

# **Lecture #12**

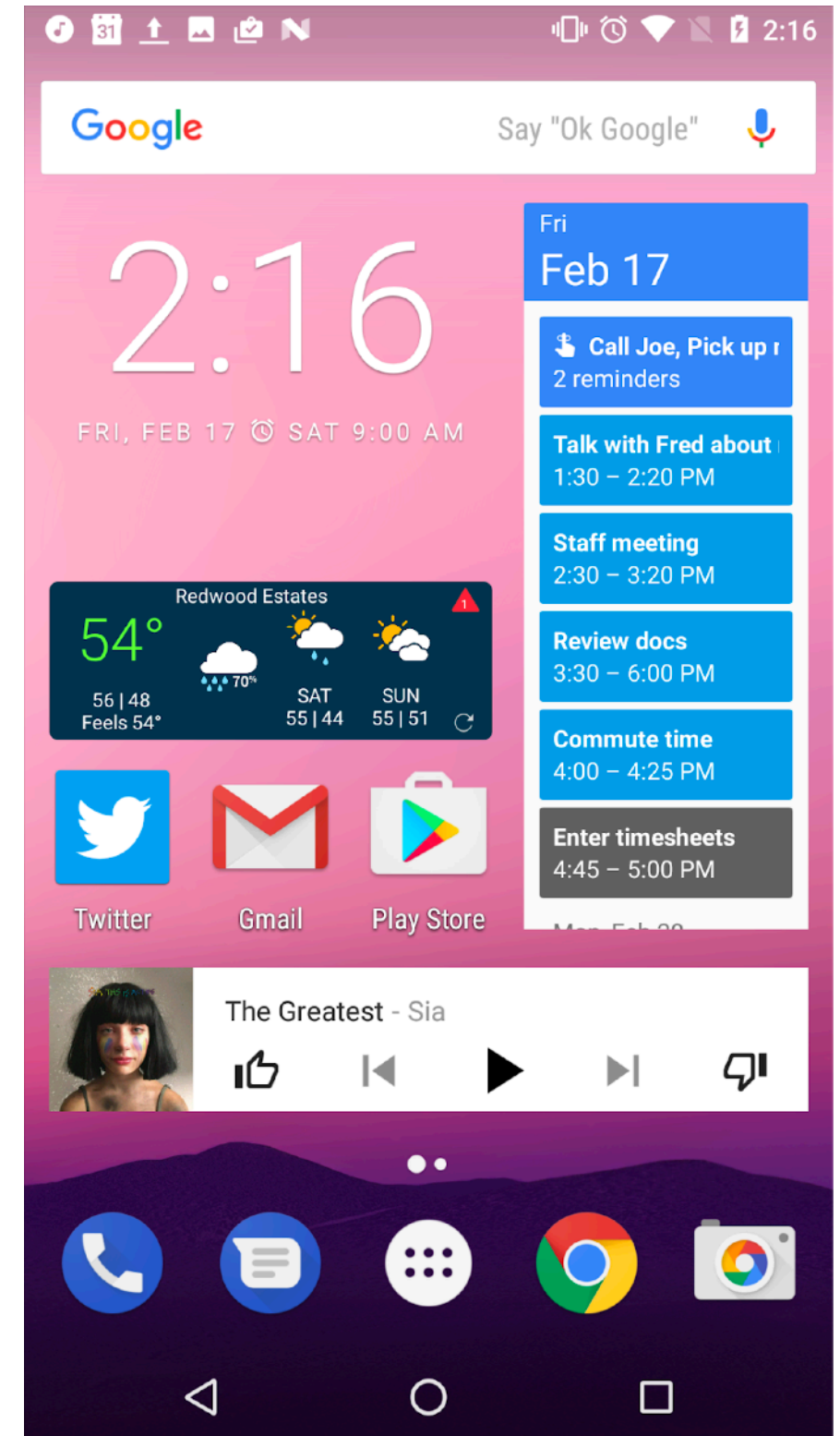
# **Advanced Mobile**

# **Development**

Mobile Applications 2019-2020

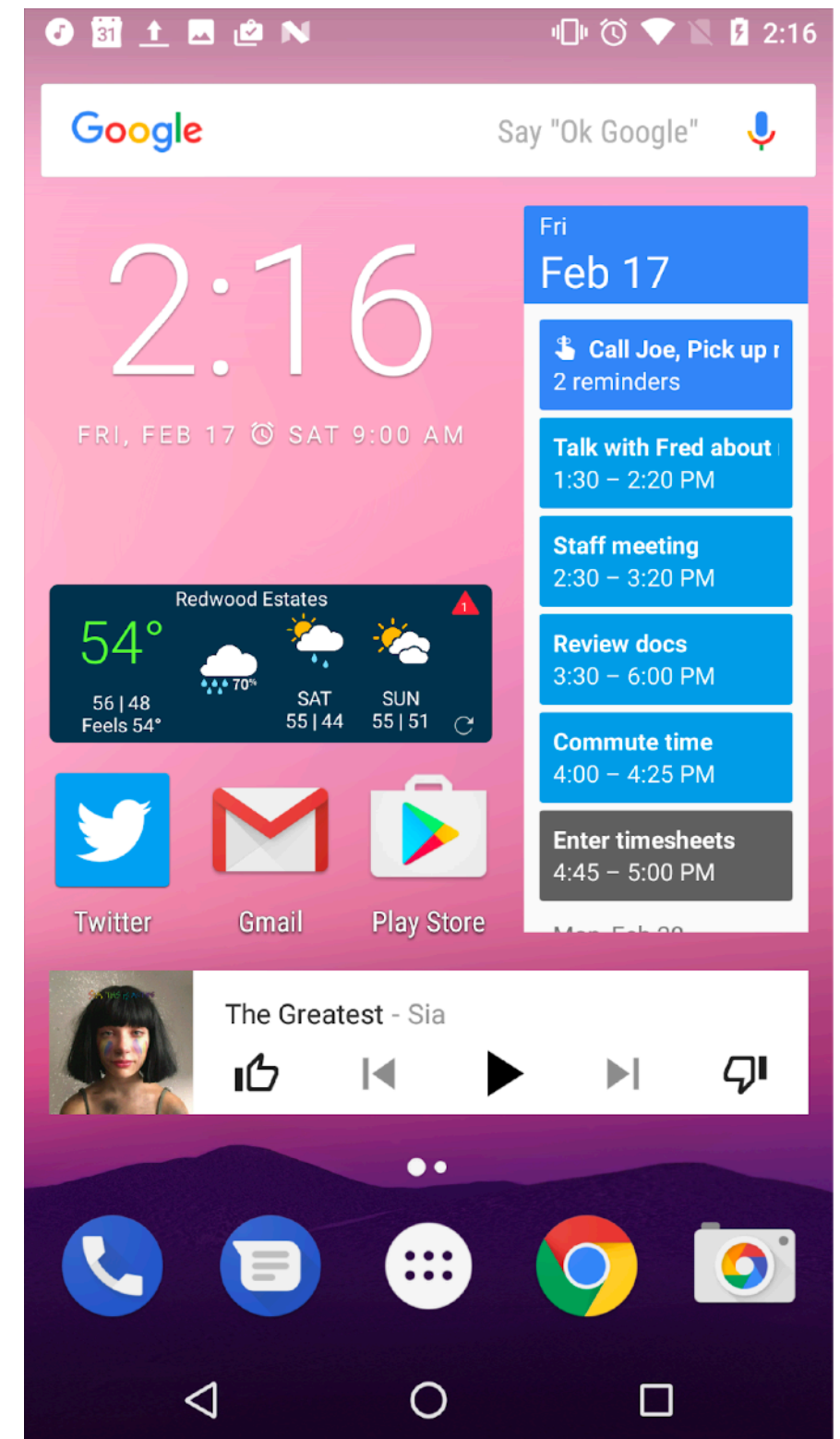
