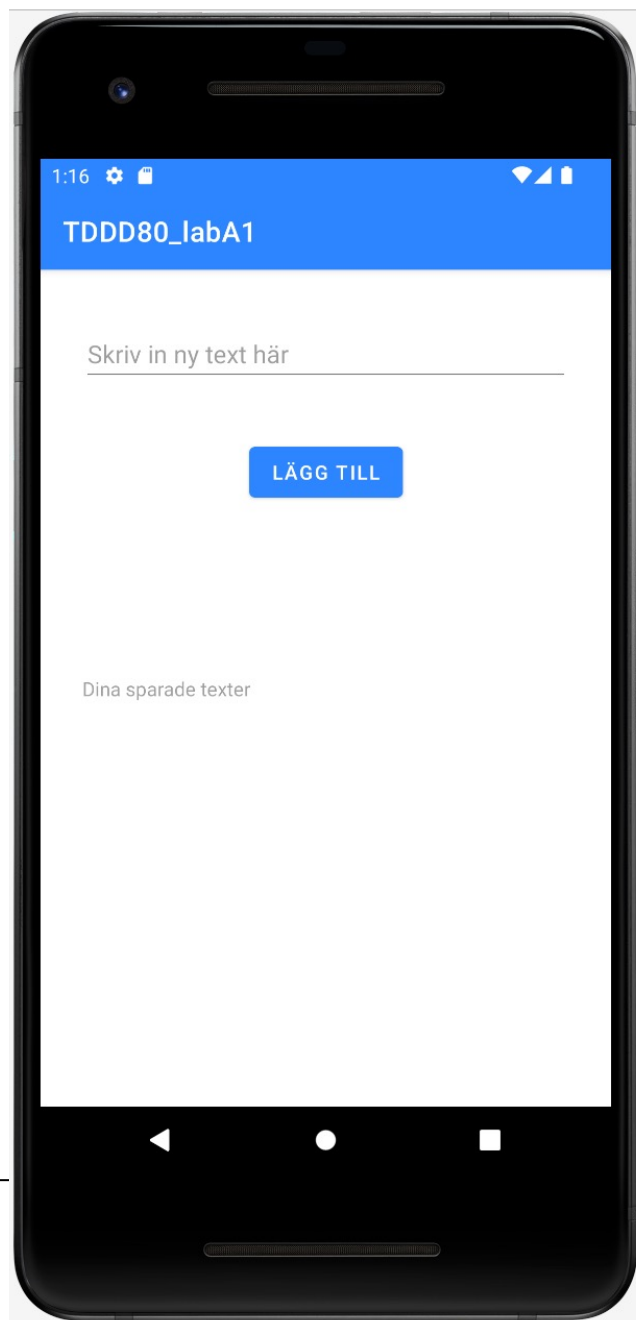
Android översikt

TDDD80 Mobila och sociala applikationer

Översikt

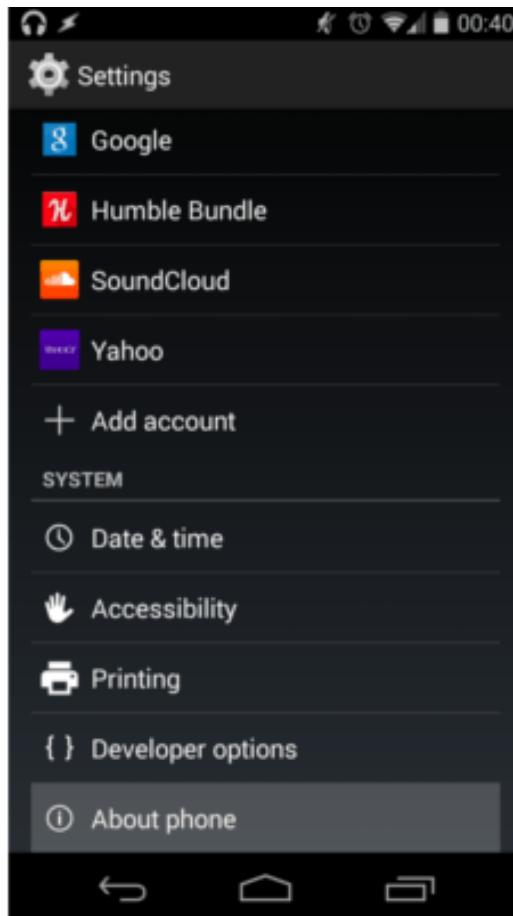
- Köra app på mobil / emulator
- Android Studio introduktion
- Android kodning

