

Lecture #10

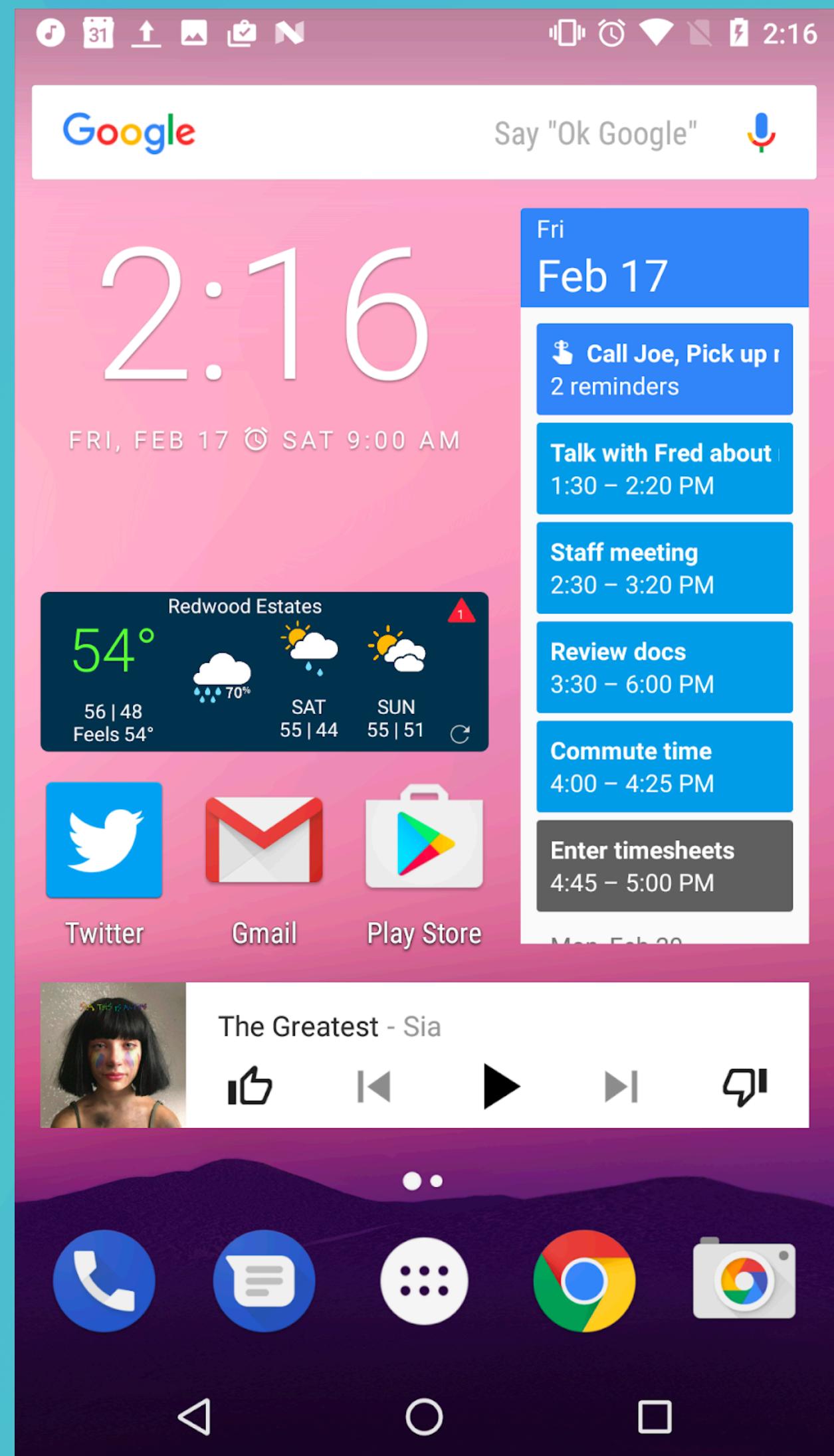
Advanced Mobile

Development

Mobile Applications 2020-2021

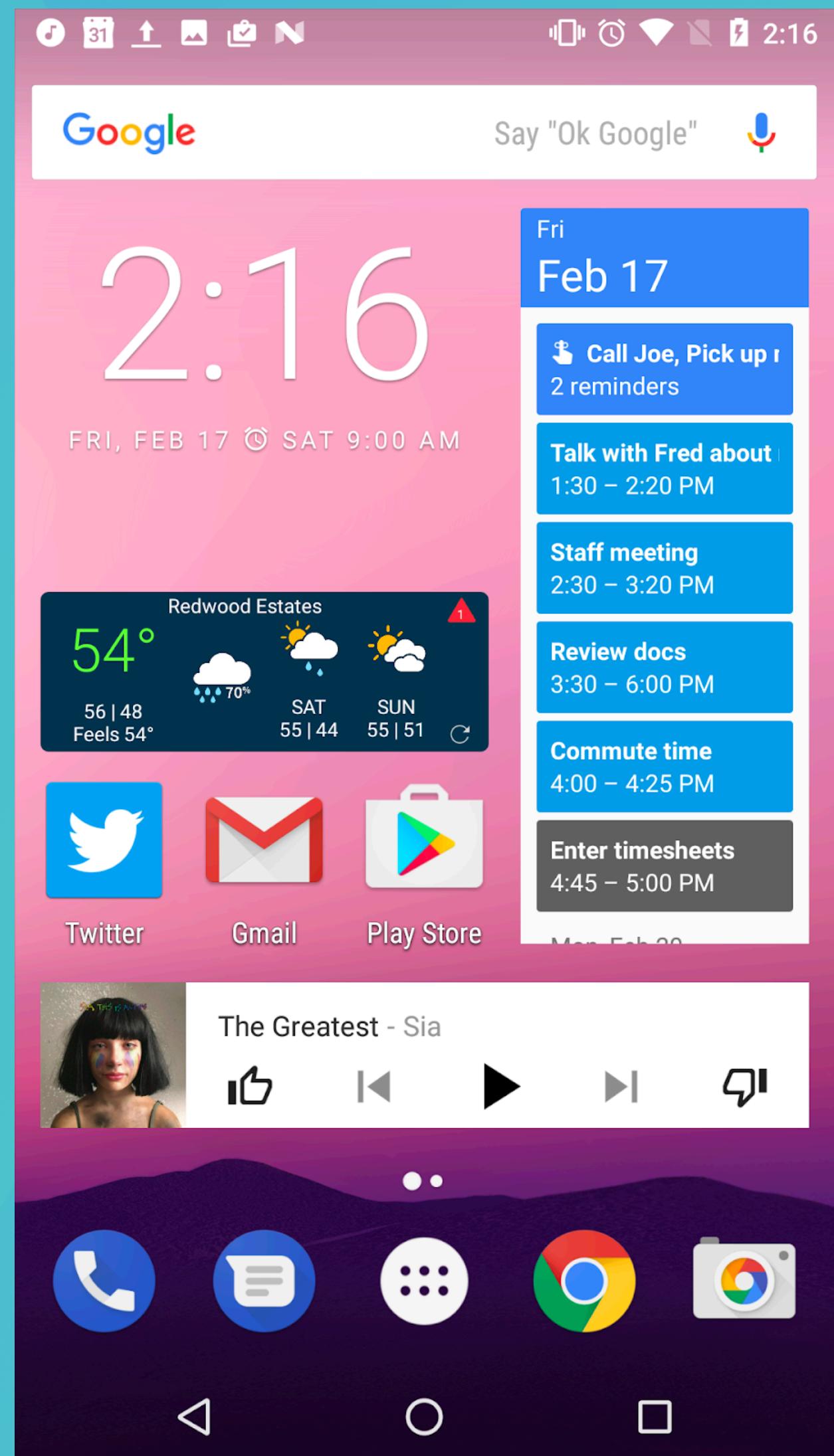
App Widgets

- A miniature app view.
- Runs on the home screen.
- Updated periodically.
- Display small amounts of information.
- Perform simple functions.