# App Widgets

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



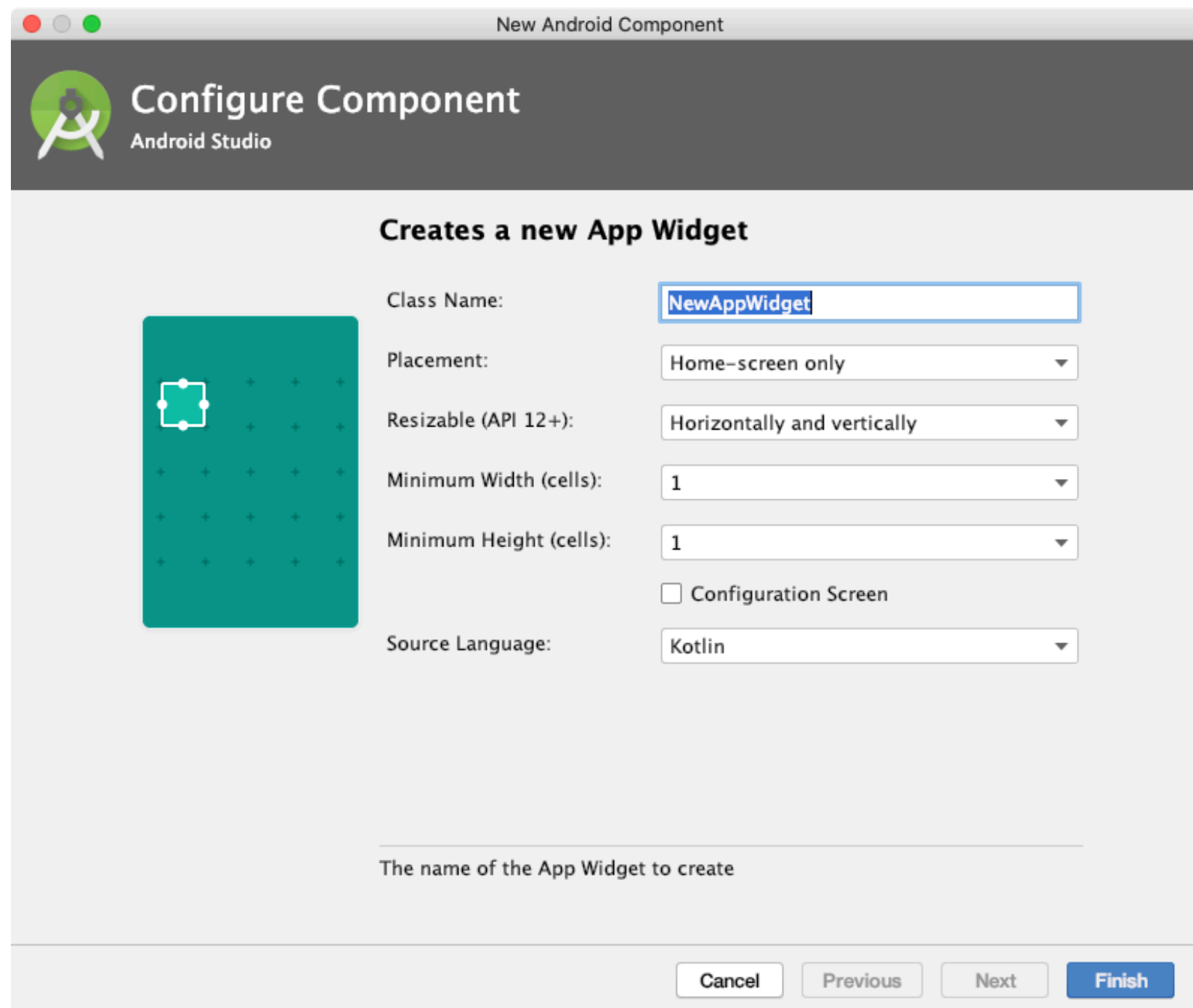
# App Widgets

- 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.