Android labb 1



Köra på mobil / emulator

Egen mobil eller platta

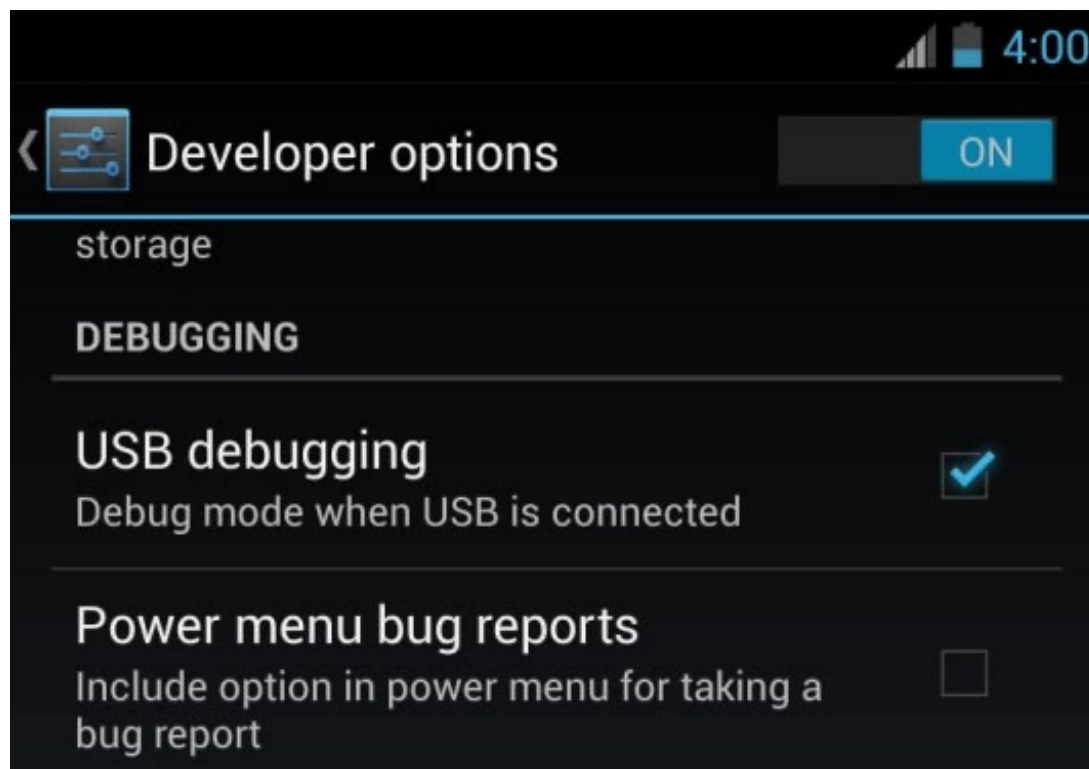


Developer mode på mobilen

- <https://www.youtube.com/watch?v=sqCGvQRZ8wY>

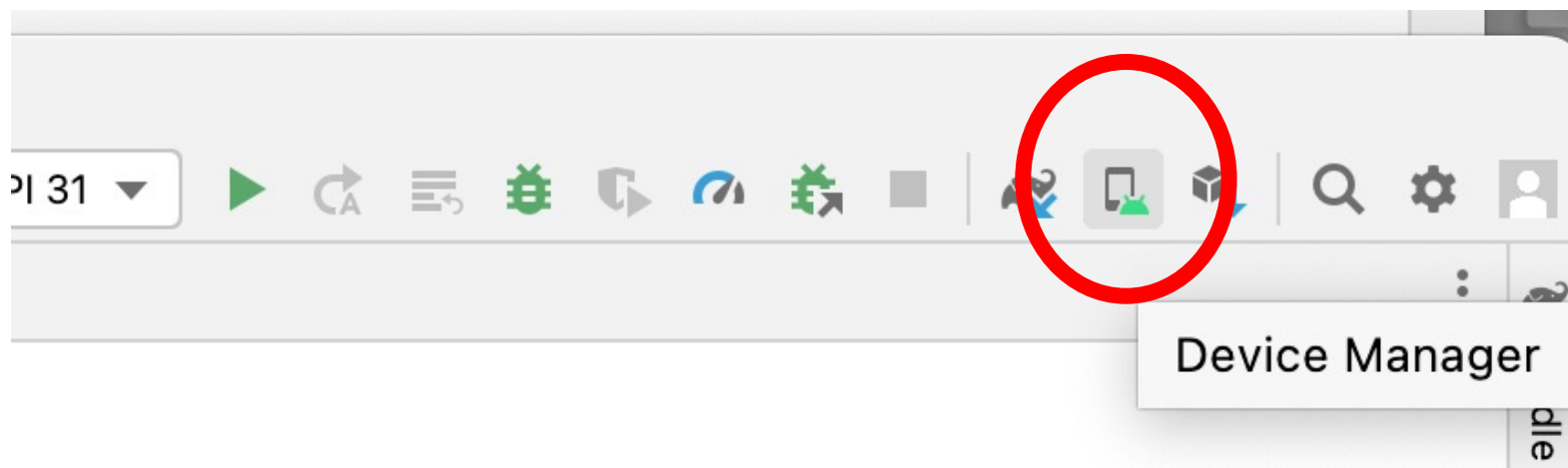


Felsökning via USB-koppling

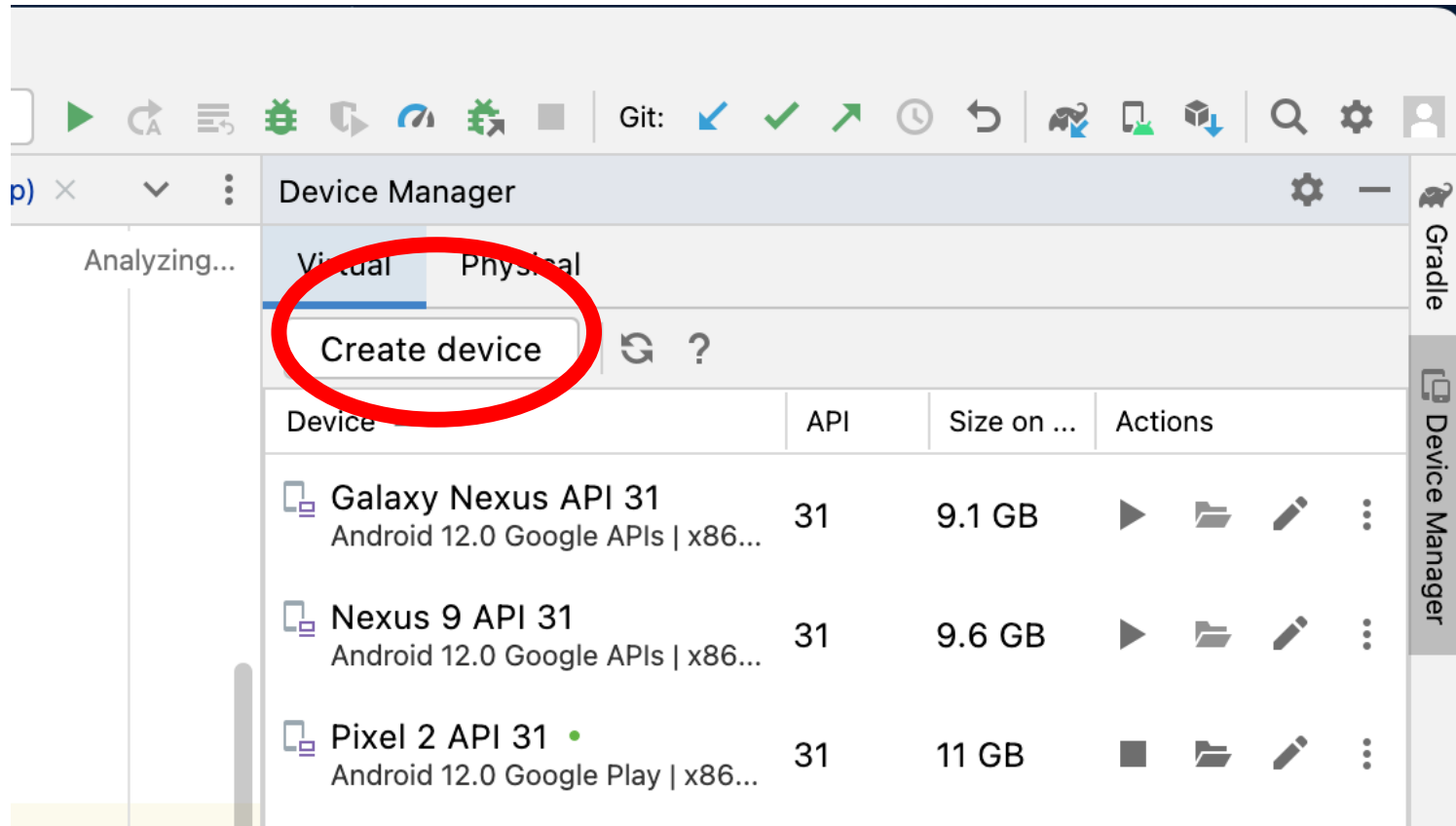


Emulator i Android Studio

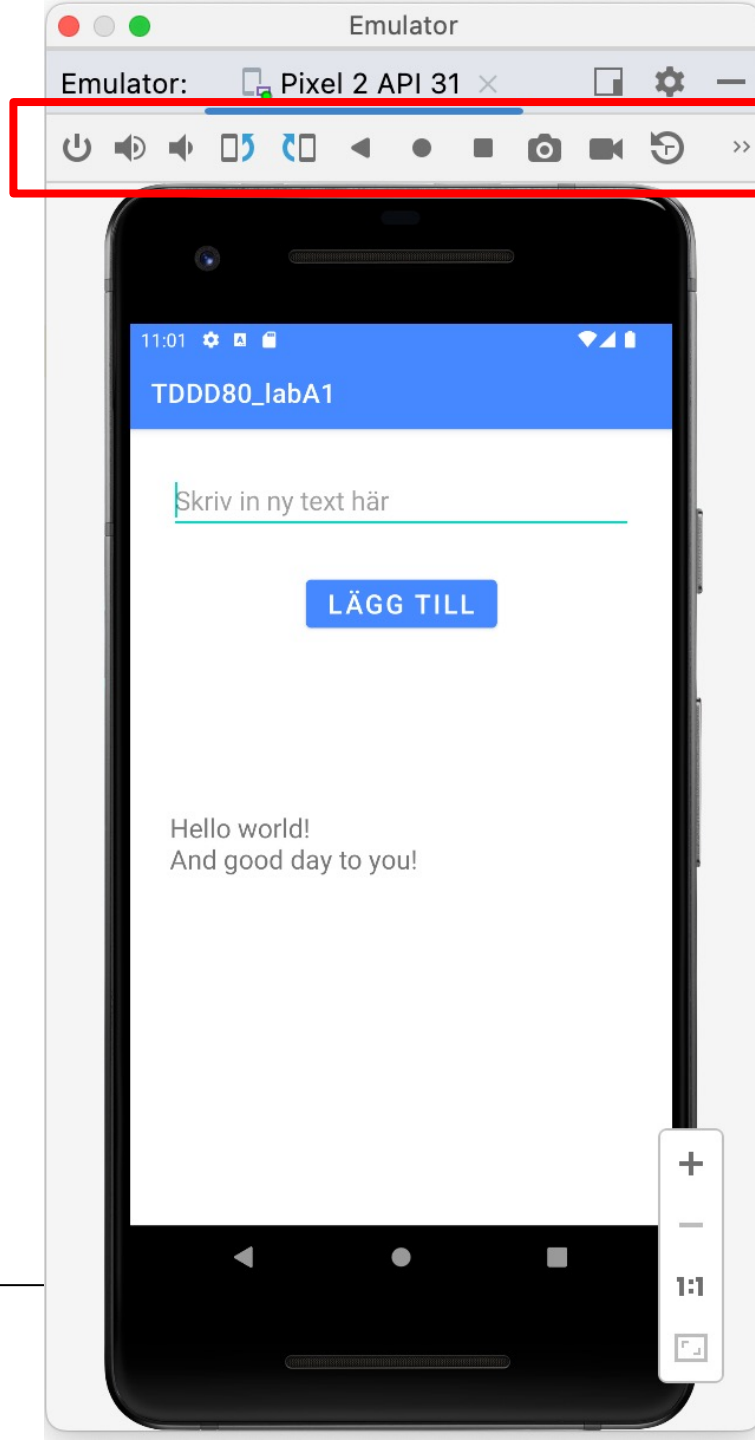
- Skapa emulator (Android Virtual Device, AVD)



Emulator i Android Studio



Emulatorn

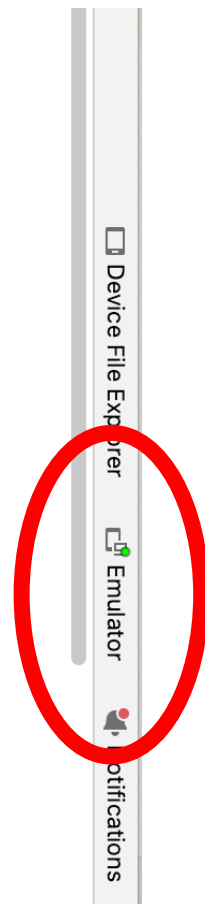


kontroller

Emulatorn kan vara långsam

- Tung att hålla igång
- Emulatorn ”försvinner” när den inte behövs
 - Fönstret stängs
 - Öppna igen genom att klicka på kanten på editorn

```
osition);  
/_container).navigate(action);  
  
:ion);  
:ontainer).navigate(action);
```



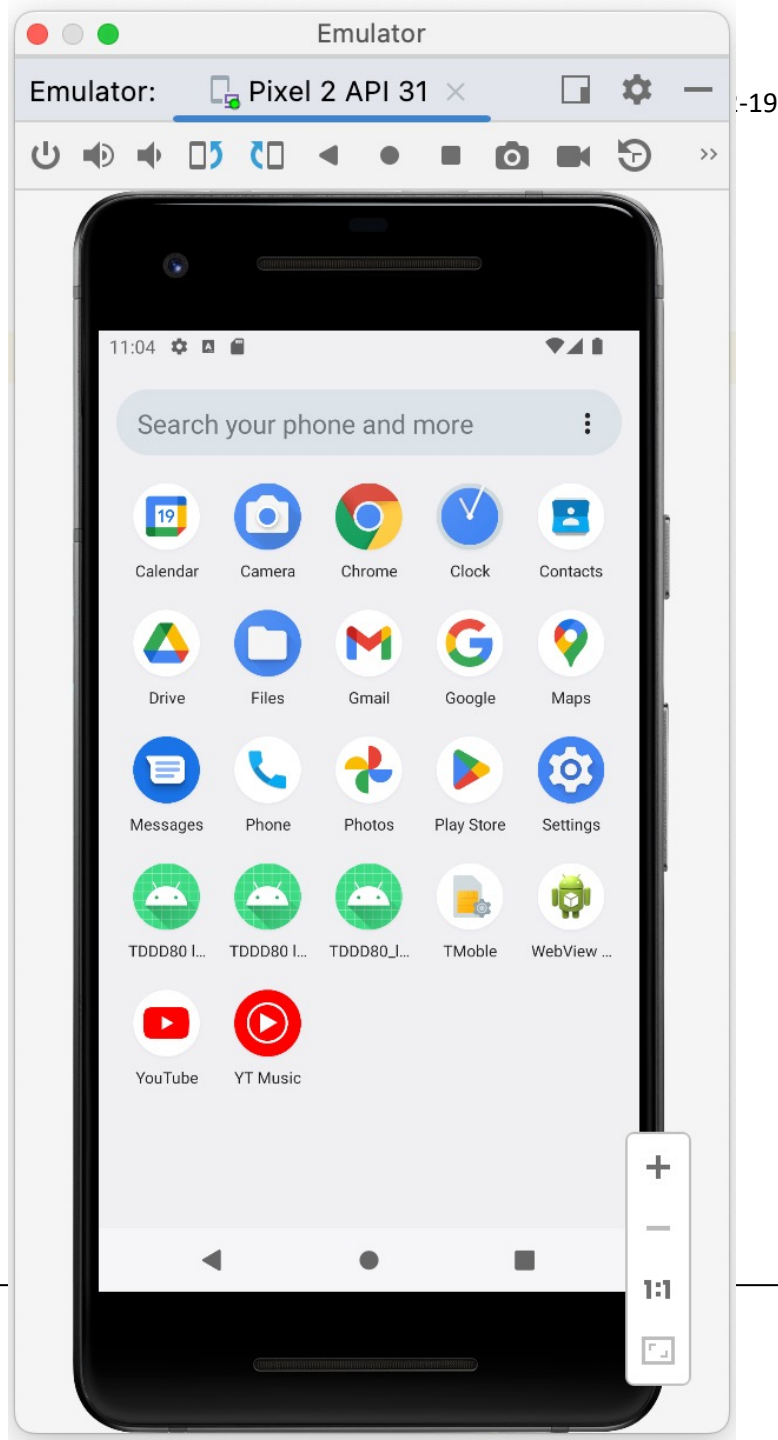
Layout Inspector

32:9 LF UTF-8 4 spaces main

Internet

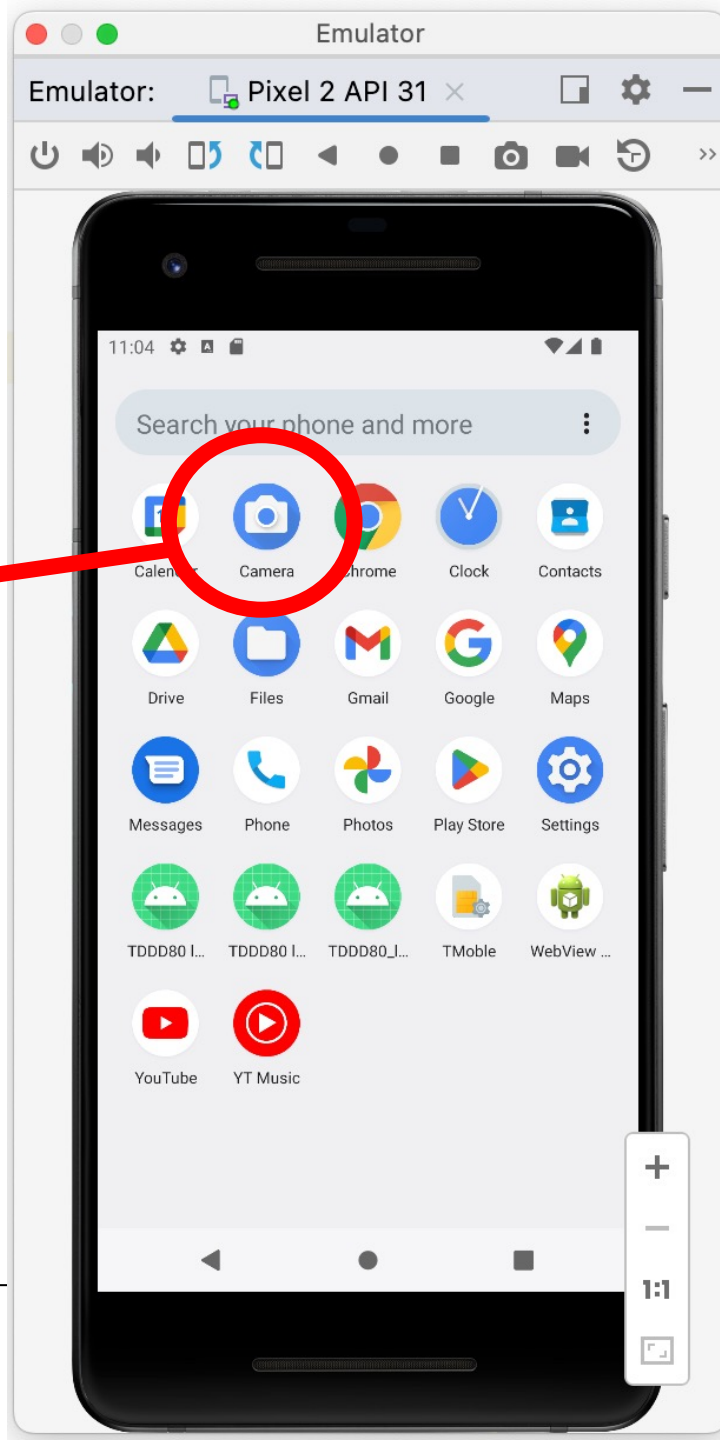
- Emulatorn delar datorns internetuppkoppling
- Kolla genom att t.ex. öppna emulatorns webbläsare, och se om kontakt med internet
 - Annars: Inställningar på datorn för att dela internet