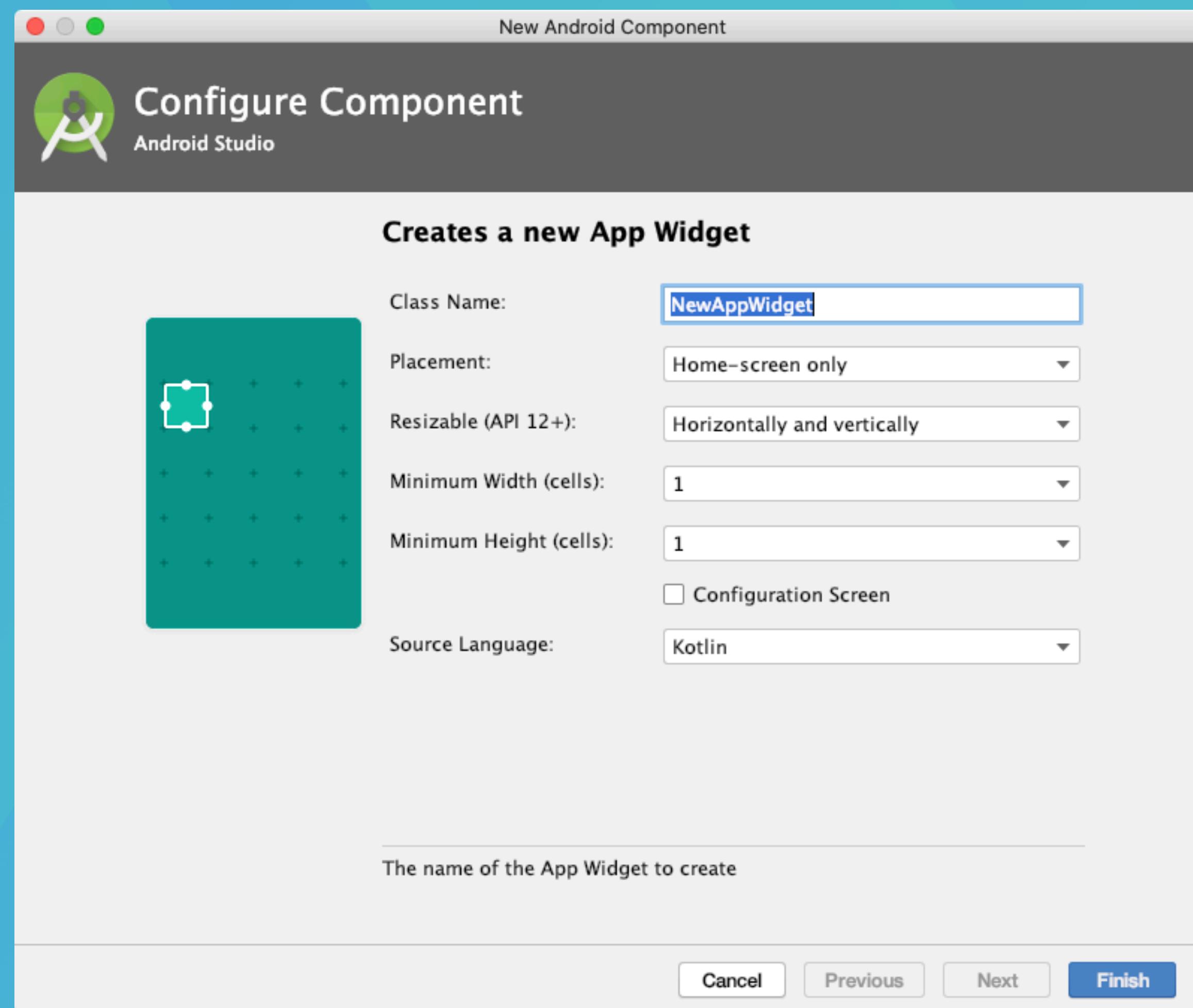
App Widgets

- Add-ons for an existing app.
- An app can have multiple widgets.
- Not available without an app.
- The default action is to start the app.

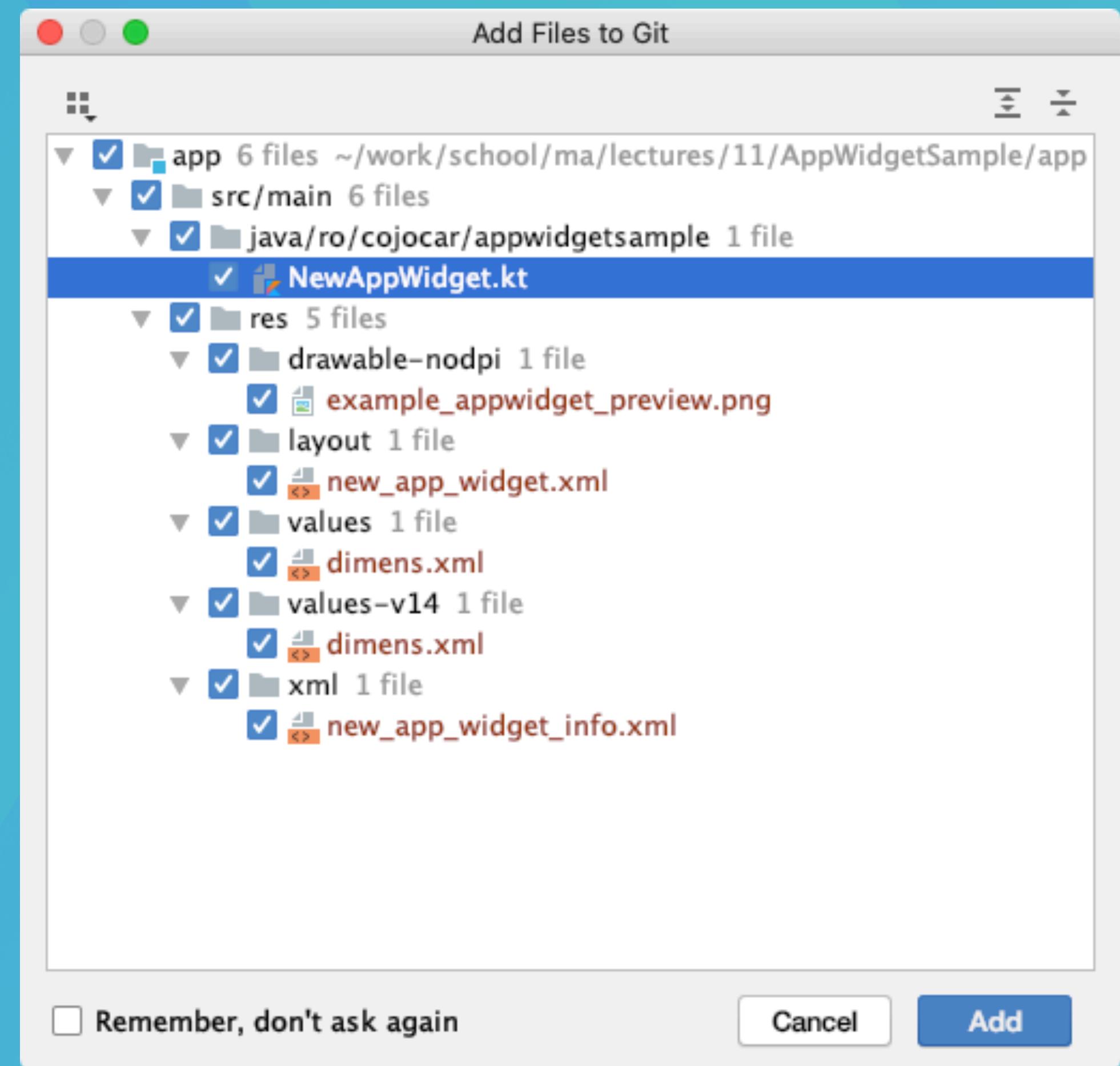


Set up the app widget project

- File > New > Widget > AppWidget.