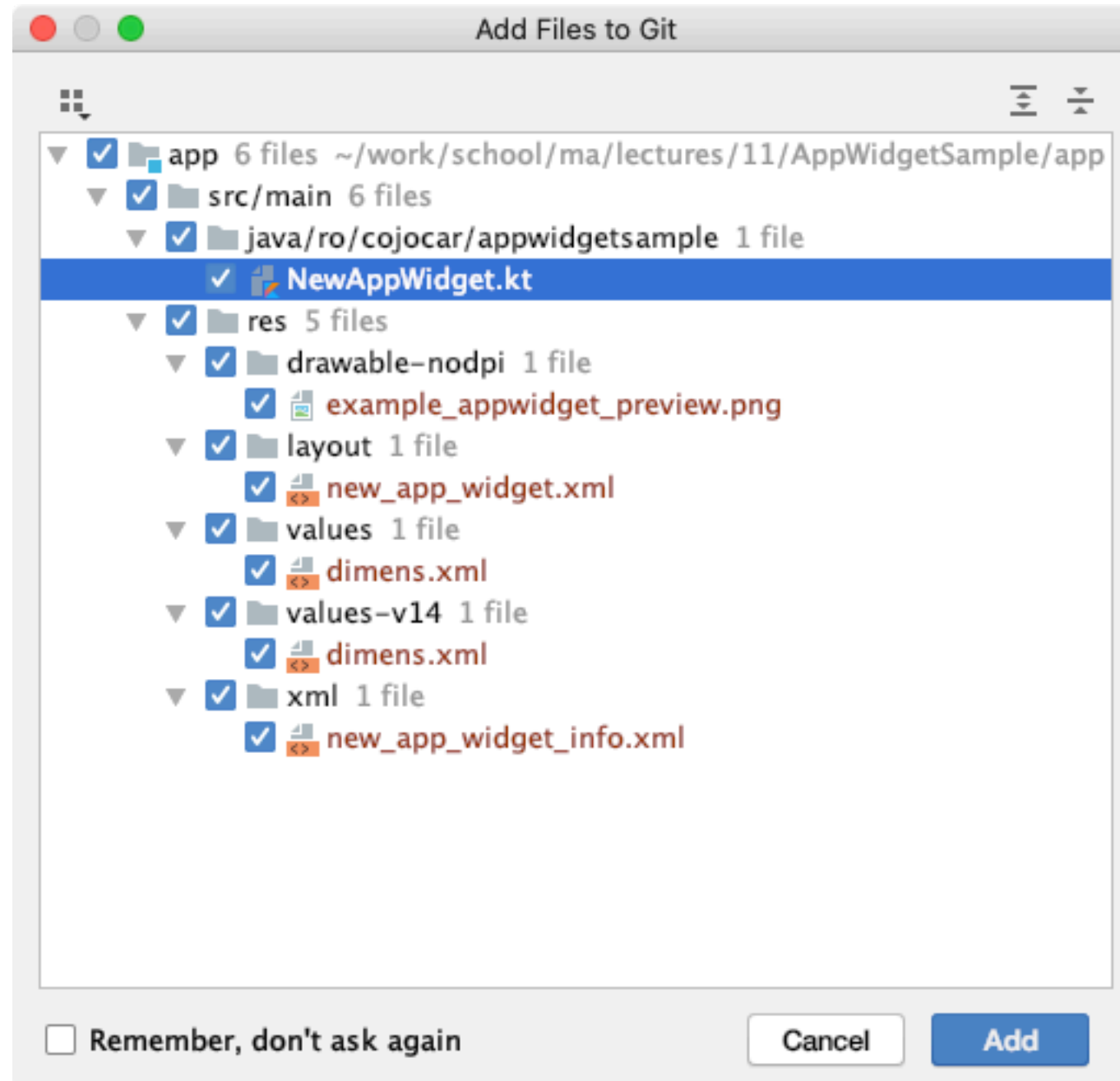


The screenshot shows the 'New Android Component' dialog in Android Studio. The title bar reads 'New Android Component'. The main header area contains the Android Studio logo and the text 'Configure Component' and 'Android Studio'. Below this, the section is titled 'Creates a new App Widget'. On the left, there is a visual representation of a widget: a teal square with a white grid of dots, and a smaller teal square with a white border and dots, indicating it is a widget. To the right of this visual are several configuration options:

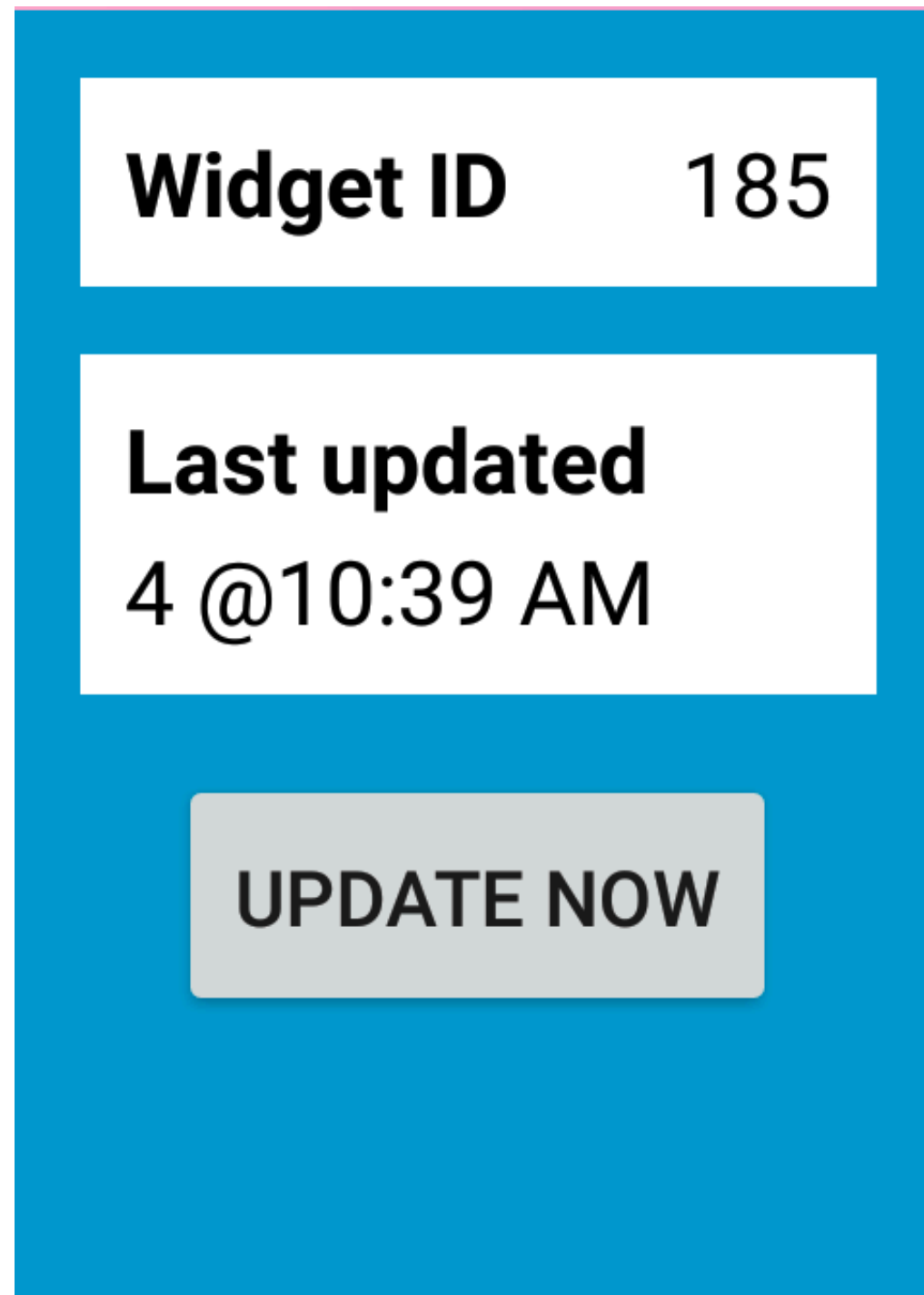
- Class Name:** A text field containing 'NewAppWidget'.
- Placement:** A dropdown menu set to 'Home-screen only'.
- Resizable (API 12+):** A dropdown menu set to 'Horizontally and vertically'.
- Minimum Width (cells):** A dropdown menu set to '1'.
- Minimum Height (cells):** A dropdown menu set to '1'.
- Configuration Screen:** An unchecked checkbox.
- Source Language:** A dropdown menu set to 'Kotlin'.

At the bottom of the dialog, there is a line of text: 'The name of the App Widget to create'. At the very bottom, there are four buttons: 'Cancel', 'Previous', 'Next', and 'Finish'.