Appar i emulatorn

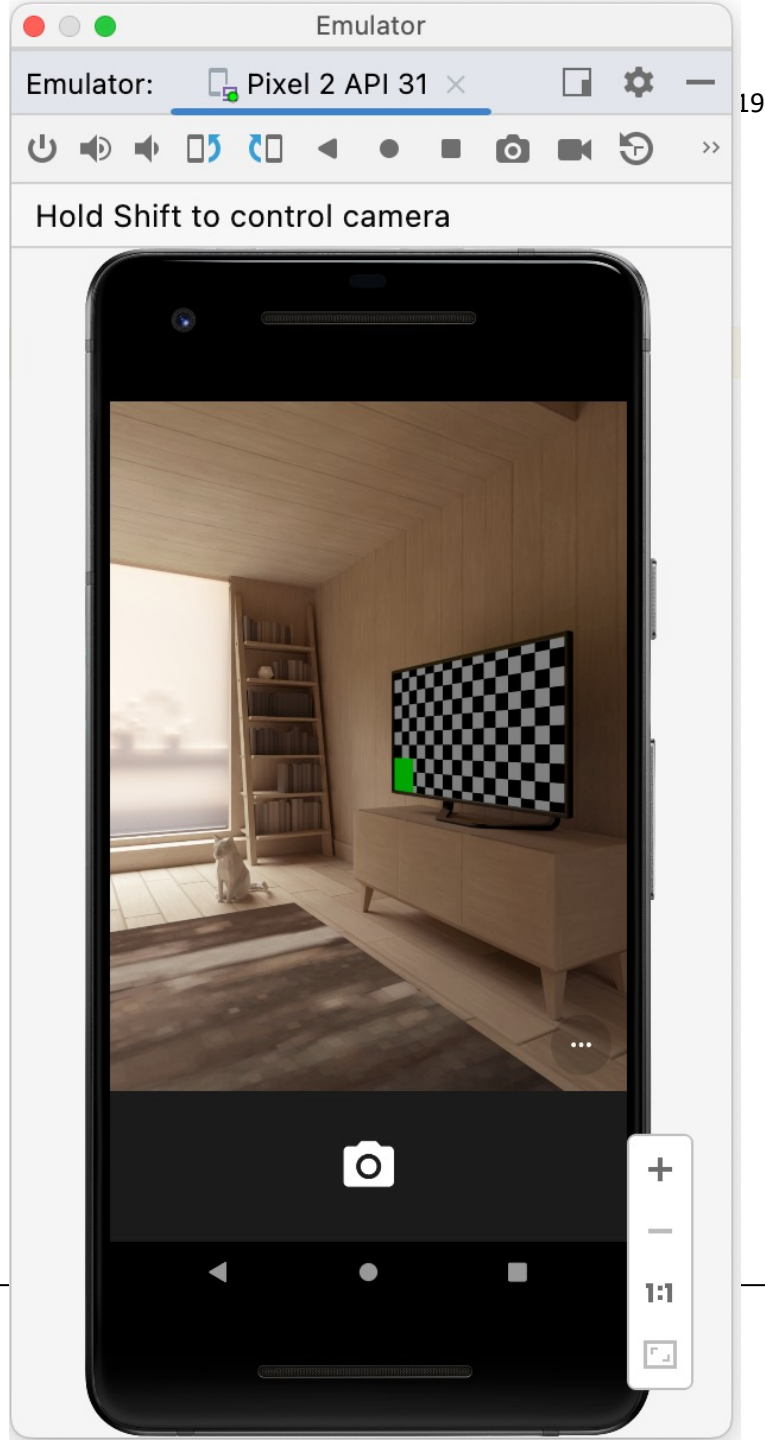


Virtuell kamera

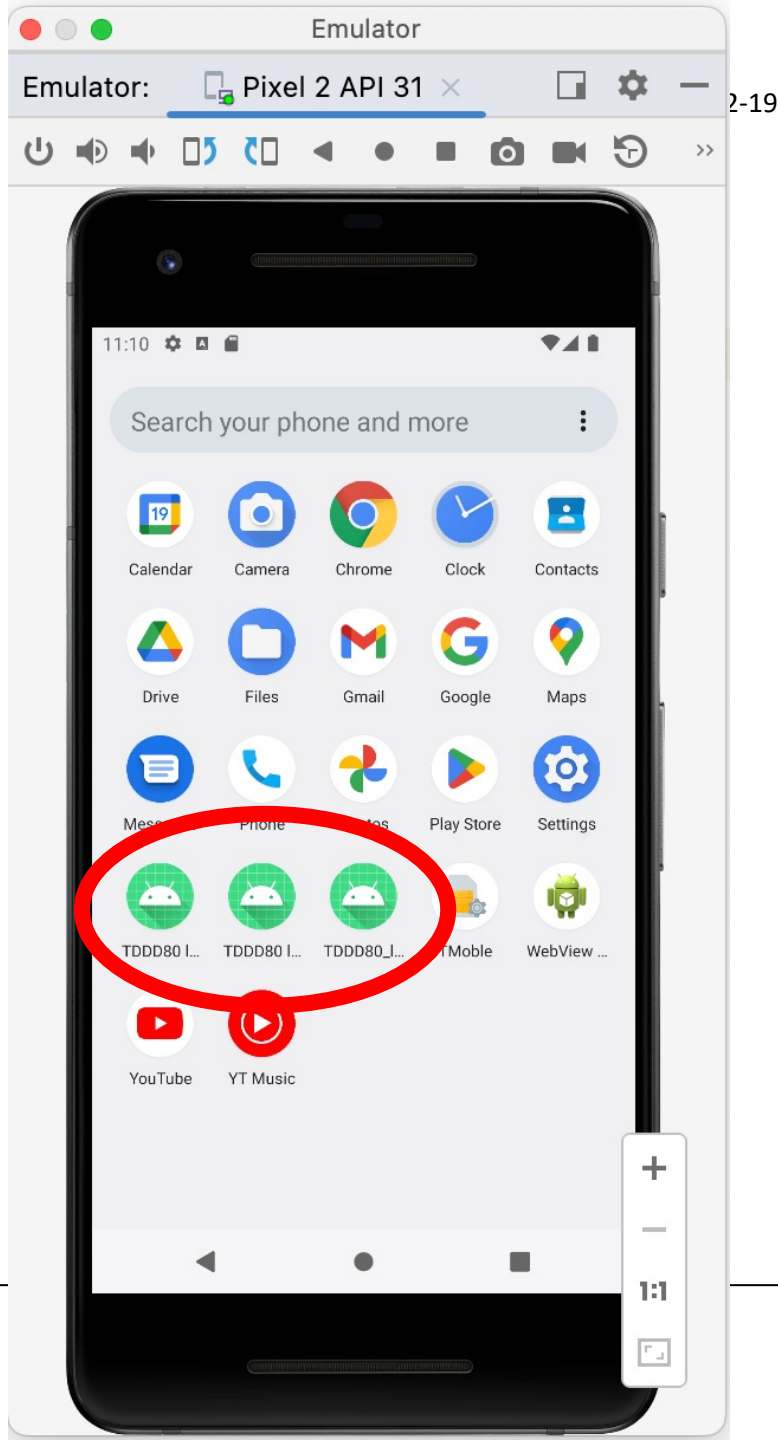
Nås genom att klicka på appen Kamera



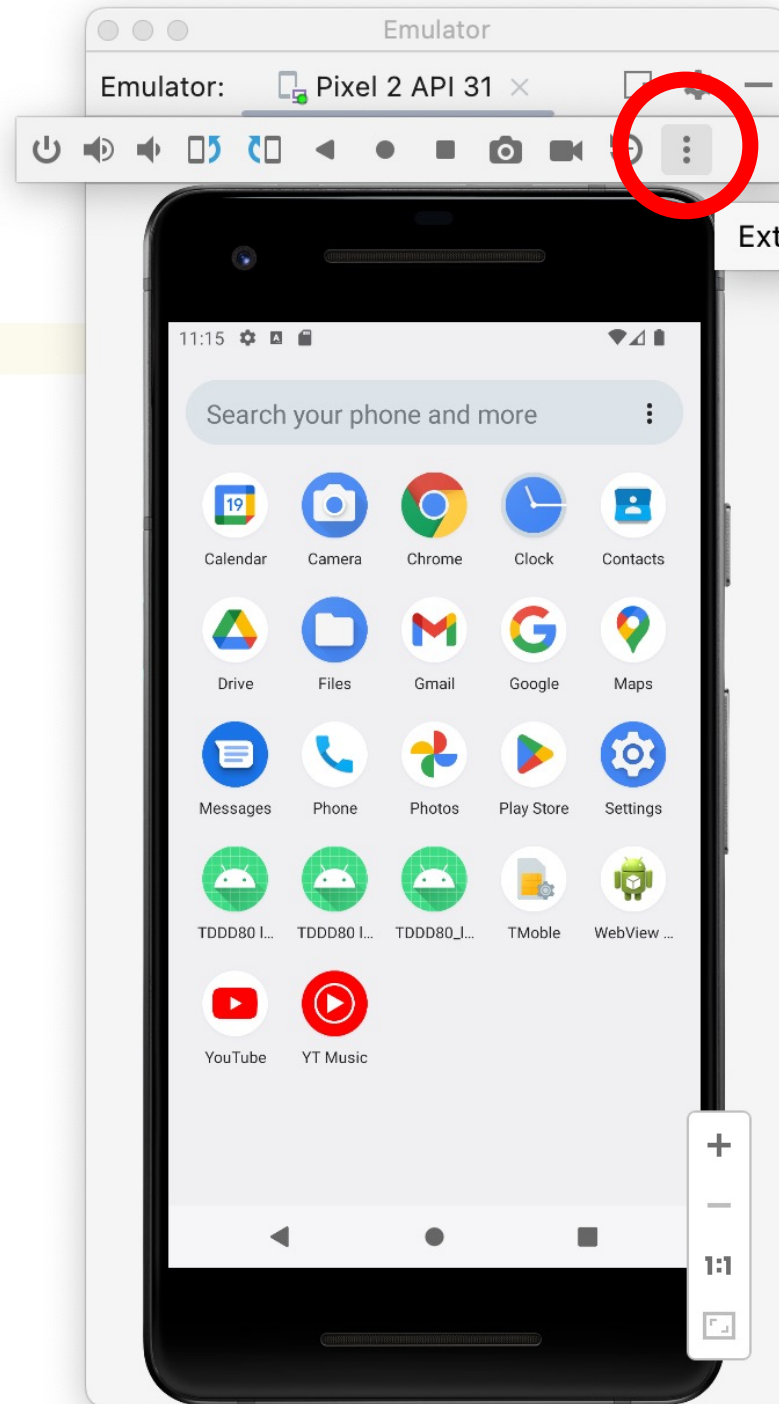
Virtuell kamera

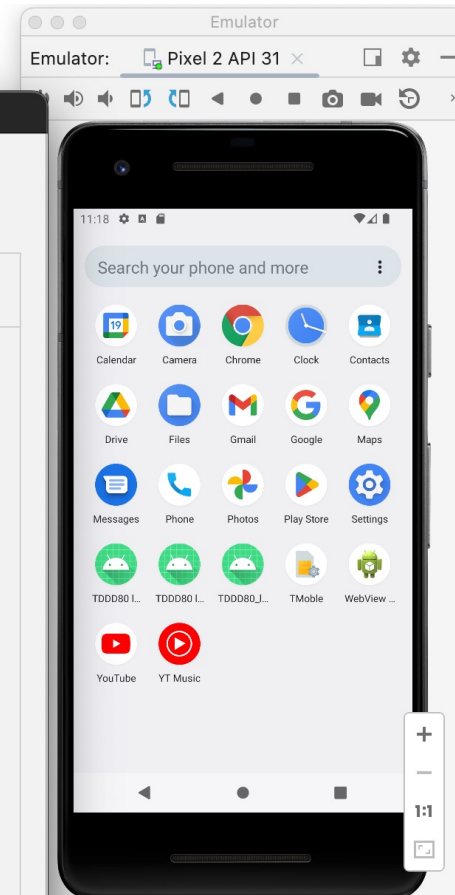
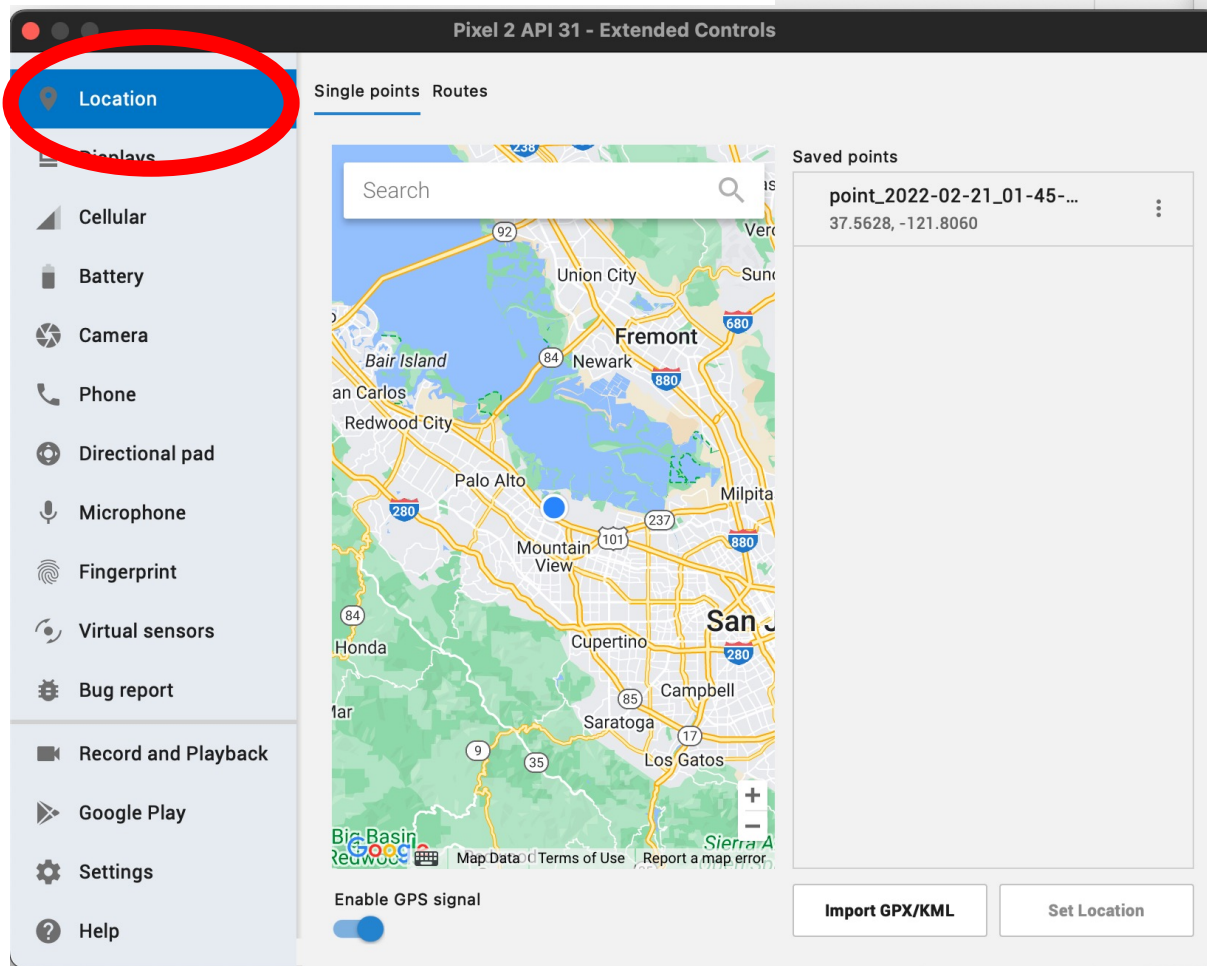