Generated Files

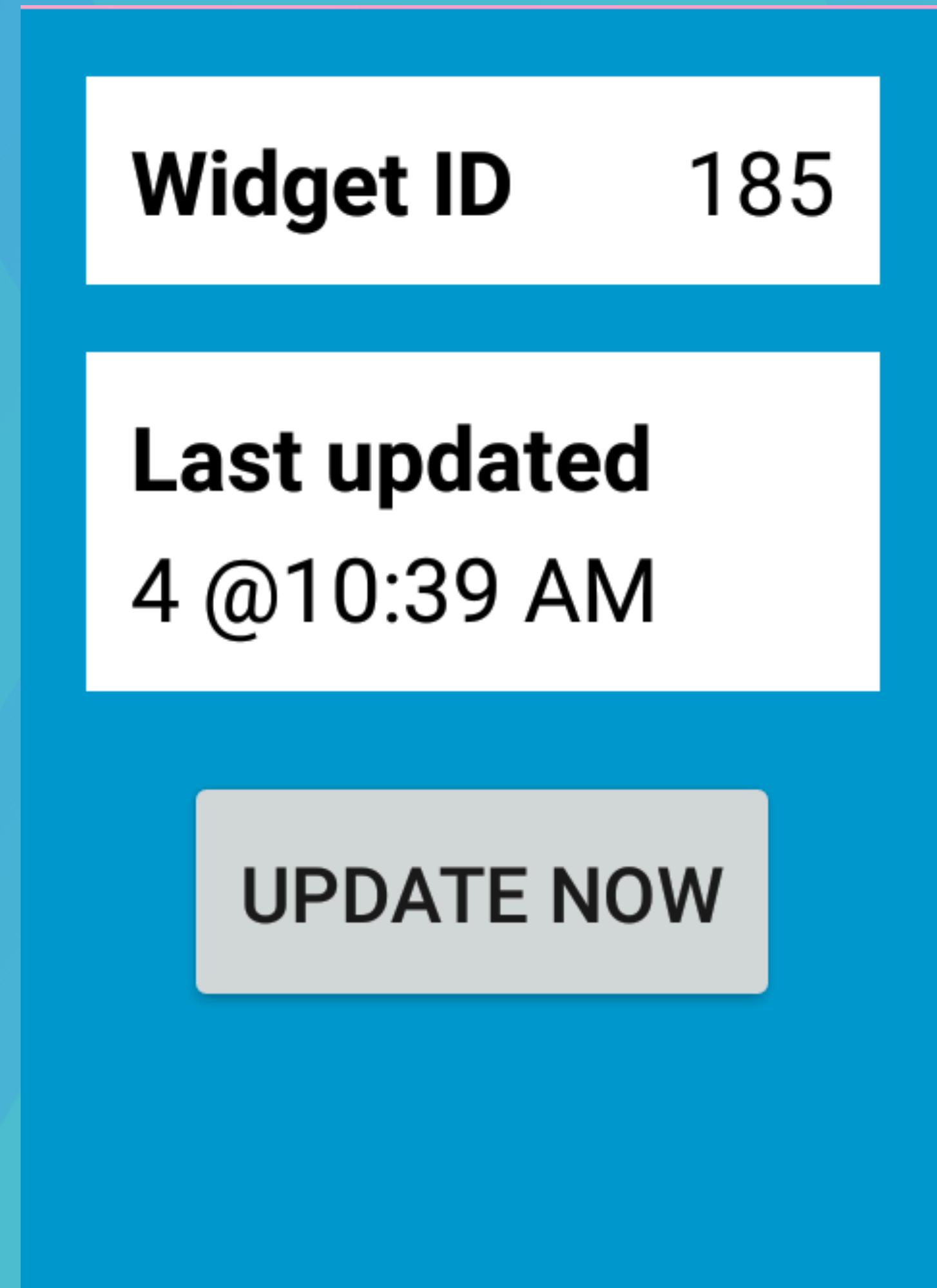


twitch.tv/dancojocar

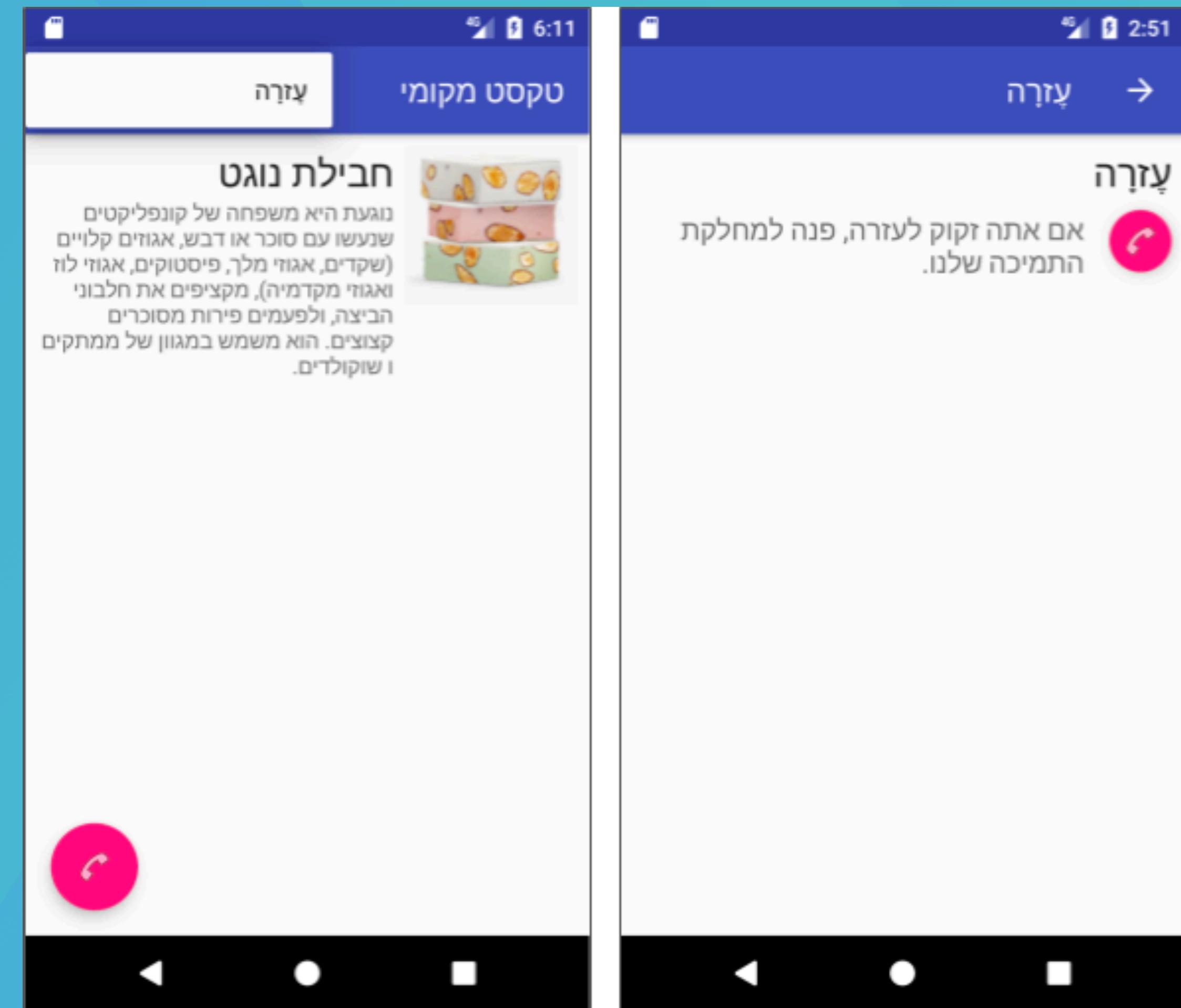
youtube.com/dancojocar

DEMO

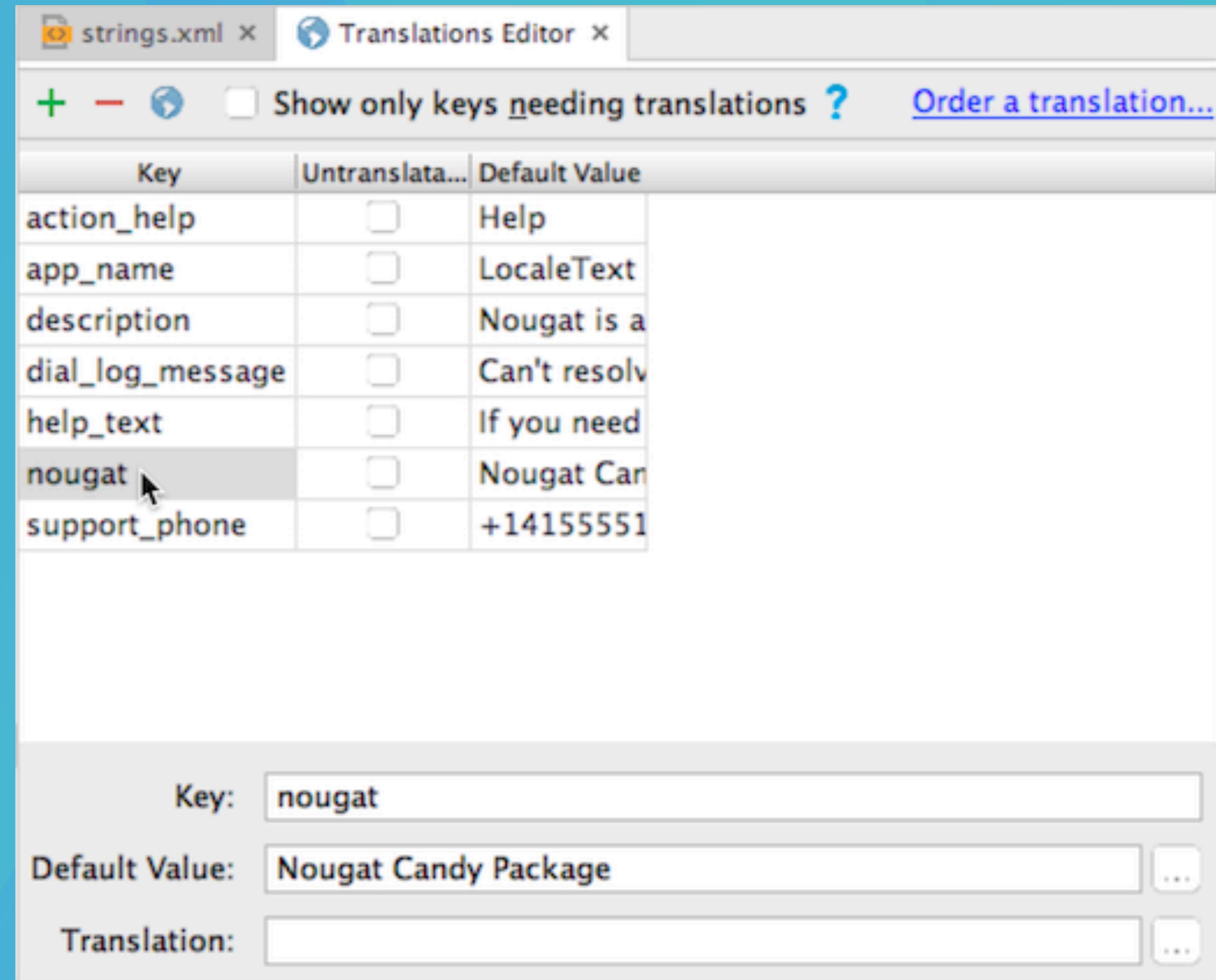