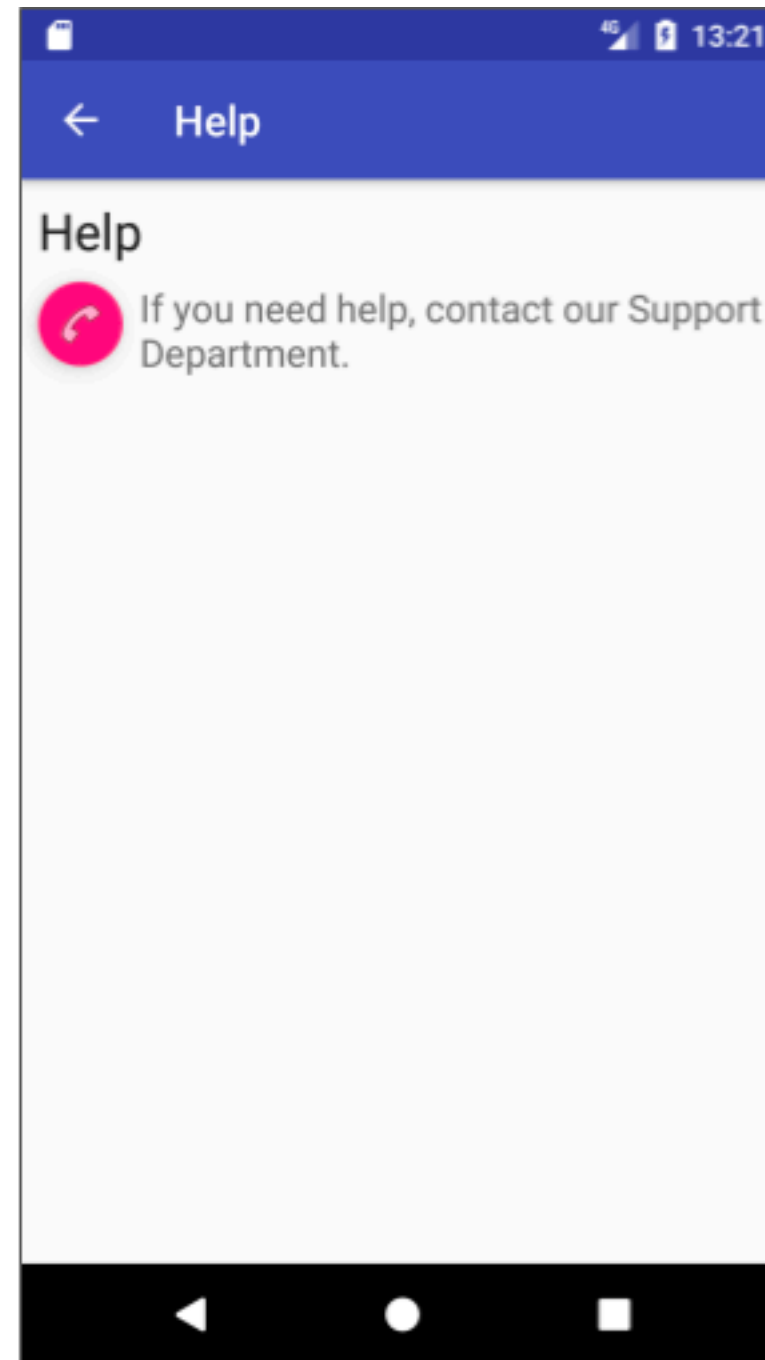
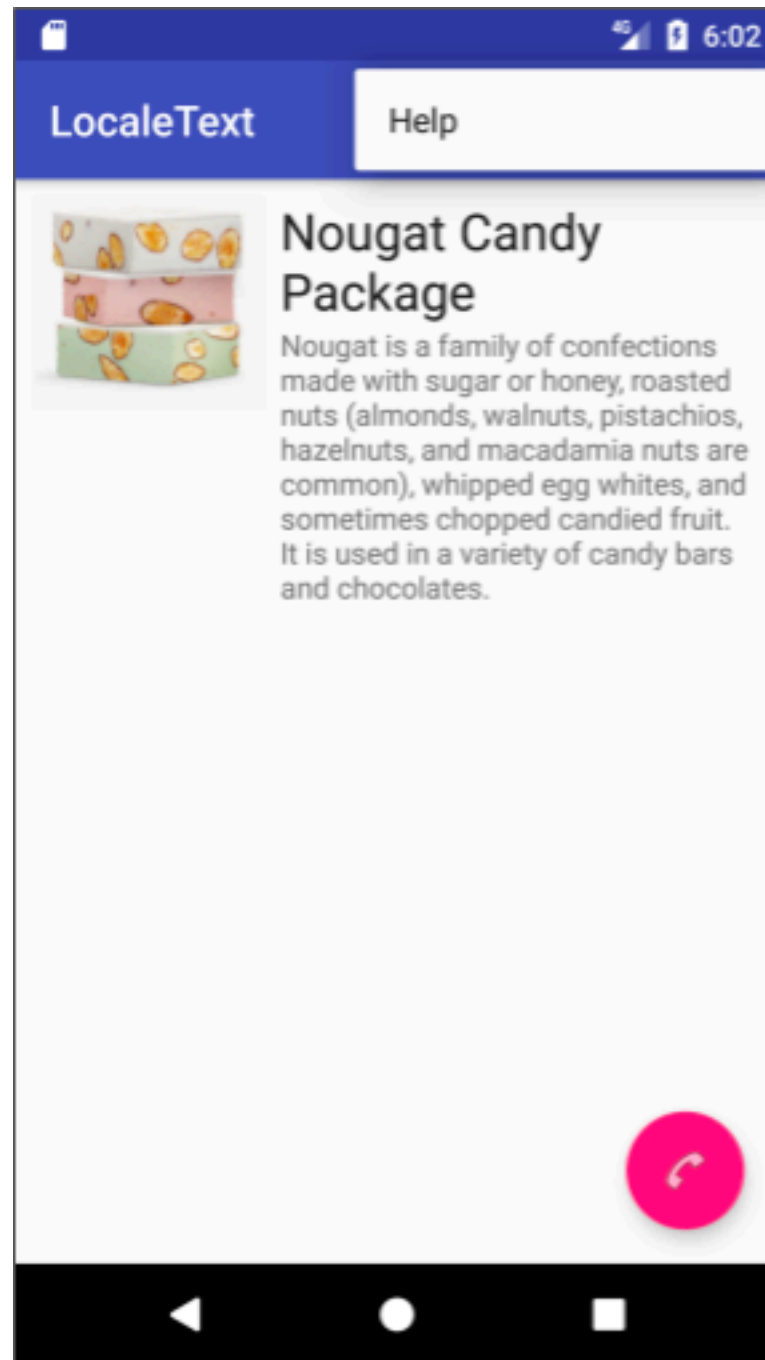
# Generated Files