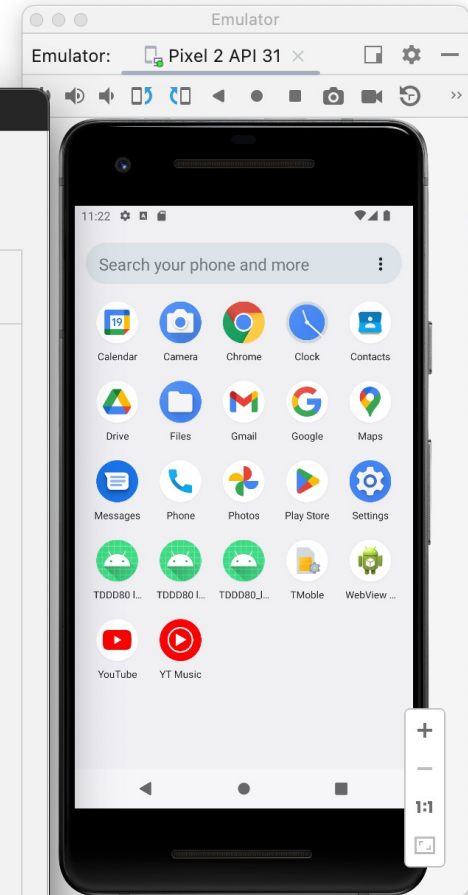
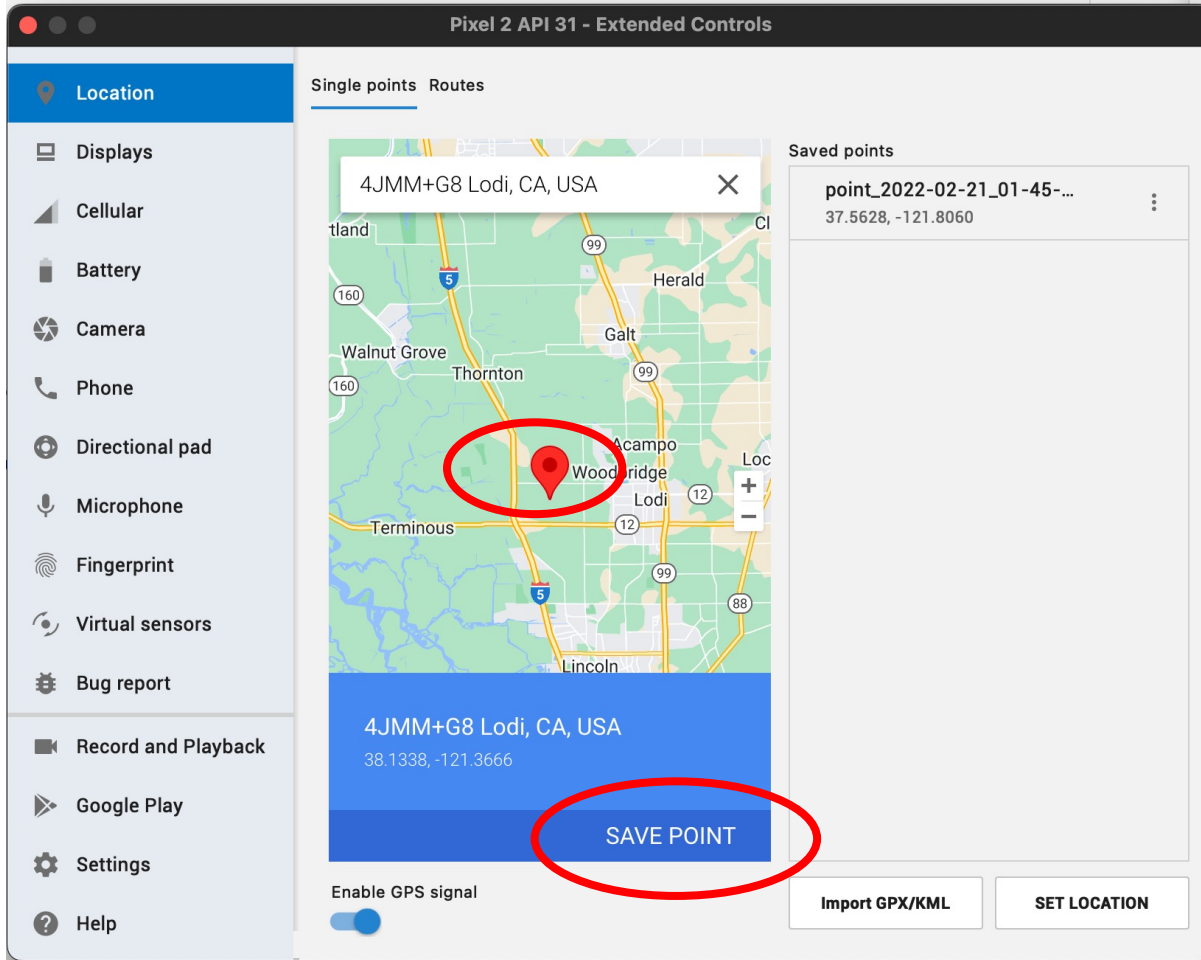
Egna appar



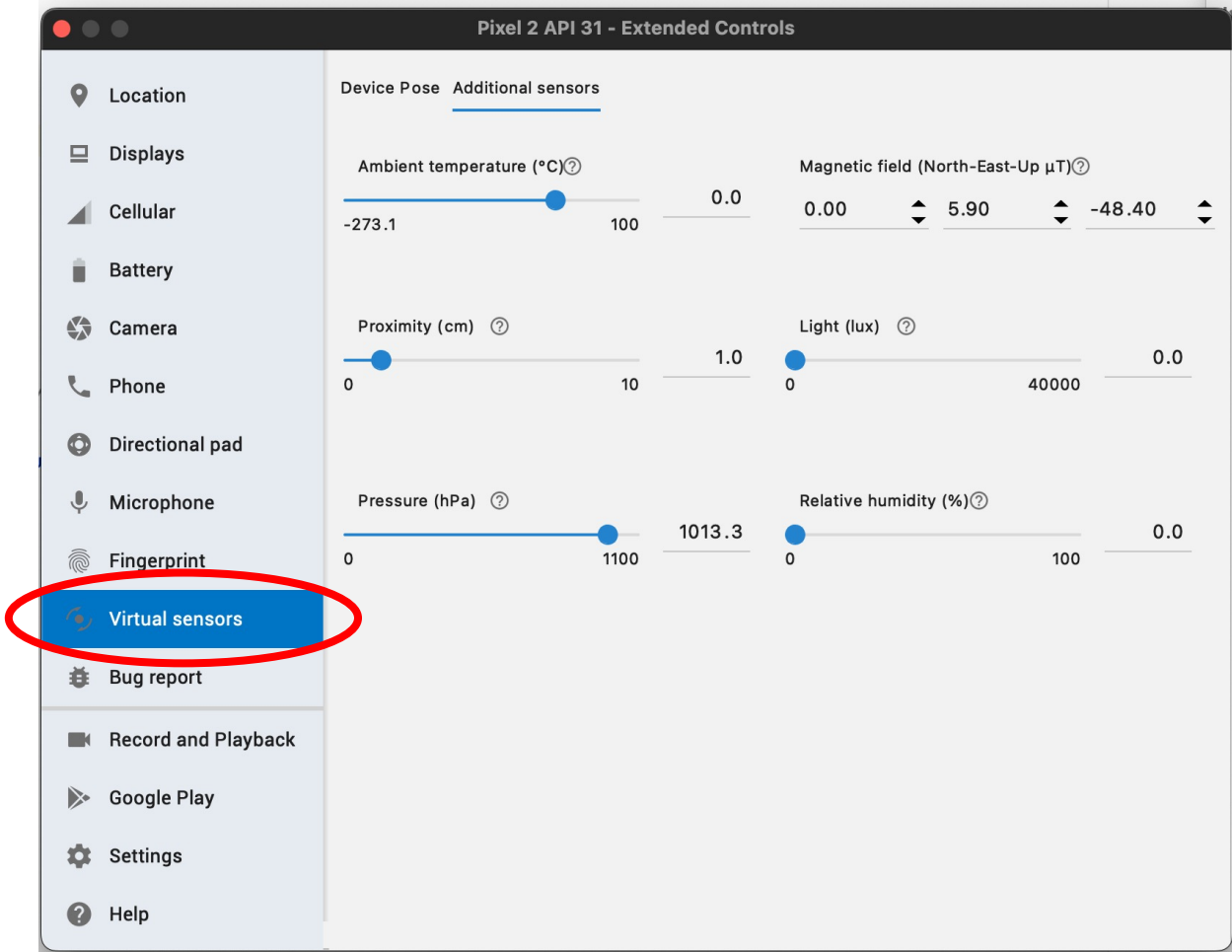
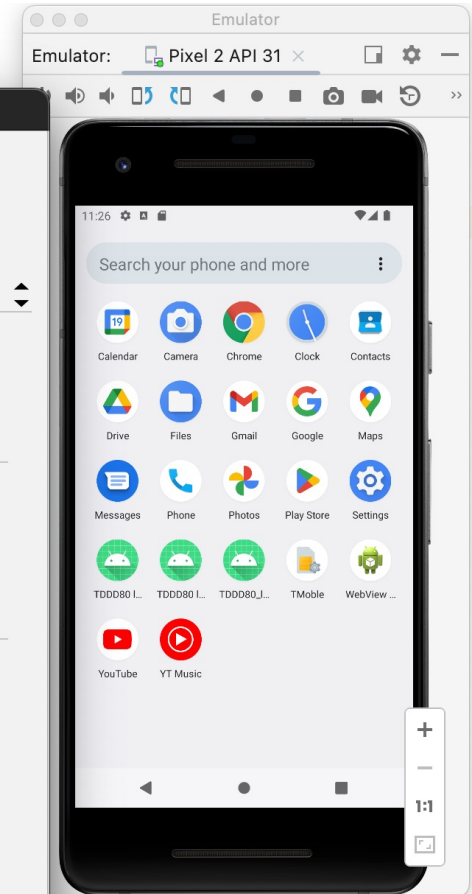
GPS...





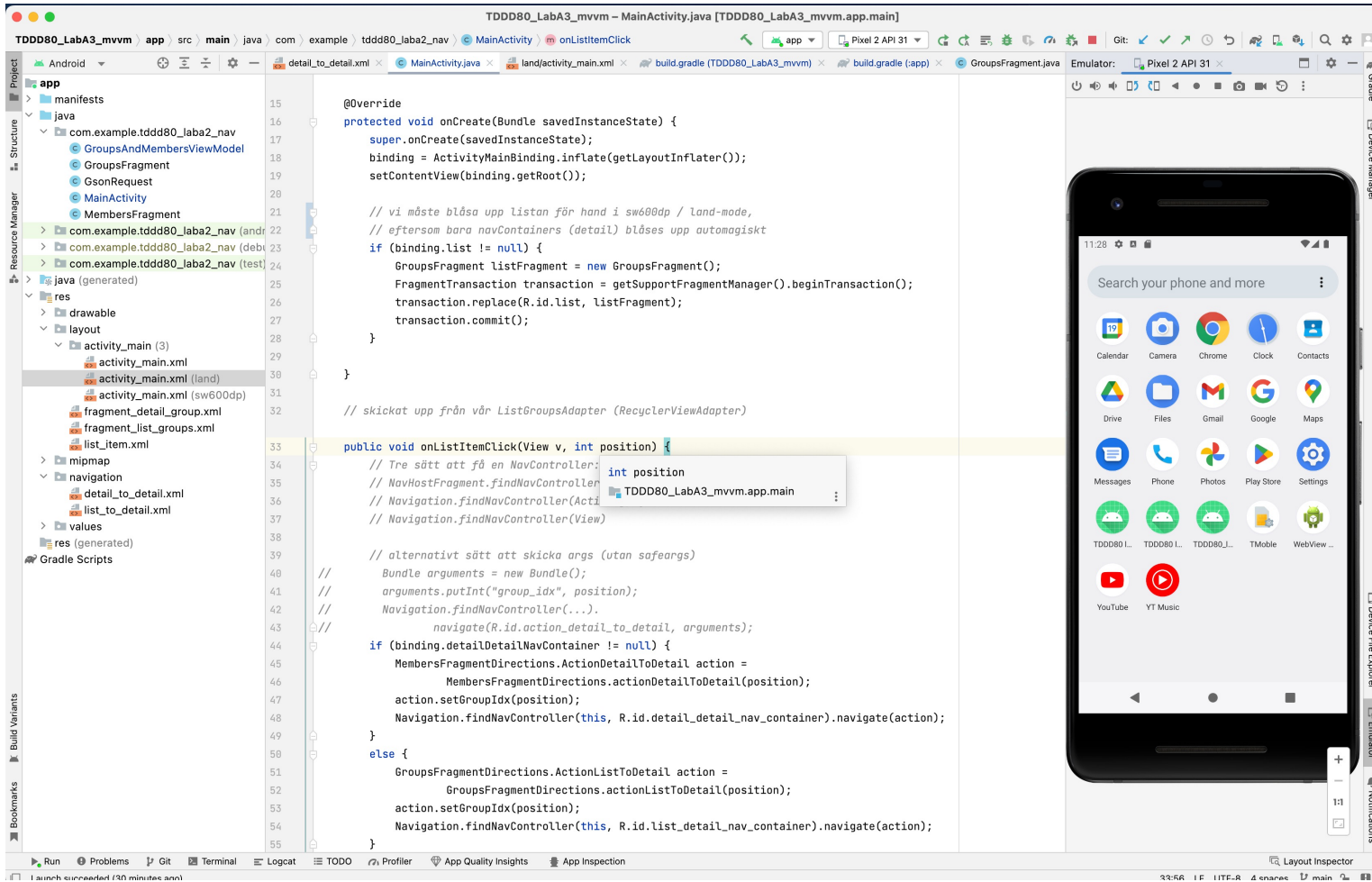


```
binding = ActivityMainBinding.inflate(getLayoutInflater());  
setContentView(binding.getRoot());
```