Customize the widget



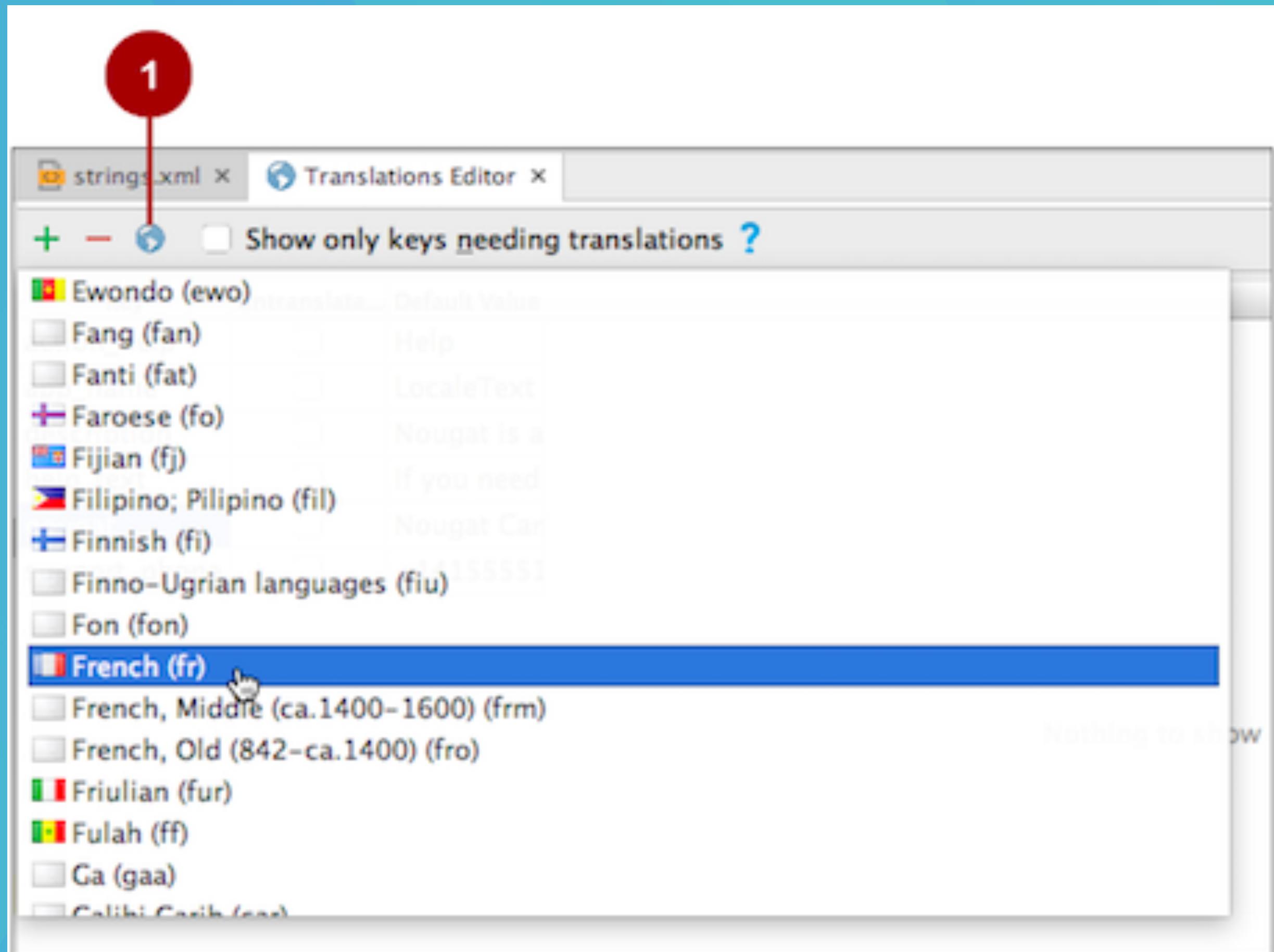
Language Support



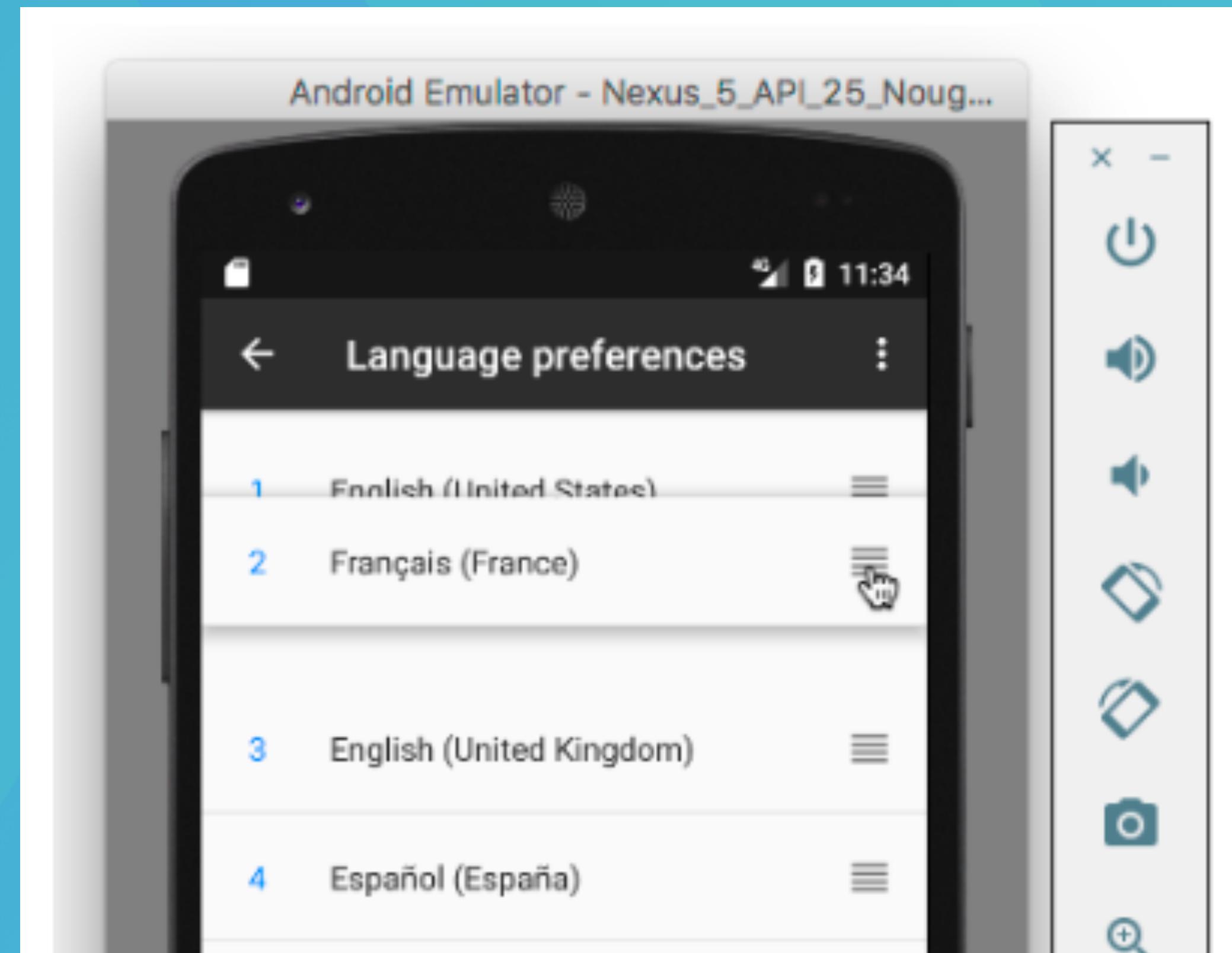
Add another language resource to the app