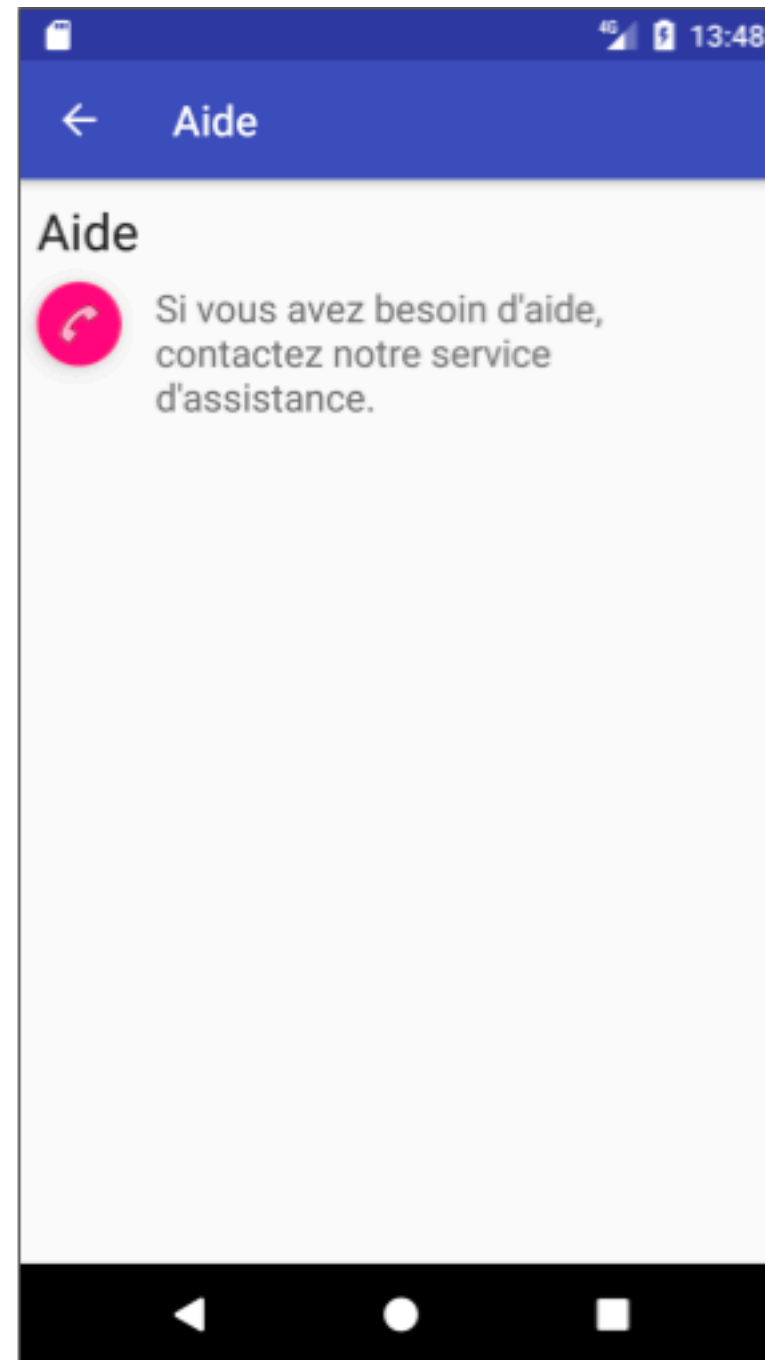
# Customize the widget



# Language Support

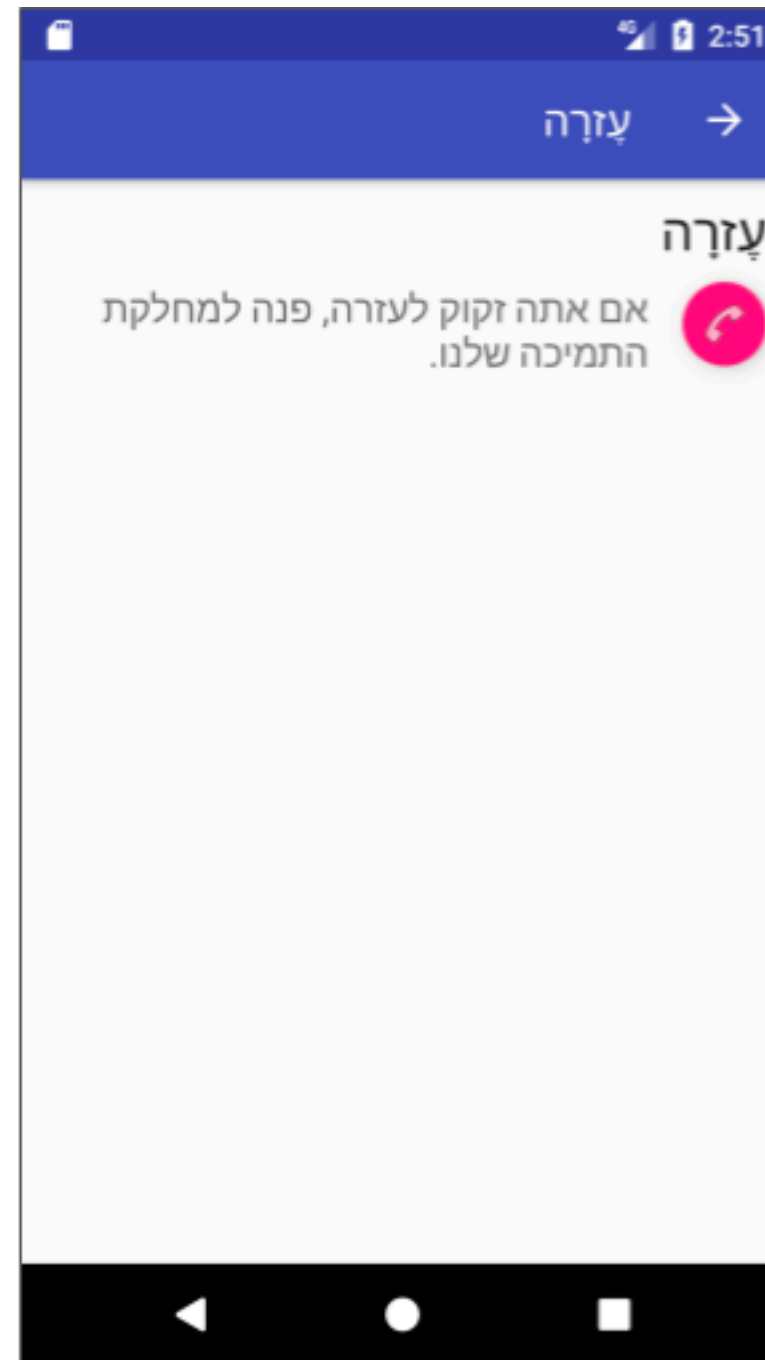


# Language Support