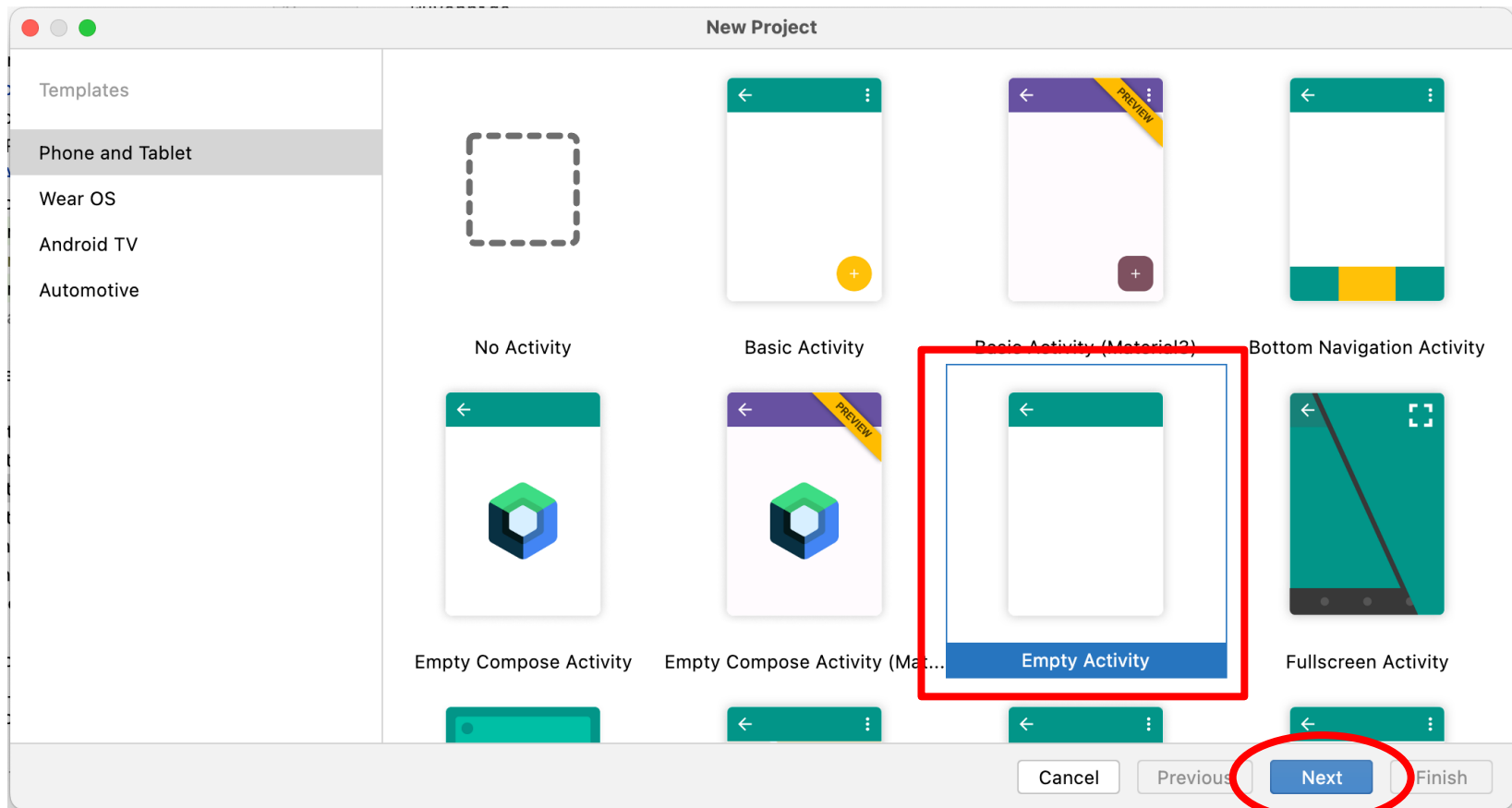


Android Studio introduktion

Android Studio



Skapa nytt projekt: File > New > New Project



Välj språk och API-nivå (= OS version)

New Project

Empty Activity

Creates a new empty activity

Name: LabA1

Package name: com.example.laba1

Save location: /Users/ritko75/Documents/development/AndroidStudioProjects/TDDD80/LabA1

Language: Java

Minimum SDK: API 24: Android 7.0 (Nougat)

i Your app will run on approximately **94,4%** of devices.
[Help me choose](#)

Use legacy android.support libraries **?**
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

Cancel Previous Next **Finish**

Samma som din telefon

API nivå i nyss skapade projektet

The screenshot shows the IDE interface for a new Android project named 'LabA1'. The 'Resource Manager' on the left shows the project structure, with 'build.gradle (Module :app)' highlighted. The main editor displays the 'build.gradle' file for the 'app' module. The code is as follows:

```
1 plugins {
2     id 'com.android.application'
3 }
4
5 android {
6     namespace 'com.example.laba1'
7     compileSdk 33
8
9     defaultConfig {
10        applicationId "com.example.laba1"
11        minSdk 24
12        targetSdk 33
13        versionCode 1
14        versionName "1.0"
15
16        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
17    }
18 }
```

A red box highlights the 'minSdk 24' line in the code. A red callout box with the text "Ska inte krascha på din telefon" (Shouldn't crash on your phone) points to the 'minSdk 24' line.

API-nivåer

Codename	Version	API level/NDK release
Android13	13	API level 33
Android12L	12	API level 32
Android12	12	API level 31
Android11	11	API level 30
Android10	10	API level 29
Pie	9	API level 28
Oreo	8.1.0	API level 27
Oreo	8.0.0	API level 26
Nougat	7.1	API level 25
Nougat	7.0	API level 24
Marshmallow	6.0	API level 23
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17

minSdkVersion

- Ska kunna köras på dessa (gamla) enheter utan att krascha
 - Dvs. är **inte** beroende av API-features högre än denna
- Kommer att kräva användning av kompatibilitetsklasser och metoder i app-koden

Android Platform/API Version Distribution

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,8%
4.3 Jelly Bean	18	99,5%
4.4 KitKat	19	99,4%
5.0 Lollipop	21	98,0%
5.1 Lollipop	22	97,3%
6.0 Marshmallow	23	94,1%
7.0 Nougat	24	89,0%
7.1 Nougat	25	85,6%
8.0 Oreo	26	82,7%
8.1 Oreo	27	78,7%
9.0 Pie	28	69,0%
10. Q	29	50,8%
11. R	30	24,3%

The minimum SDK version determines the lowest level of Android that your app will run on.

You typically want to target as many users as possible, so you would ideally want to support everyone -- with a minimum SDK version of 1. However, that has some disadvantages, such as lack of features, and very few people use devices that old anymore.

Your choice of minimum SDK level should be a tradeoff between the distribution of users you wish to target and the features that your application will need.

Click each Android Version/API level for more information.

Cancel

OK

API nivå i nyss skapade projektet

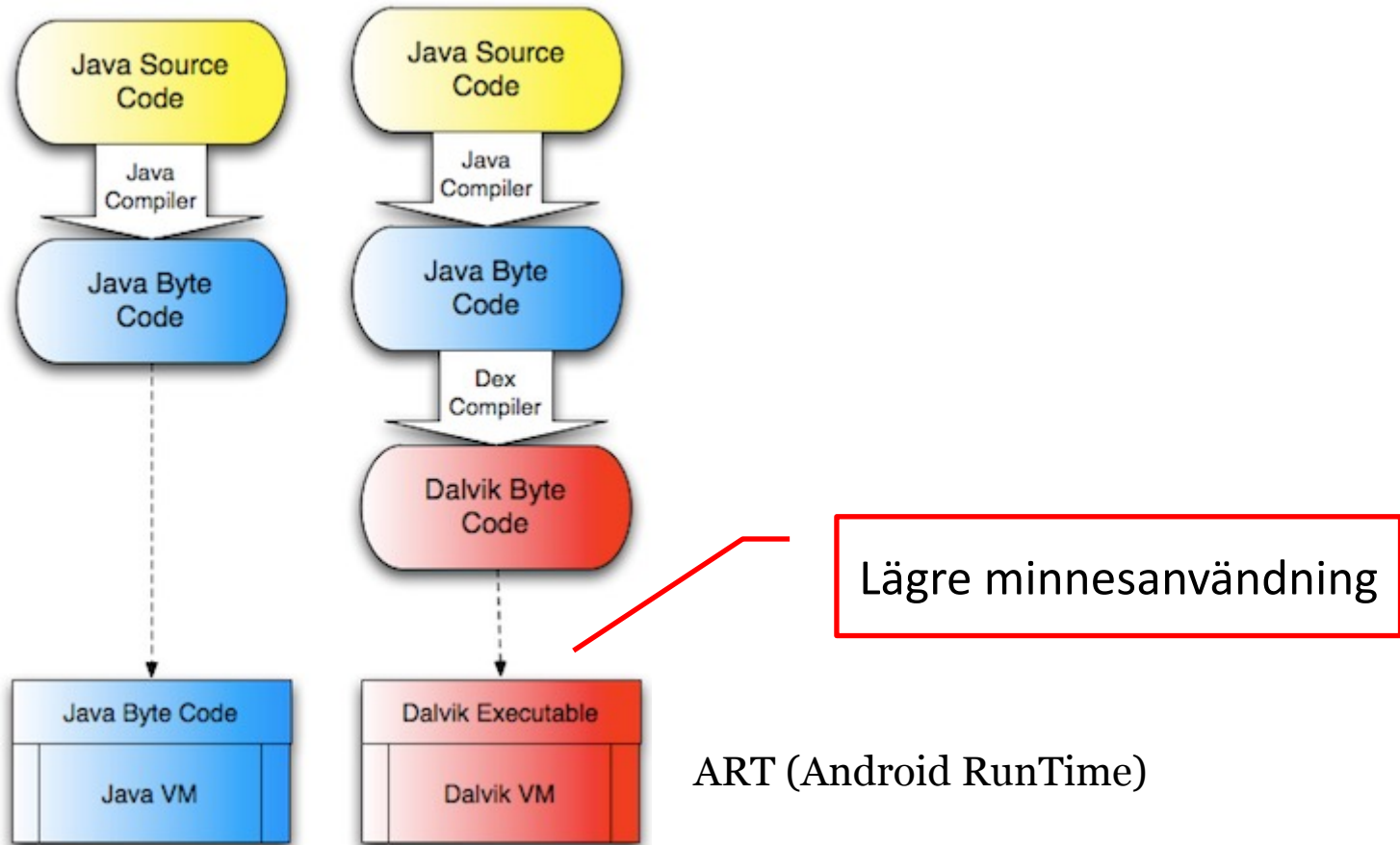
The screenshot shows the IDE interface for a new Android project named 'LabA1'. The 'build.gradle' file for the module ':app' is open, showing the following configuration:

```
1 plugins {
2     id 'com.android.application'
3 }
4
5 android {
6     namespace 'com.example.laba1'
7     compileSdk 33
8
9     defaultConfig {
10        applicationId "com.example.laba1"
11        minSdk 24
12        targetSdk 33
13        versionCode 1
14        versionName "1.0"
15
16        testInstrumentationRunn
17    }
18 }
```