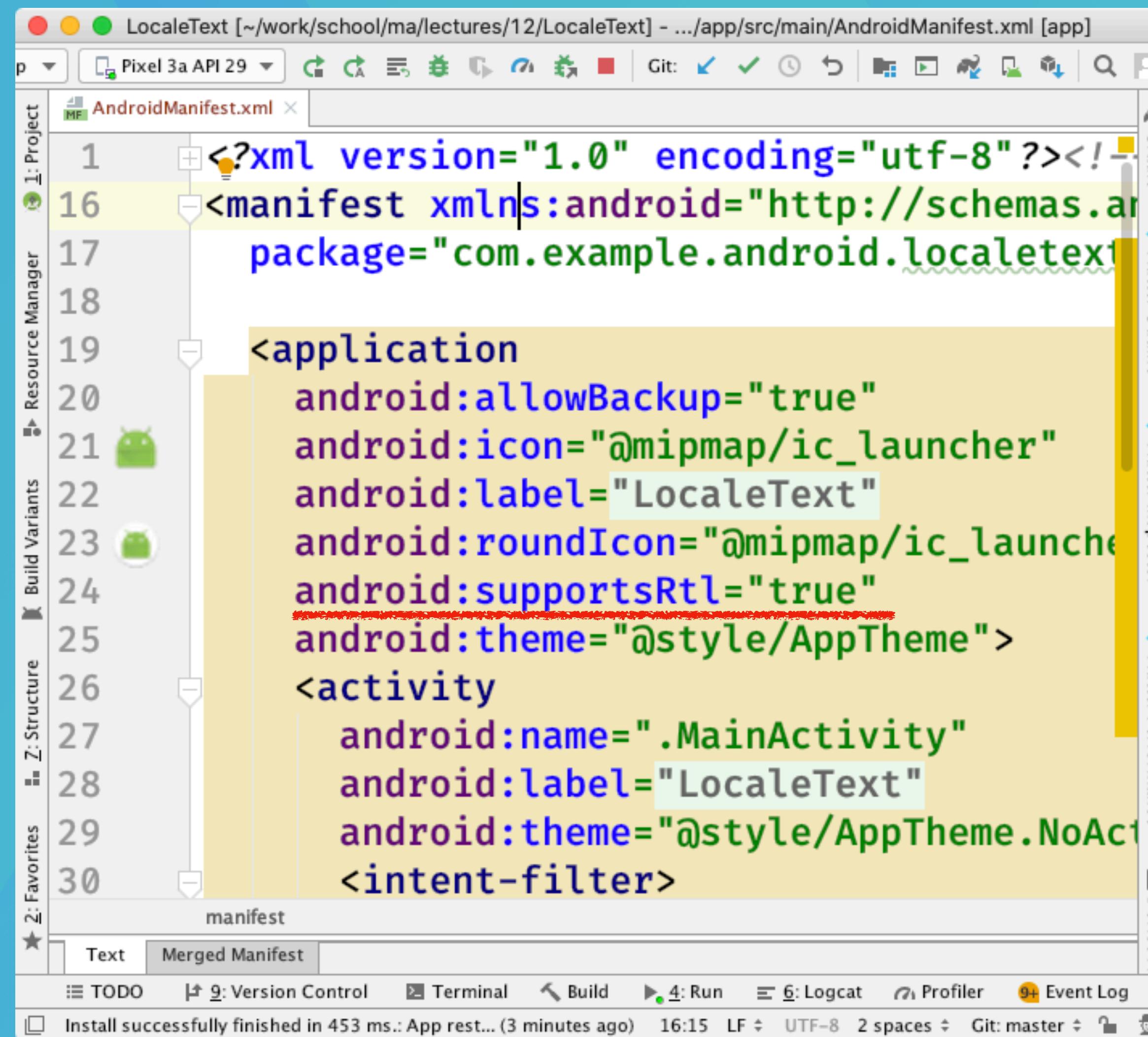
Add another language resource to the app



Run the app and switch languages



Add a right-to-left (RTL) language



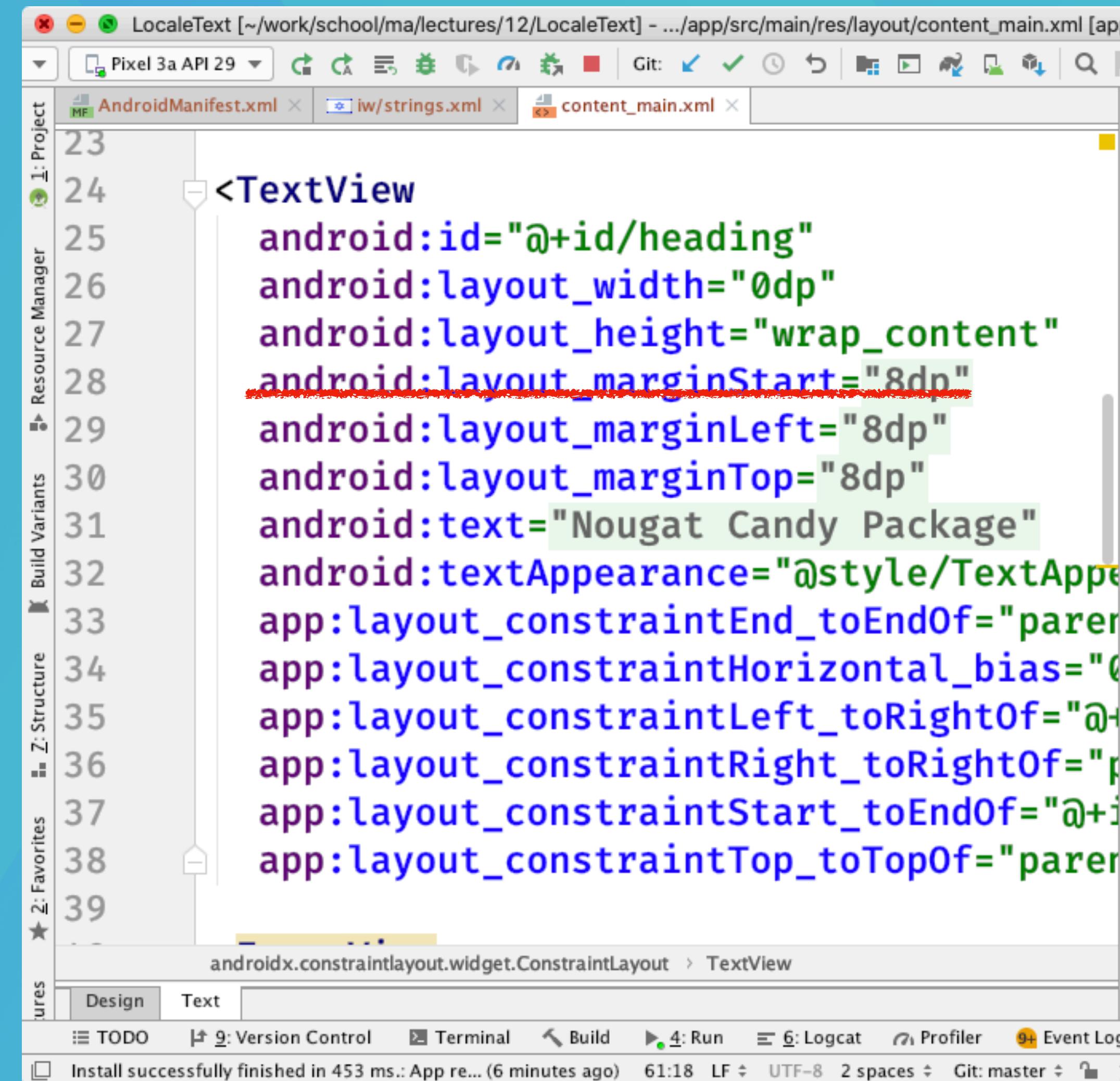
The screenshot shows the AndroidManifest.xml file in the Android Studio editor. The XML code is as follows:

```
<?xml version="1.0" encoding="utf-8"?><!DOCTYPE manifest PUBLIC "-//android//DTD Manifest File 1.0//EN" "http://schemas.android.com/manifest/dtd/manifest_1_0.dtd"><manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.android.localetext" supportsRtl="true">    <application        android:allowBackup="true"        android:icon="@mipmap/ic_launcher"        android:label="LocaleText"        android:roundIcon="@mipmap/ic_launcher_round"        android:theme="@style/AppTheme">        <activity            android:name=".MainActivity"            android:label="LocaleText"            android:theme="@style/AppTheme.NoActionBar">            <intent-filter>
```

The line `android:supportsRtl="true"` is highlighted with a red underline, indicating an error or warning.