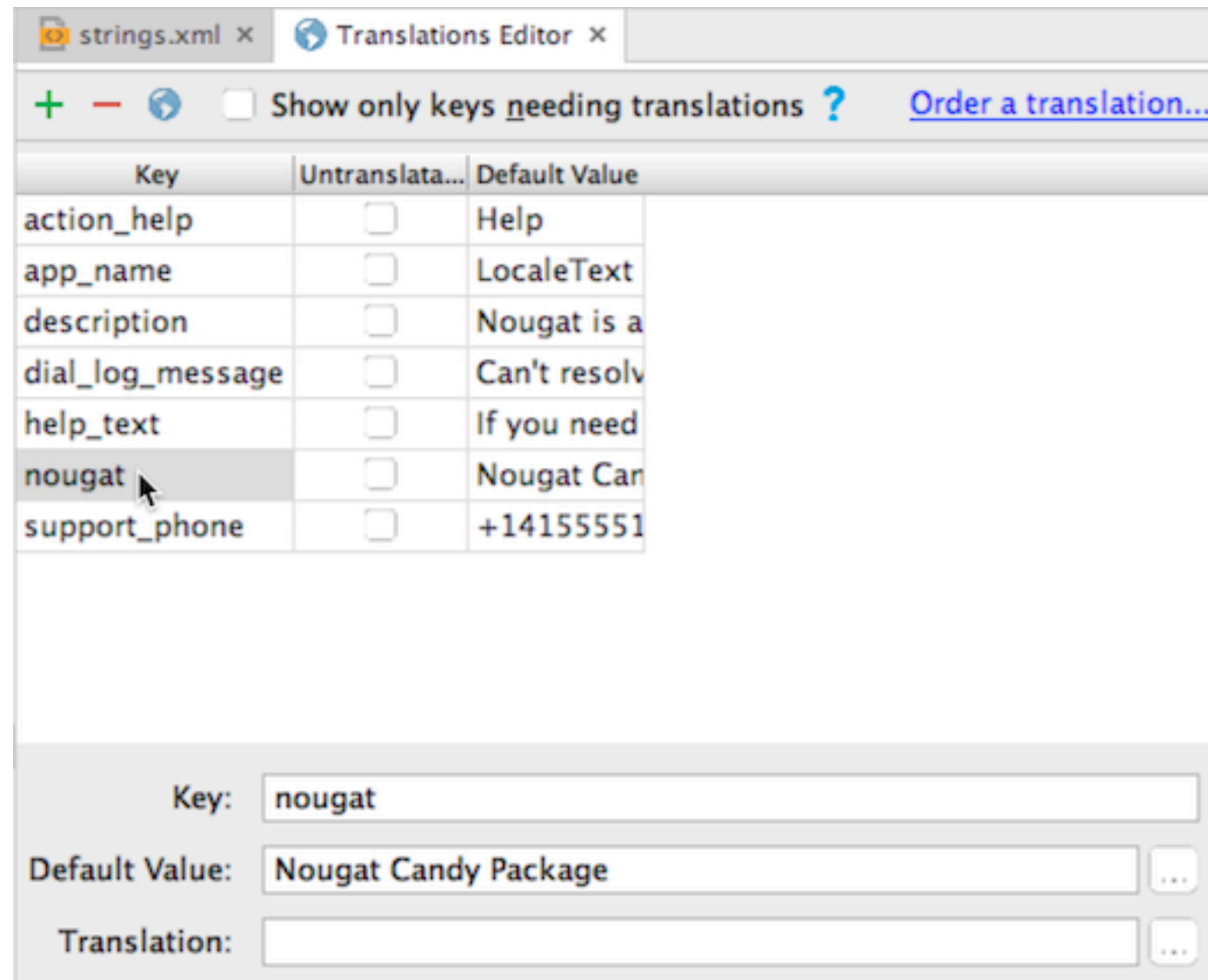




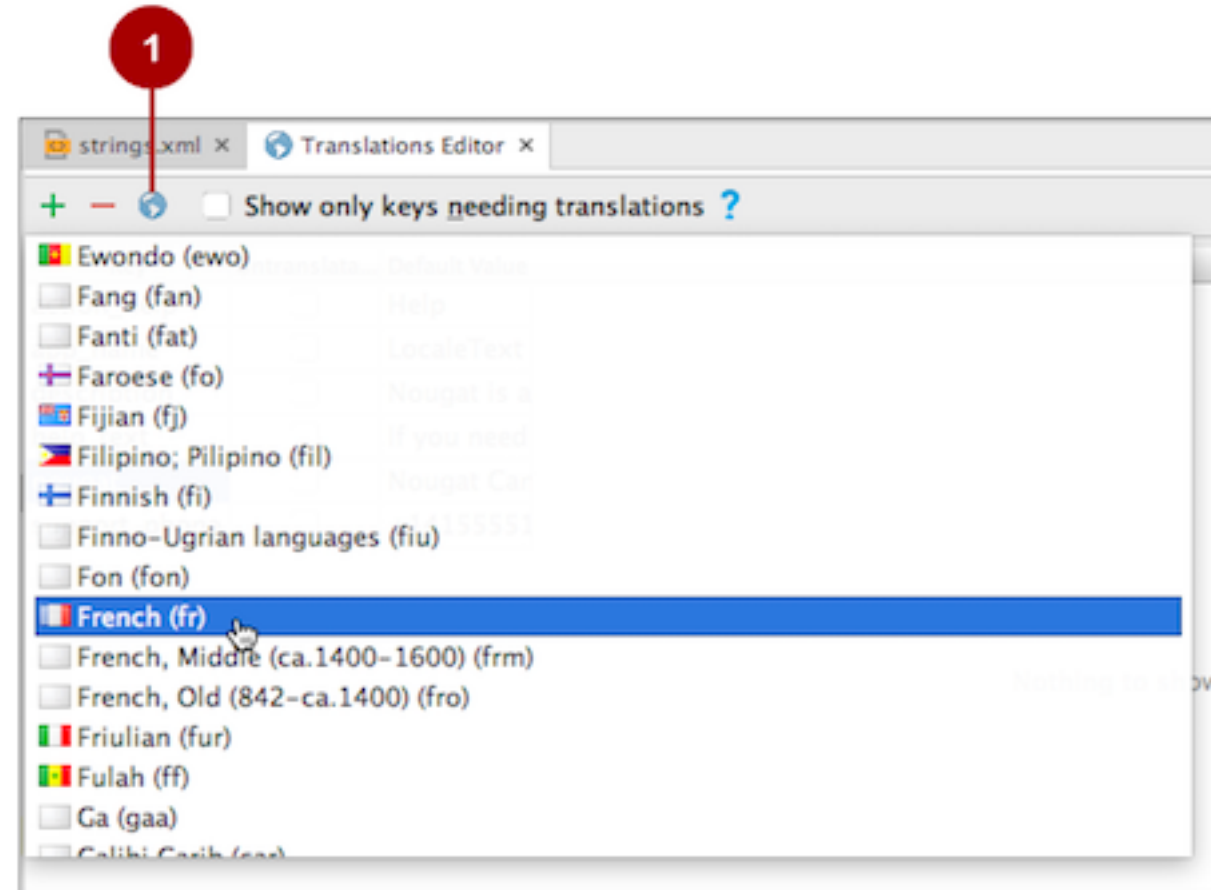
# Language Support



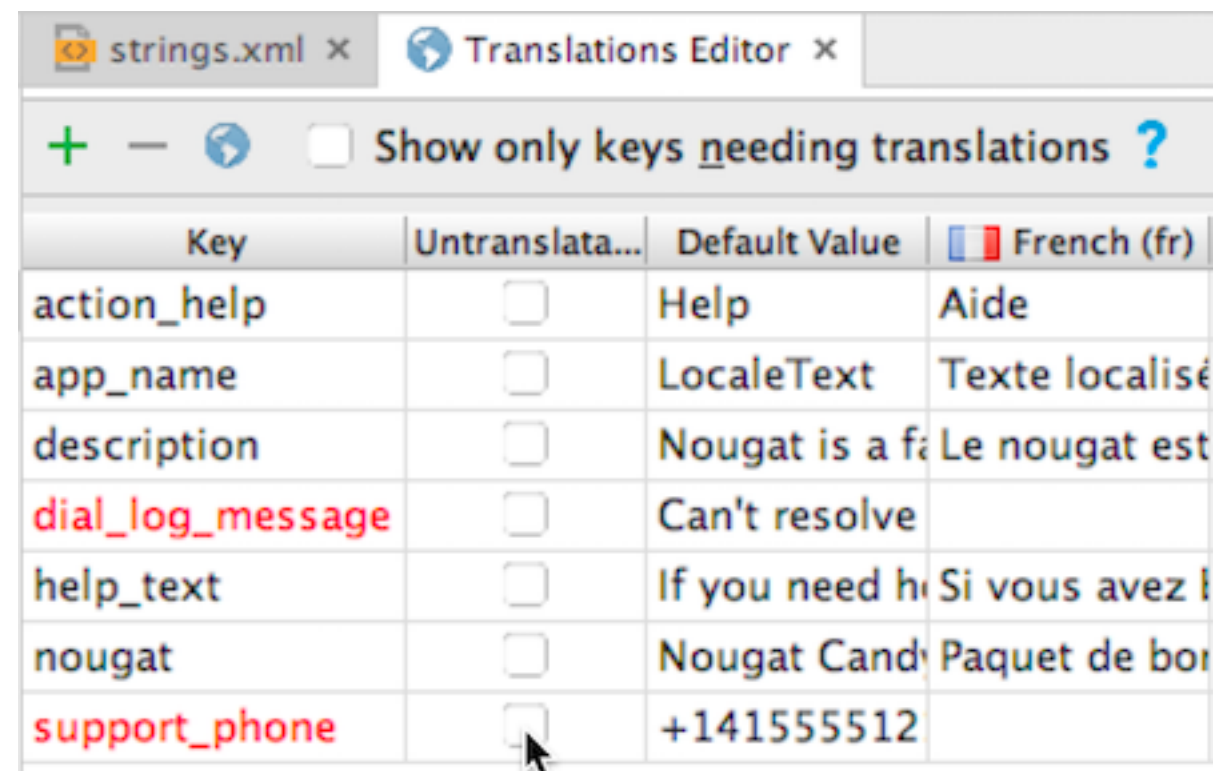