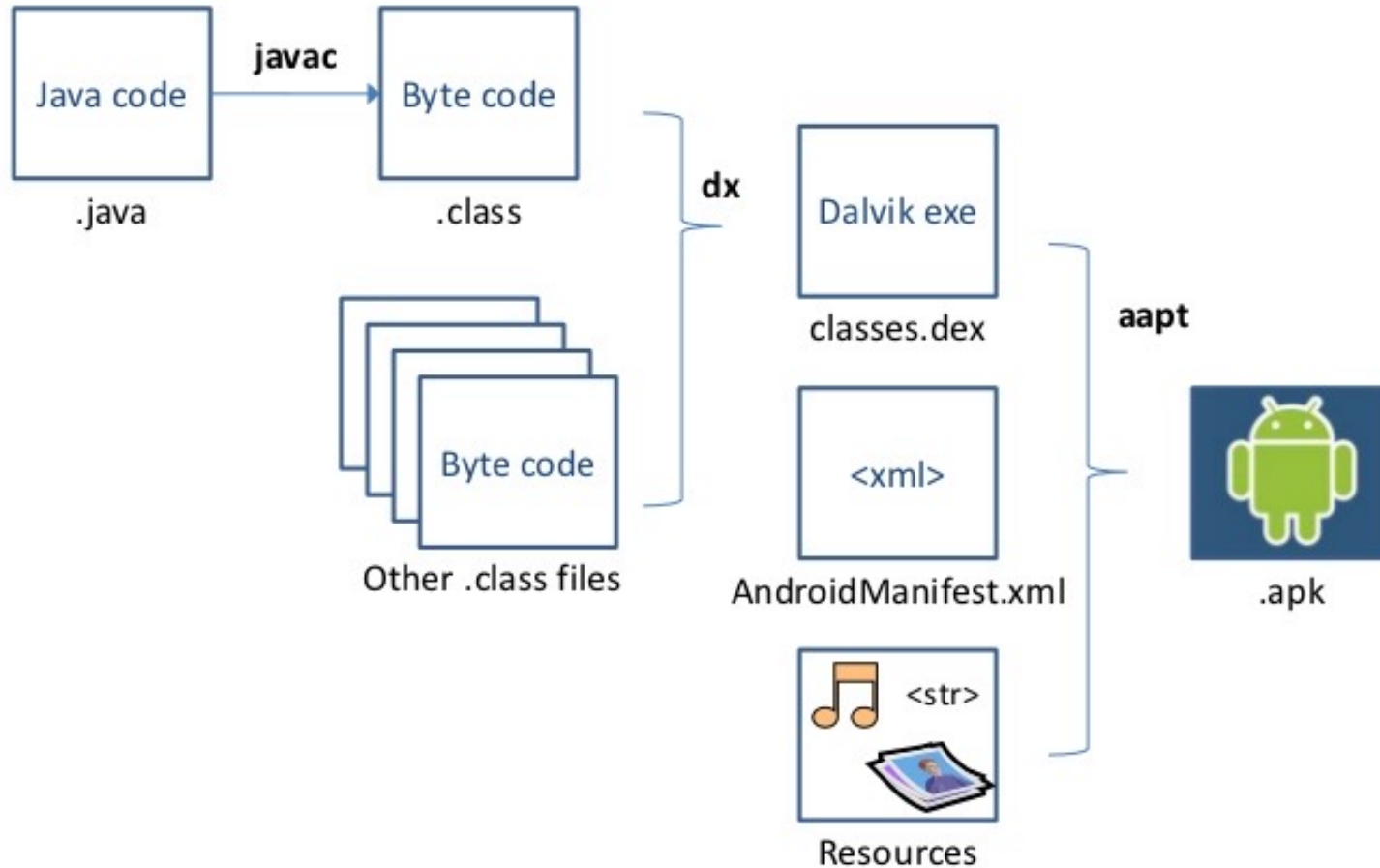
Annotations in the image:

- A red box around `compileSdk 33` has an arrow pointing to the text: "Kommer kompilera med senaste verktyget".
- A red box around `targetSdk 33` has an arrow pointing to the text: "Ska utnyttja högre API-features på bättre telefoner".

Java – Android



Android app



Kompilering

- Sköts av Gradle
 - Kompilerar och länkar in de bibliotek som används i appen
 - Olika inställningsfiler för Gradle
 - Talar om vilka moduler som ska kompileras
 - Vilka bibliotek man vill ha med

Projektstruktur

MyApp/

Project

build.gradle

settings.gradle

app/

Module

build.gradle

build/

libs/

src/

main/

Sourceset

java/

com.example.myapp/

res/

drawable/

layout/

...

AndroidManifest.xml

Två nivåer av build.gradle

- Projekt: Googles repo, övergripande dependencies

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
buildscript {
    repositories {
        google()
    }
    dependencies {
        def nav_version = "2.5.3"
        classpath "androidx.navigation:navigation-safe-args-gradle-plugin:$nav_version"
    }
}

plugins {
    id 'com.android.application' version '7.4.1' apply false
    id 'com.android.library' version '7.4.1' apply false
}
```

Två nivåer av build.gradle

- App: Plugins och bibliotek för just din app

```
plugins {  
    id 'com.android.application'  
    id 'androidx.navigation.safeargs'  
}  
  
android {  
    compileSdk 33  
  
    defaultConfig {  
        applicationId "com.example.tddd80_laba3_mvvm"  
        minSdk 31  
        targetSdk 31  
        versionCode 1  
        versionName "1.0"  
  
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
    }  
}
```

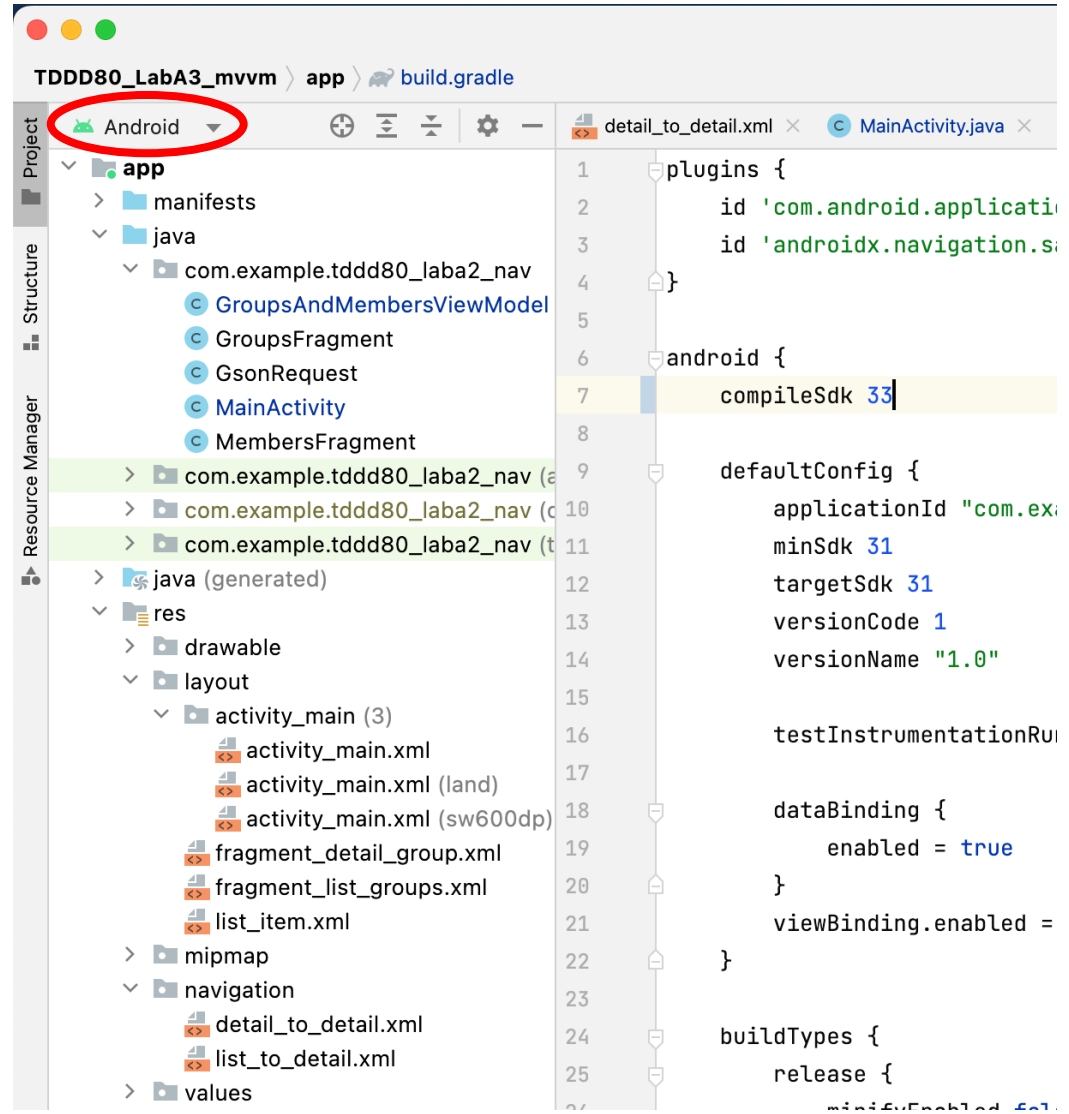
Alla filer ska inte versionshanteras...

- Skapa en `.gitignore` fil där ni anger vad som inte ska med i git
 - Vill t.ex. inte ha kompilerade filer
 - Filer som autogenereras
 - IDE-specifik meta information
- Exempel på bra `.gitignore` för Android-projekt:
 - <https://github.com/github/gitignore/blob/master/Android.gitignore>

Projektstruktur

Kod

Resurser



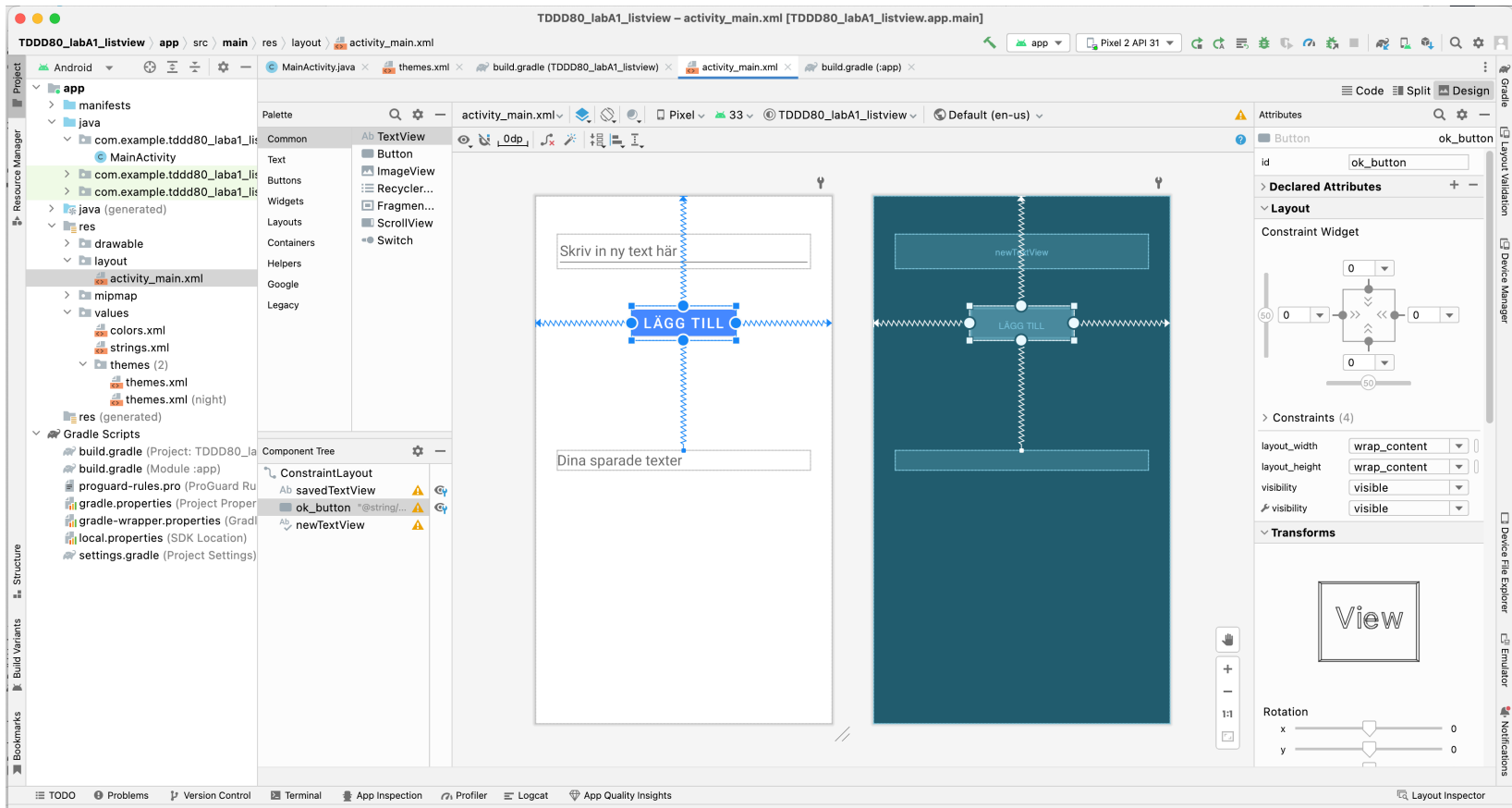
The screenshot displays the project structure of an Android application in an IDE. The top toolbar shows the 'Android' icon circled in red. The project structure on the left is as follows:

- app
 - manifests
 - java
 - com.example.tddd80_laba2_nav
 - GroupsAndMembersViewModel
 - GroupsFragment
 - GsonRequest
 - MainActivity
 - MembersFragment
 - com.example.tddd80_laba2_nav (e)
 - com.example.tddd80_laba2_nav (c)
 - com.example.tddd80_laba2_nav (t)
 - java (generated)
 - res
 - drawable
 - layout
 - activity_main (3)
 - activity_main.xml
 - activity_main.xml (land)
 - activity_main.xml (sw600dp)
 - fragment_detail_group.xml
 - fragment_list_groups.xml
 - list_item.xml
 - mipmap
 - navigation
 - detail_to_detail.xml
 - list_to_detail.xml
 - values

The main editor shows the `build.gradle` file with the following content:

```
1 plugins {
2     id 'com.android.application'
3     id 'androidx.navigation.safeargs'
4 }
5
6 android {
7     compileSdk 33
8
9     defaultConfig {
10        applicationId "com.ex
11        minSdk 31
12        targetSdk 31
13        versionCode 1
14        versionName "1.0"
15
16        testInstrumentationRun
17
18        dataBinding {
19            enabled = true
20        }
21        viewBinding.enabled =
22    }
23
24    buildTypes {
25        release {
26            minifyEnabled false
27    }
```

Resurser > layout-mapp > activity_main.xml



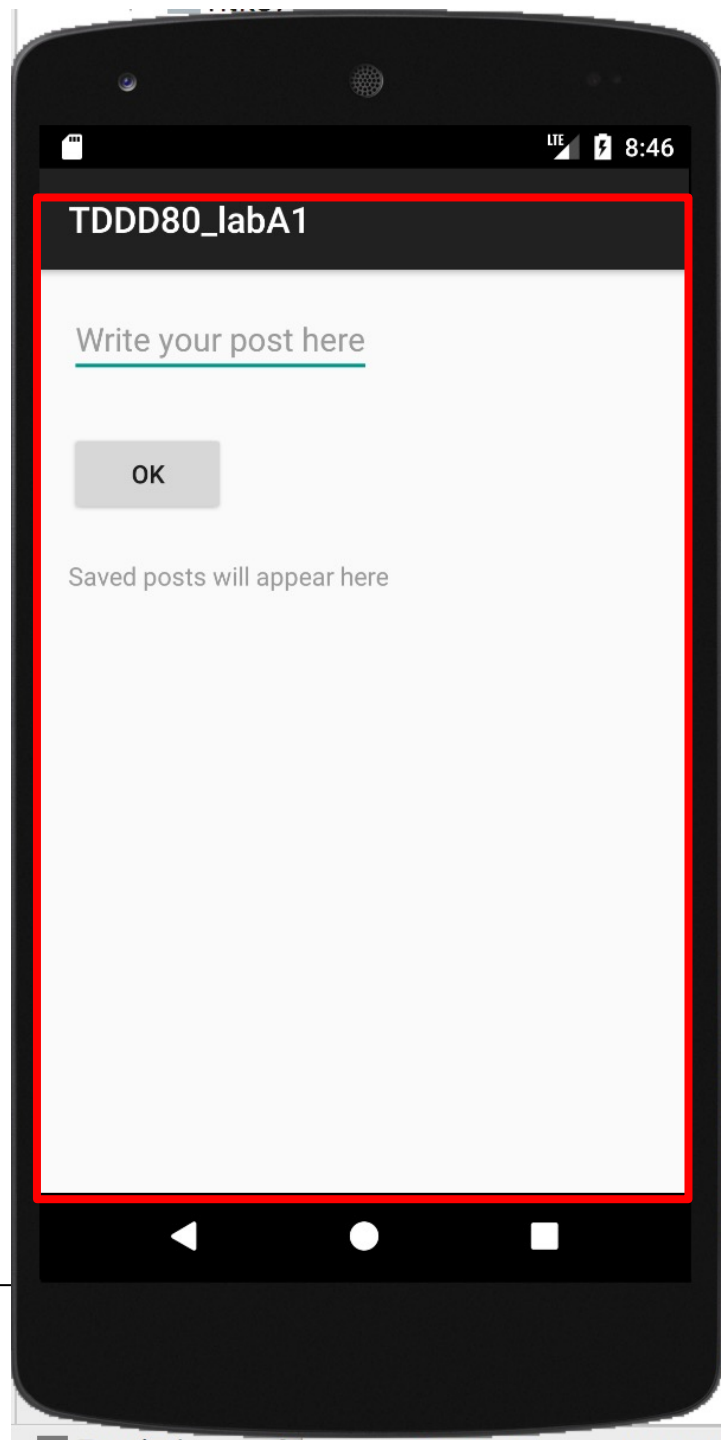
Constraint layout tutorial

- <https://www.youtube.com/watch?v=XamMbnzI5vE>

Android kodning

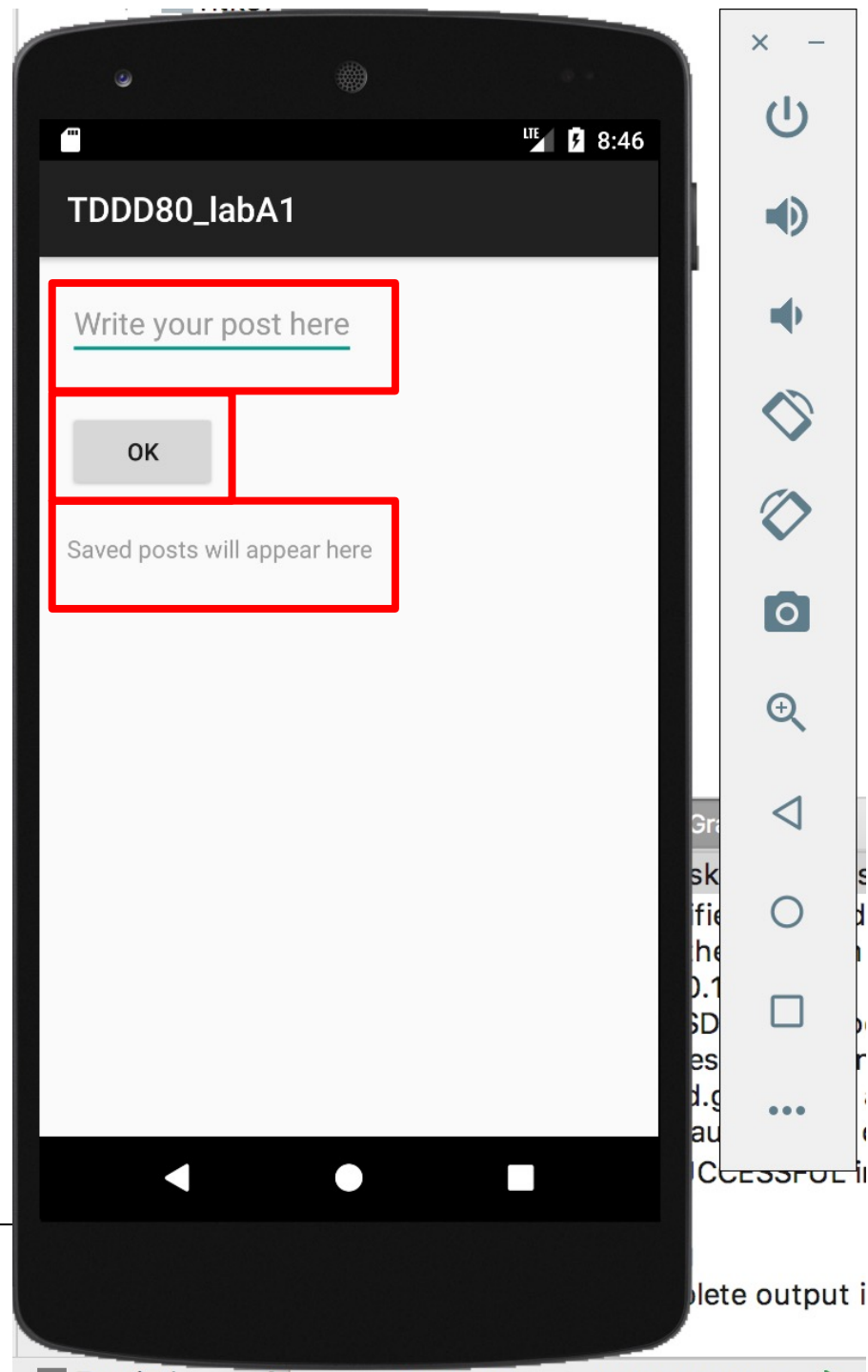
Android labb 1

Activity



Android labb 1

Views: textfält, knappar



Deklarativ layout

- Skalet, dvs. användargränssnittet (UI) på appen: t.ex. textfält, knappar
 - Beskrivs i separat ”textfil” (XML)
- Beteendet
 - Klickhantering, etc. beskrivs i Java-kod

XML

- eXtensible Markup Language
- Liknar html, men...
 - För vilka data som helst
 - Vendor (Android) bestämmer taggar och attribut

HTML: `...`

XML: `<step number="3">...</step>`

Android UI layout i XML

```
<LinearLayout xmlns:android="http://schemas.android.com/..."  
    android:orientation="vertical"
```

```
<EditText  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/inputText"  
    android:inputType="textAutoCorrect"  
    android:hint="@string/editTextHint" />
```

id används för att
hitta denna View
från java-koden

```
</LinearLayout>
```

XML vs. kod

- XML
 - Statisk placering och utseende på komponenter (dvs. kan inte ta bort under körning)
 - Tomma behållare (t.ex. `FrameLayout`) kan sedan fyllas dynamiskt från java-kod
- Java kod
 - Dynamiskt tillägg/borttagning av element
 - Fyllning med innehåll

Binding – binder xml till javakod

- I app build.gradle, lägg till dataBinding

```
android {  
    compileSdkVersion 29  
  
    defaultConfig {  
        applicationId "com.example.tddd80_laba1_listview"  
        minSdkVersion 26  
        targetSdkVersion 29  
        versionCode 1  
        versionName "1.0"  
  
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
    }  
  
    buildFeatures {  
        viewBinding true  
    }  
  
    buildTypes {
```

I koden utförs bindningen

```
public class MainActivity extends AppCompatActivity {
```

```
    ActivityMainBinding binding;
```

Autogenererad klass

var dekl

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        binding = ActivityMainBinding.
```

Blås liv i xml:en

metod

```
            inflate(getLayoutInflater());
```

```
        setContentView(binding.getRoot());
```

Bind genom root-view

activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/..."  
    android:orientation="vertical"
```

```
    <EditText  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:id="@+id/editText"  
        android:inputType="textAutoCorrect"  
        android:hint="@string/editTextHint" />
```

```
</LinearLayout>
```

Hitta textfält i XML-layouten

```
public class MainActivity extends AppCompatActivity {  
    ActivityMainBinding binding;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        ...  
    }  
  
    private void saveText() {  
        autoCompleteTextView newTextV = binding.editText;  
        ...  
    }  
}
```

Fånga upp klick (enkelt)

- I XML:

- `<Button`

- `android:layout_width="wrap_content"`

- `android:layout_height="wrap_content"`

- `android:text="@string/button_send"`

- `android:onClick="sendMessage" />`**

- I MainActivity.java:

- `public void sendMessage(View view) {...}`

Fånga upp klick (rekommenderat)

```
public class MainActivity extends AppCompatActivity {
    ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        Button button = binding.okButton;
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                saveText();
            }
        });
    }
}
```

```
private void saveText() {
```

```
...
```

Fördelar med denna lösning

- Anonym lyssnare
 - Kan ha olika typer av lyssnare kopplade till olika knappar, input-fält, etc.
 - Blir inte rörigt i klassen
- Skickar anropet direkt vidare till en lokal metod
 - Kan själv också anropa metoden utan att behöva klicka på knappen varje gång
 - Underlättar testning

Activity lifecycles

Vad som skiljer Android från Java

- Resurssnål
 - Appar i bakgrunden dödas när minnet behövs för annat
 - Så fort appen hamnar i bakgrunden är det fritt fram för Android att döda den
 - Jfr. desktop-app som körs tills **användaren** klickar Quit
- Responsiv
 - Appar får ett par sekunder på sig att reagera på användarinput, sedan avbryter operativsystemet (OS)

Ingen konstruktor, utan onCreate()