DEMO

Add a right-to-left (RTL) language



```
23<TextView  
24    android:id="@+id/heading"  
25    android:layout_width="0dp"  
26    android:layout_height="wrap_content"  
27    android:layout_marginStart="8dp"  
28    android:layout_marginLeft="8dp"  
29    android:layout_marginTop="8dp"  
30    android:text="Nougat Candy Package"  
31    android:textAppearance="@style/TextAppe  
32    app:layout_constraintEnd_toEndOf="parer  
33    app:layout_constraintHorizontal_bias="0  
34    app:layout_constraintLeft_toRightOf="@+  
35    app:layout_constraintRight_toRightOf="P  
36    app:layout_constraintStart_toEndOf="@+i  
37    app:layout_constraintTop_toTopOf="parer  
38    ...  
39
```

The screenshot shows the Android Studio interface with the XML file `content_main.xml` open. The code defines a `TextView` with various attributes. The line `android:layout_marginStart="8dp"` is underlined in red, indicating a potential issue or a point of interest. The XML code is color-coded for syntax highlighting.

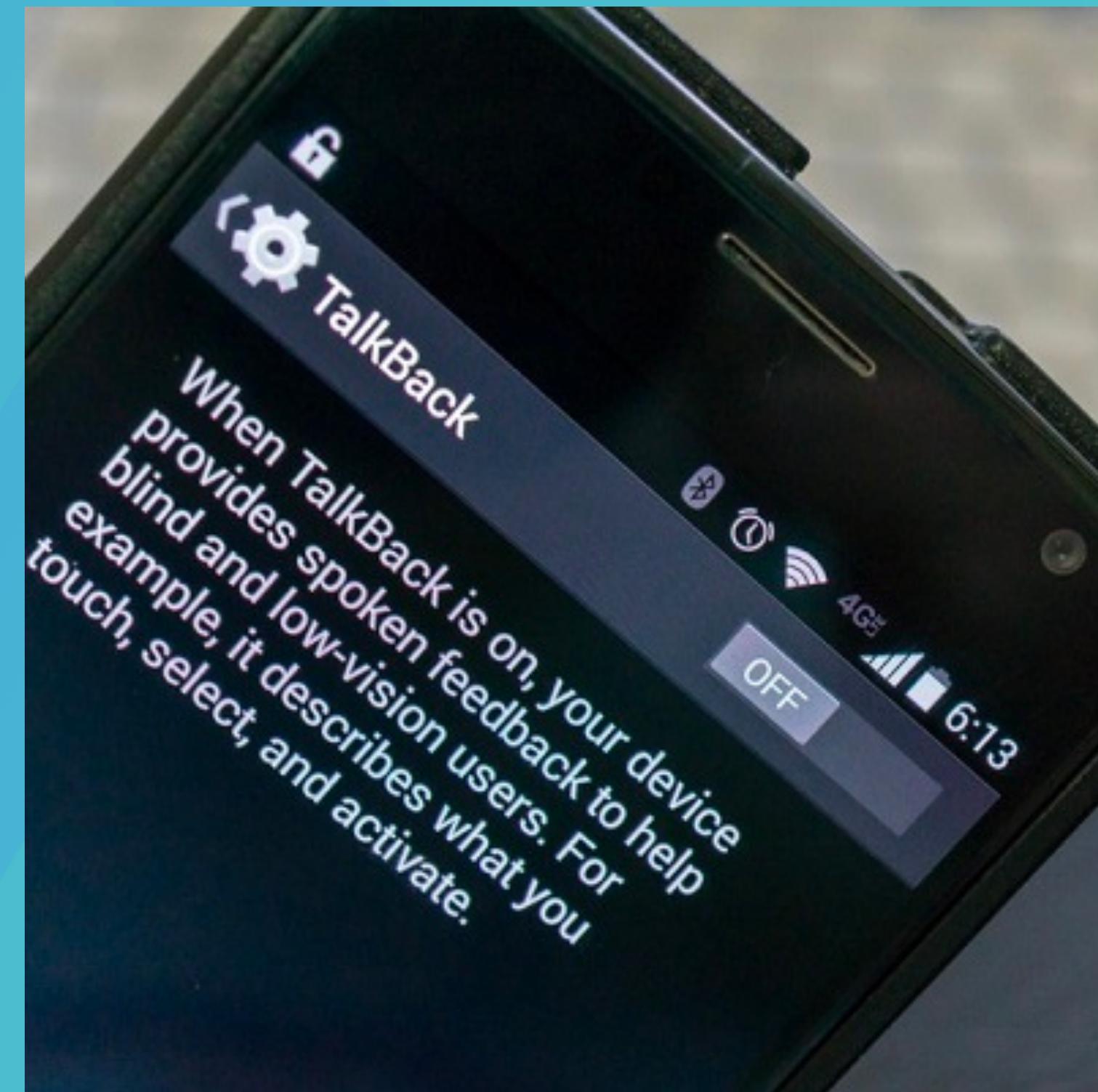
Accessibility

- Blindness.
- Low vision.
- Color blindness.
- Deafness or hearing loss.
- Restricted motor skills.



TalkBack

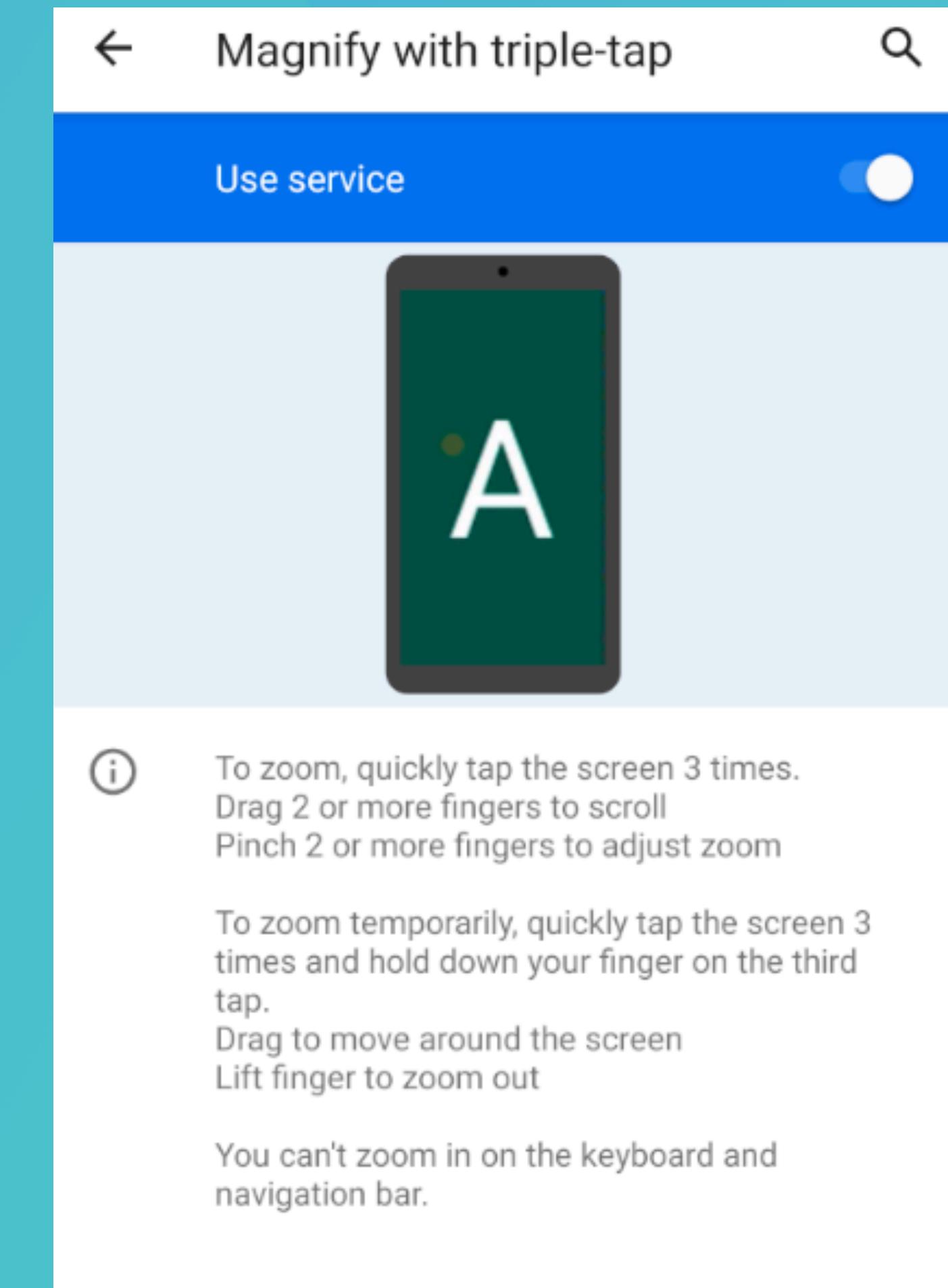
- Settings > Accessibility > TalkBack
- Settings > Accessibility > TalkBack > Settings > Launch TalkBack tutorial.