# Add another language resource to the app



# Add another language resource to the app



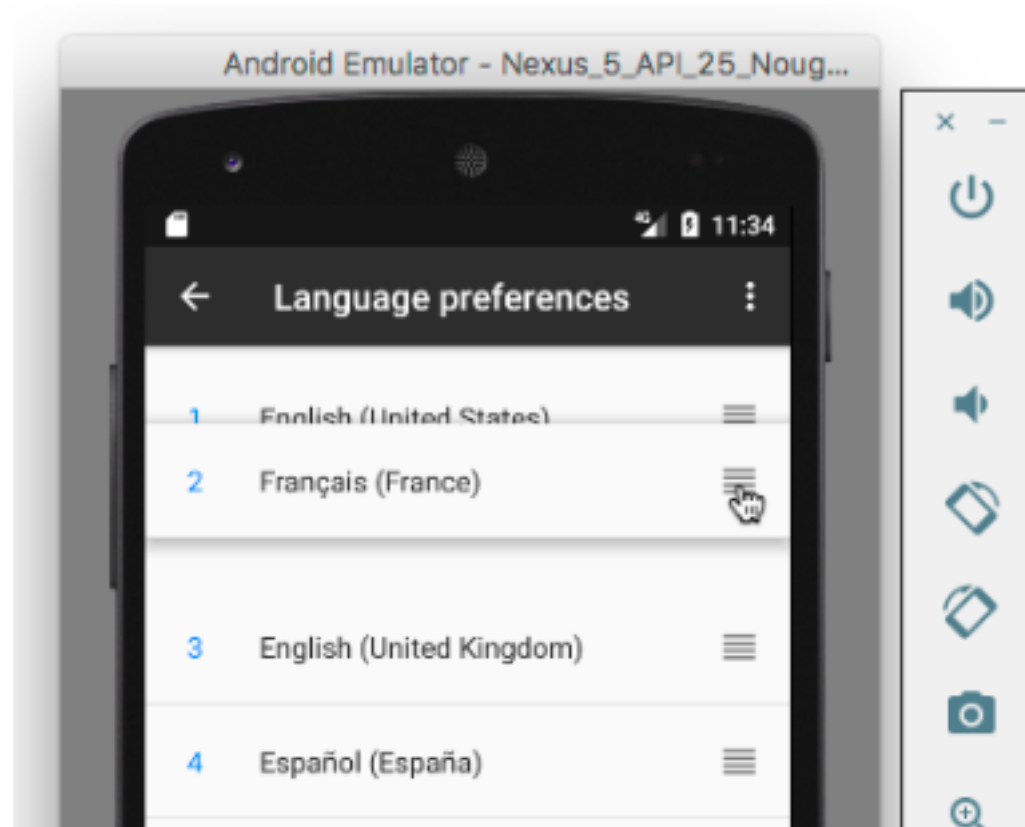
# Add another language resource to the app