```
public class MainActivity extends AppCompatActivity {
```

```
    ActivityMainBinding binding;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        binding = ActivityMainBinding.
```

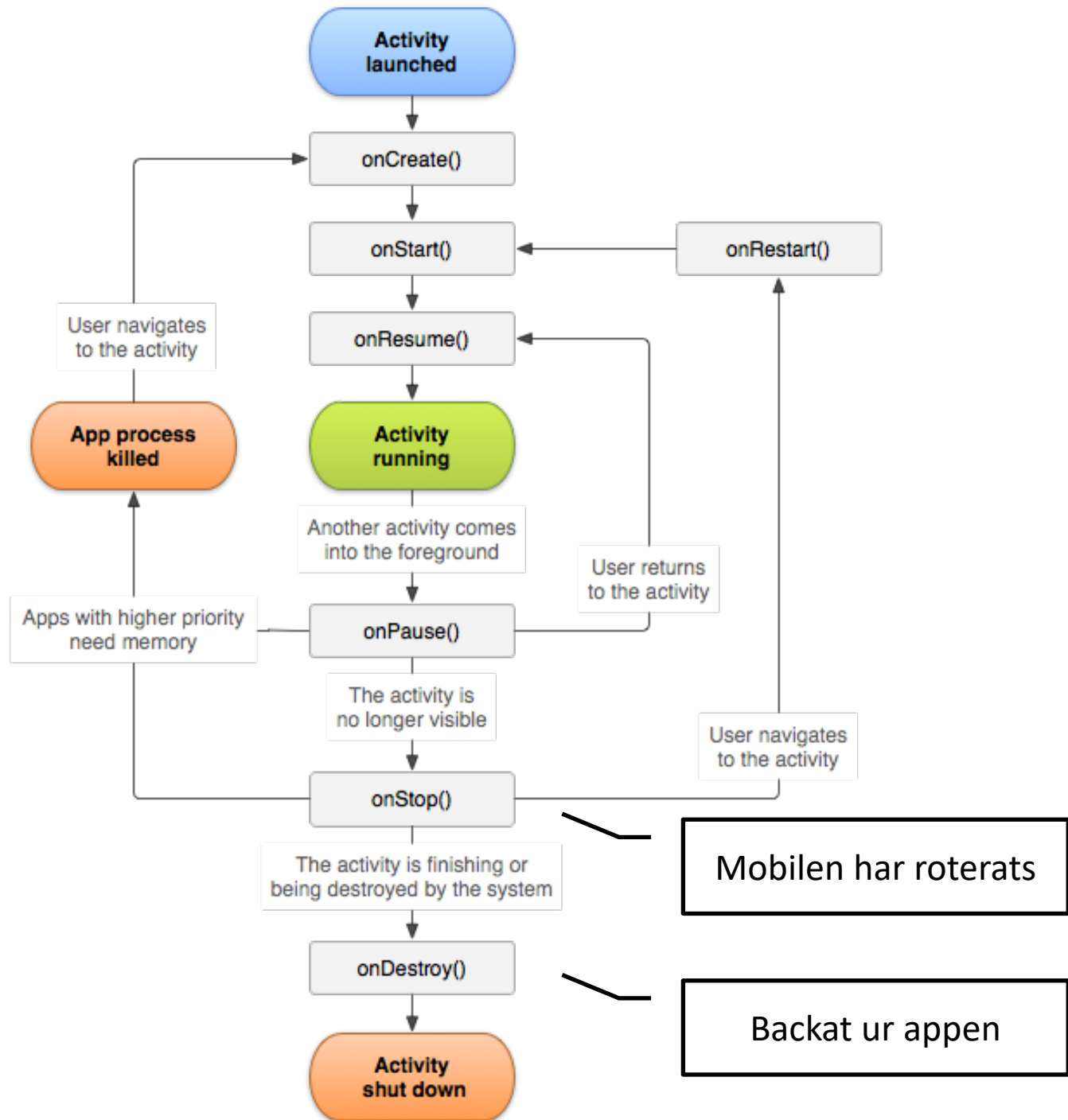
```
            inflate(getLayoutInflater());
```

```
        setContentView(binding.getRoot());
```

} metod

Aktiviteter skapas och dödas av Android

- Stoppas om
 - T.ex. telefonen roteras
- Dödas om
 - T.ex. appen hamnat i bakgrunden (och ont om minne)
- Spara undan tillstånd i god tid. Två sätt:
 - Ge dina vyer Id, och låt systemet spara dessa vyer
 - Override på `onPause()` el. `onSaveInstanceState(Bundle)`



Metoder för onPause(), etc.

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(...) {
```

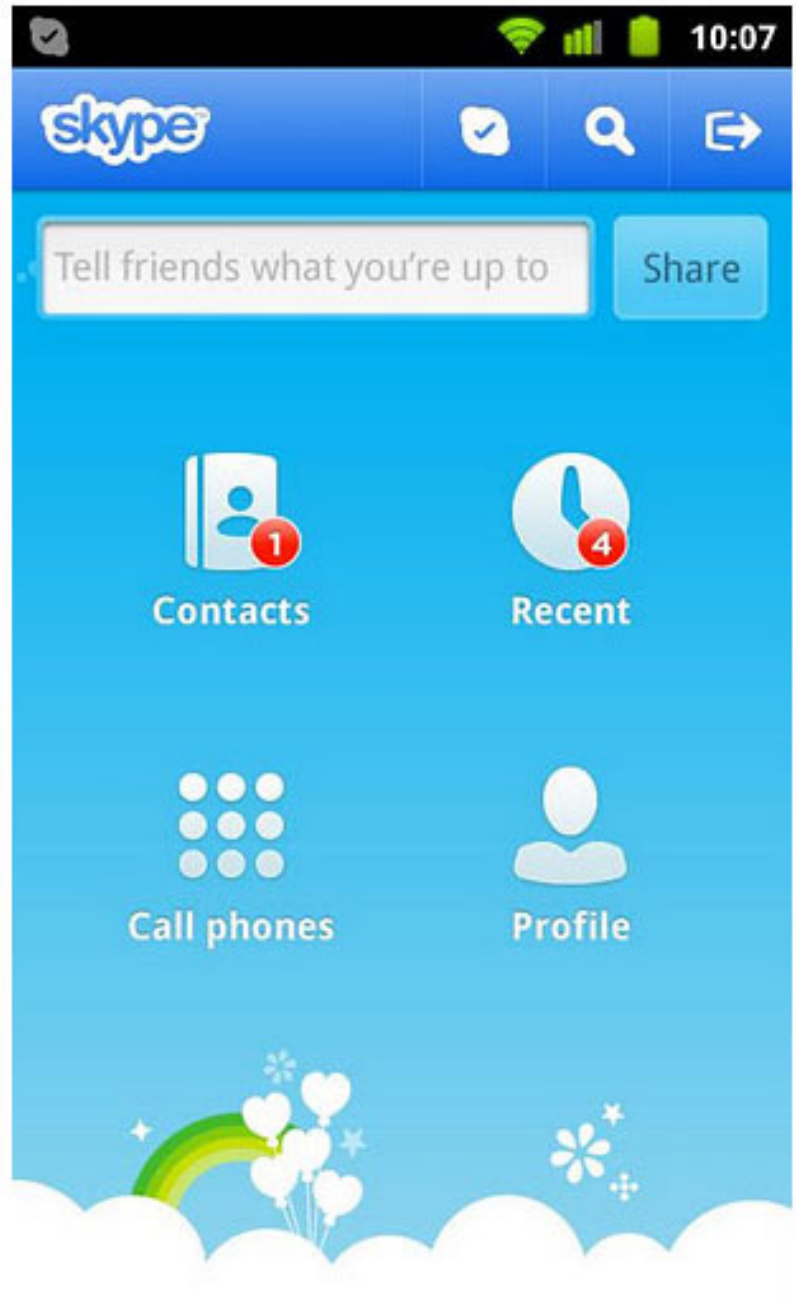
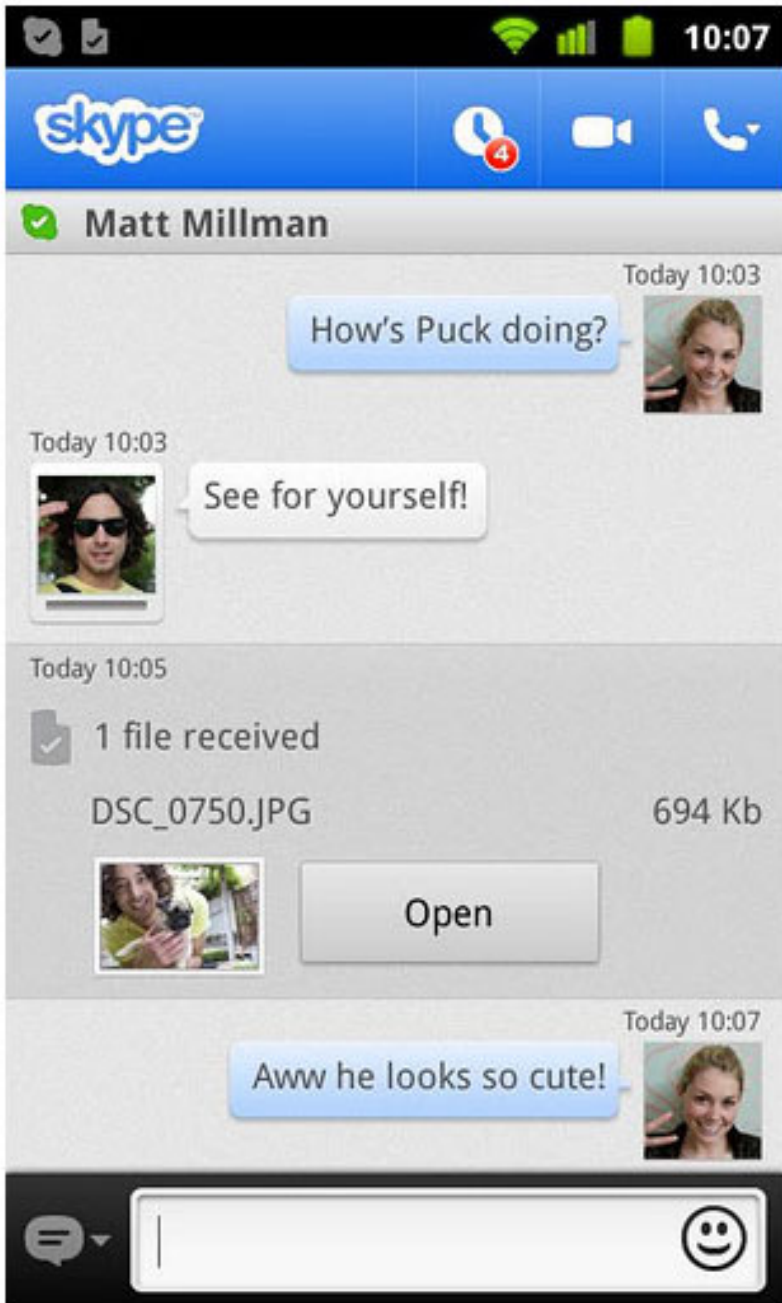
```
        ... // koppla till xml, hitta textfält, etc.
```

```
    @Override
```

```
    protected void onPause(...) {
```

```
        ... // spara undan data, anv.val, etc. till backend
```

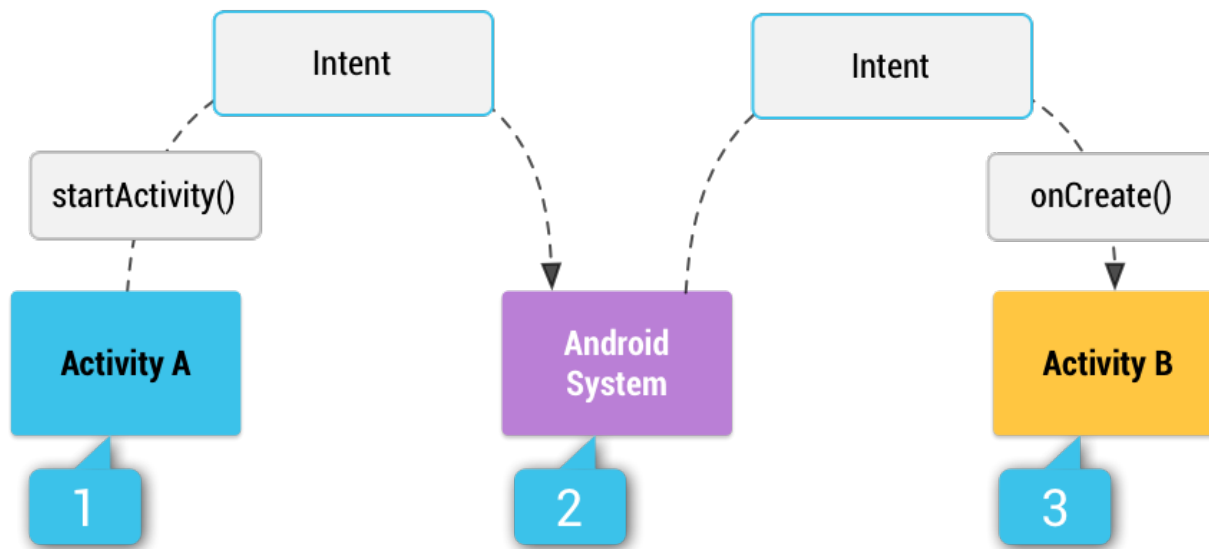
Anropa en annan aktivitet





“Anrop” med intent

- Går via OS
 - Vet inte alltid om finns aktivitet som kan ta emot
 - Android kan launcha annan app om nödvändigt



“Anrop” av aktiviteter med intent

- Förfrågan att få en beräkningsuppgift utförd



vill se foto, etc.

```
Intent intent = new Intent(Intent.ACTION_VIEW);
```

... fyll på med extra information att skicka med

```
startActivity(intent);
```

“Anrop” av aktiviteter med intent

- Starta aktivitet **i egna appen**

```
Intent intent = new Intent(this, MyNextActivity.class);  
... fyll på med extra information att skicka med  
startActivity(intent);
```

Exempel på intent

```
public void createAlarm(String message, int hour, int minutes) {  
    Intent intent = new Intent(AlarmClock.ACTION_SET_ALARM)  
        .putExtra(AlarmClock.EXTRA_MESSAGE, message)  
        .putExtra(AlarmClock.EXTRA_HOUR, hour)  
        .putExtra(AlarmClock.EXTRA_MINUTES, minutes);  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivity(intent);  
    }  
}
```

Testa om det finns
aktivitet därute som kan
svara på anropet

Att ta emot resultat från en aktivitet

- Callback:
 - Activity1 implementerar **onActivityResult()** som Android OS anropar när activity2 är klar

@Override

```
protected void onActivityResult(..., Intent data) {...}
```



Intent med resultat-data

Nästa gång

- Ytterligare komponenter
 - Fragments
- Fyll UI-komponenter med innehåll
- Nätverksanrop
 - Hålla appen synkad mot en server

www.liu.se