DEMO

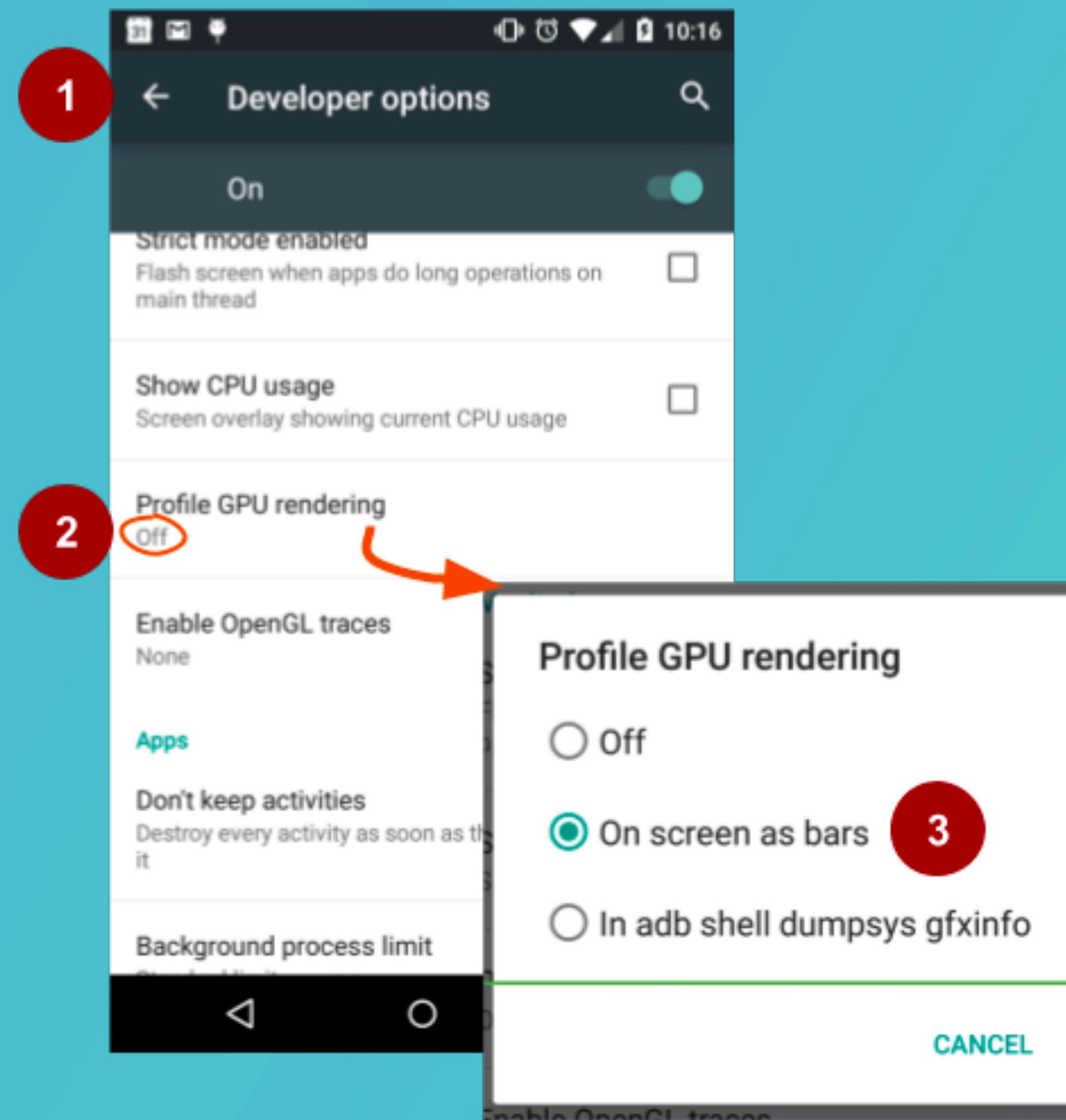
Font and Color

- Settings > Accessibility > Magnification gesture.
- Settings > Accessibility > Font size.
- Settings > Accessibility > High contrast text.



GPU Profiling

- Settings
 - Developer options
 - Monitoring
 - Profile GPU rendering

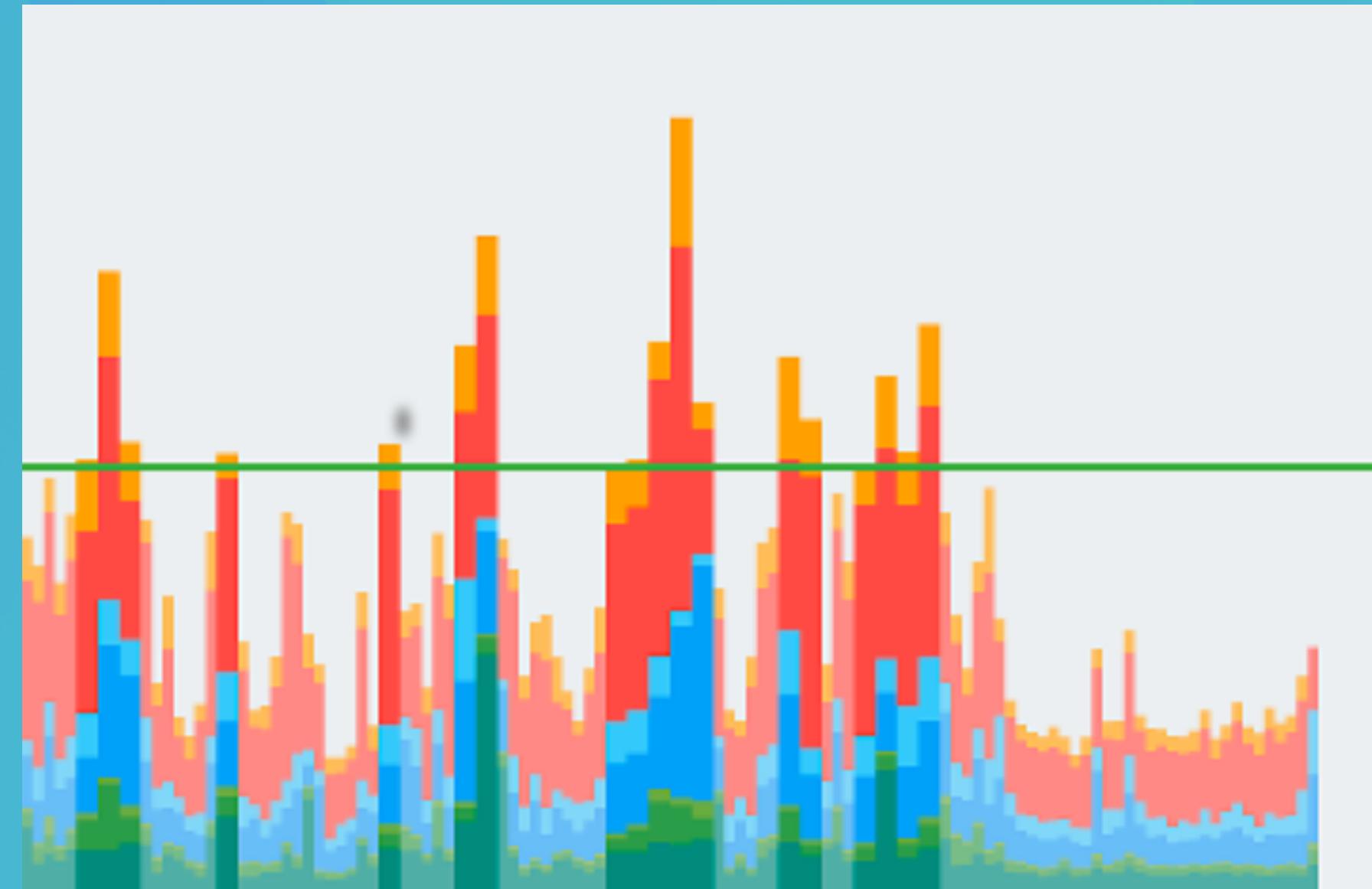


twitch.tv/dancojocar

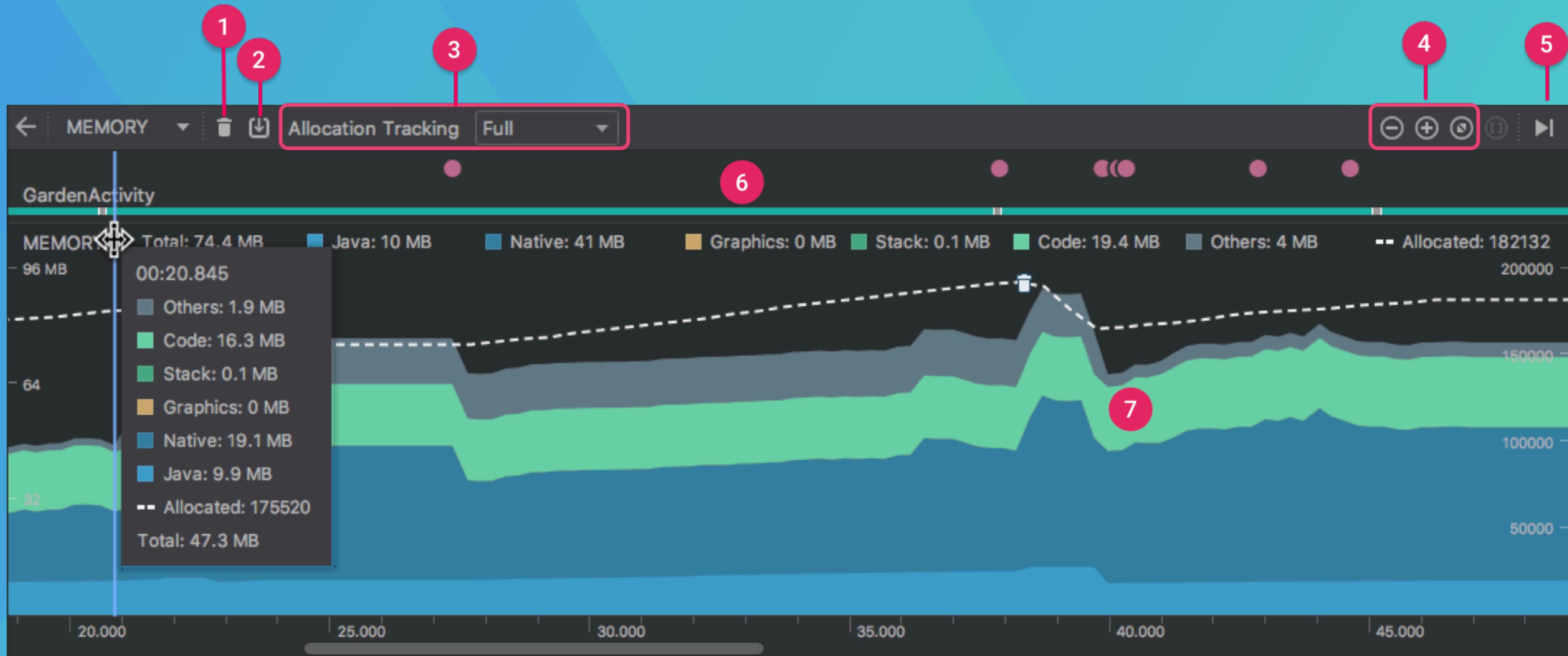
youtube.com/dancojocar

DEMO

GPU Profiling



Misc Input Anim. Measure Draw Upload Issue Swap

DEMO

ViewBinding

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/mainTitle"
        tools:text="Main Title" />

    <TextView
        android:id="@+id/subTitle"
        tools:text="Main Subtitle" />
</RelativeLayout>
```

ViewBinding

```
public class MainActivity extends AppCompatActivity {  
    private TextView txtViewMainTitle;  
    private TextView txtViewSubTitle;  
  
    @Override  
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        txtViewMainTitle = findViewById(R.id.mainTitle);  
        txtViewSubTitle = findViewById(R.id.subTitle);  
  
        txtViewMainTitle.setText("This is my main title");  
        txtViewSubTitle.setText("This is my subTitle");  
    }  
}
```

ViewBinding

Android Open Source Project CHANGES ▾ DOCUMENTATION ▾ BROWSE ▾

Merged as [637b173](#) | [882241](#): Sample updates: Fragment state, synth accessors

Updated	Jan 30, 2019
Owner	Jakub Gielzak
Assignee	
Reviewers	Treehugger Robot Florina Muntenescu Jelle Fresen
CC	Nikita Frukt
Repo	platform/frameworks/support
Branch	androidx-master-dev
Parent	2a9664f
Topic	No topic
Hashtags	

Sample updates: Fragment state, synth accessors

1) Moved click count to Fragment state
This verifies / highlights FragmentStateAdapter's ability to correctly handle Fragment state.

2) Replaced kotlinx synthetic with findViewById
kotlinx.android.synthetic is no longer a recommended practice. Removing in favour of explicit findViewById.

Bug: [122659289](#)
Test: manual

Change-Id: [Ic472f90e28f7133822edcf53f44b83dc333f768e](#)

Code-Review +1 Florina Muntenescu
+2 Jelle Fresen

ViewBinding

The Argument Over Kotlin Synthetics

- They Are Kotlin Only.
- They Don't Expose Nullability.
- The Code Generated Is Not Guaranteed To Be Performant.
- Everything Exists In A Global Namespace.
- Typing Isn't Guaranteed.

ViewBinding

Android Studio 3.6

```
app/build.gradle
  android {
    ...
    viewBinding {
      enabled = true
    }
  }
```



ViewBinding

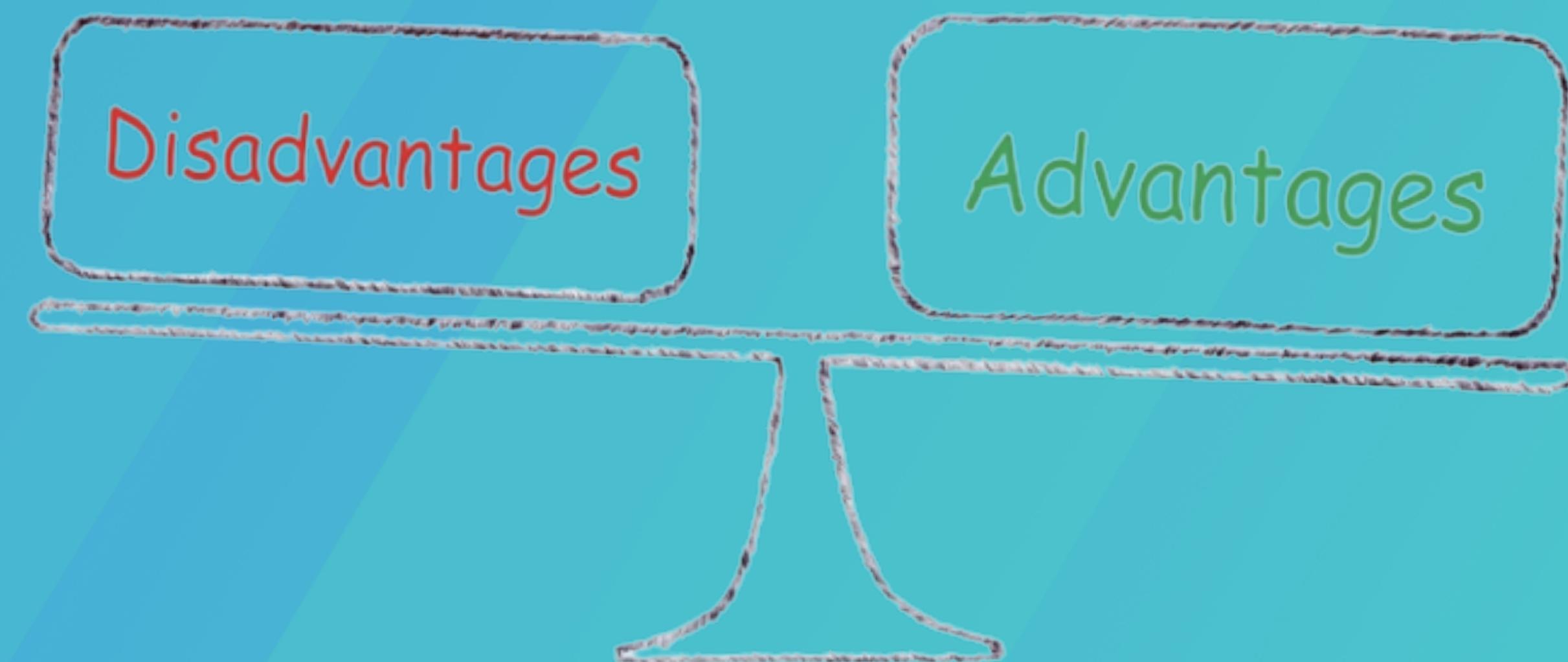
build.gradle:

```
buildscript {  
    ext.kotlin_version = '1.4.10'  
    repositories {  
        google()  
        jcenter()  
  
    }  
    dependencies {  
        classpath "com.android.tools.build:gradle:4.1.1"  
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
    }  
}
```

DEMO

ViewBinding

- The auto-generated .class increases app size.
- Harder to debug.
- Null safety.
- Type safety.
- Speed.



Lecture outcomes

- Identify app widgets, and understand the key parts of an app widget.
- Implement app widget actions when an element of an app widget is tapped.
- Add support for different languages.
- Test your app for accessibility in a variety of ways.
- Use the Profile GPU Rendering tool to visualize Android drawing the screen.
- Use Memory Profiler to collect data about your app.