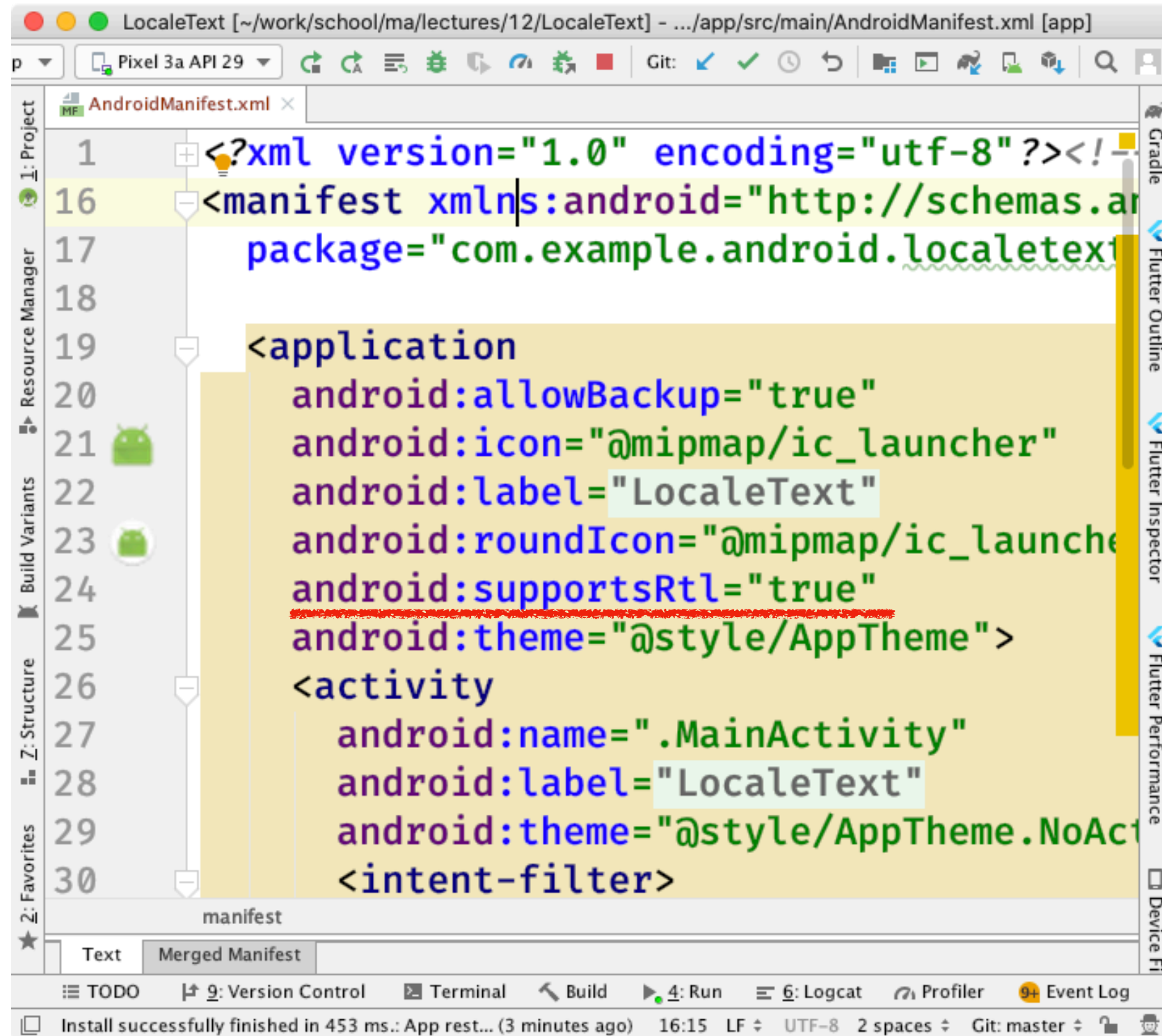
The screenshot shows the 'Translations Editor' window in Android Studio. It displays a table of string resources for the 'strings.xml' file. The table has four columns: 'Key', 'Untranslated', 'Default Value', and 'French (fr)'. The 'Untranslated' column contains checkboxes for each row. The 'Default Value' column contains the original string values. The 'French (fr)' column contains the translated string values. A mouse cursor is hovering over the checkbox in the 'Untranslated' column for the 'support\_phone' key.

Key	Untranslated	Default Value	French (fr)
action_help	<input type="checkbox"/>	Help	Aide
app_name	<input type="checkbox"/>	LocaleText	Texte localisé
description	<input type="checkbox"/>	Nougat is a fa	Le nougat est
dial_log_message	<input type="checkbox"/>	Can't resolve	
help_text	<input type="checkbox"/>	If you need h	Si vous avez l
nougat	<input type="checkbox"/>	Nougat Candy	Paquet de bon
support_phone	<input type="checkbox"/>	+141555512	

# Run the app and switch languages

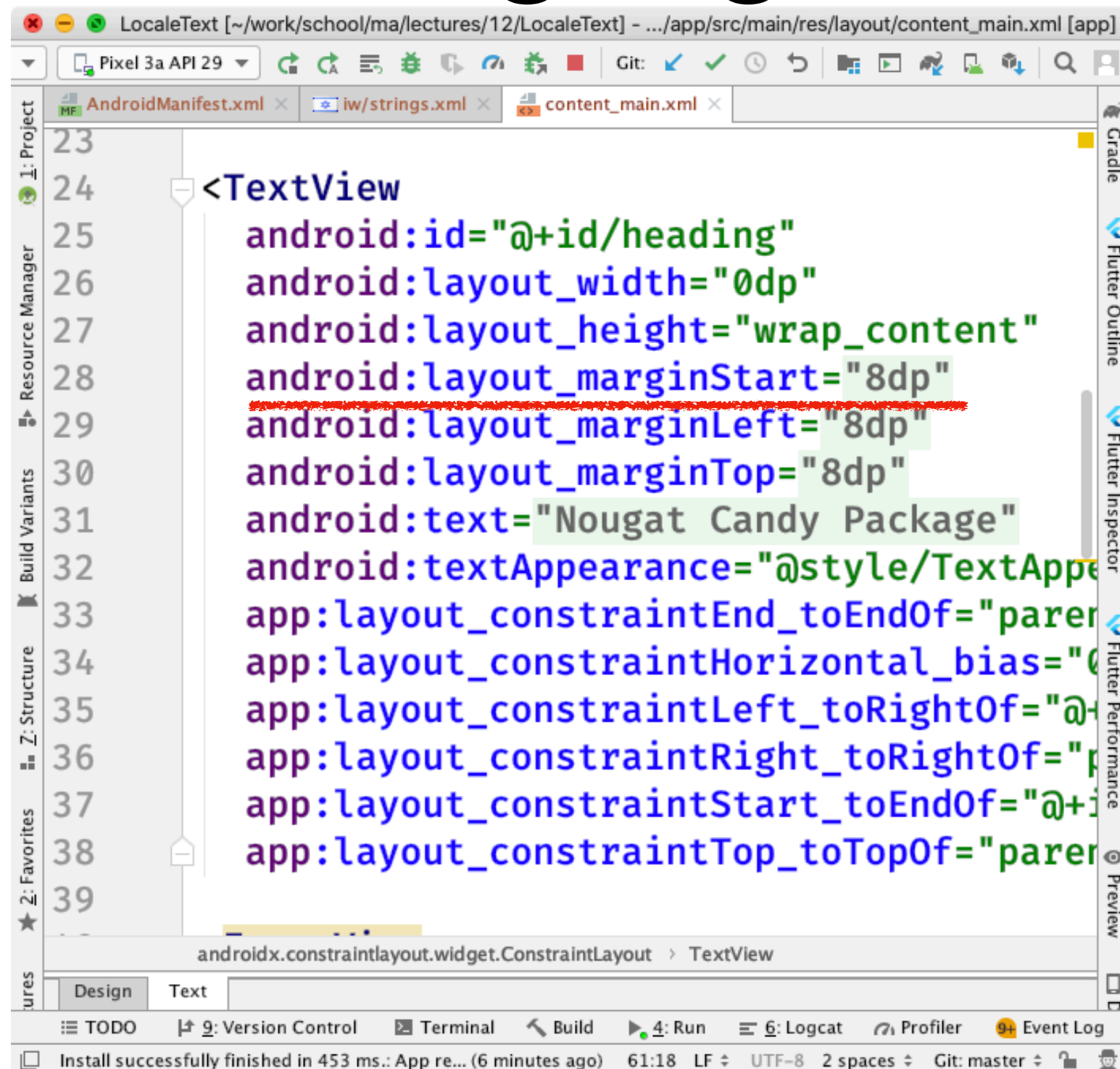


# Add a right-to-left (RTL) language



# Add a right-to-left (RTL) language

DEMO



# Accessibility

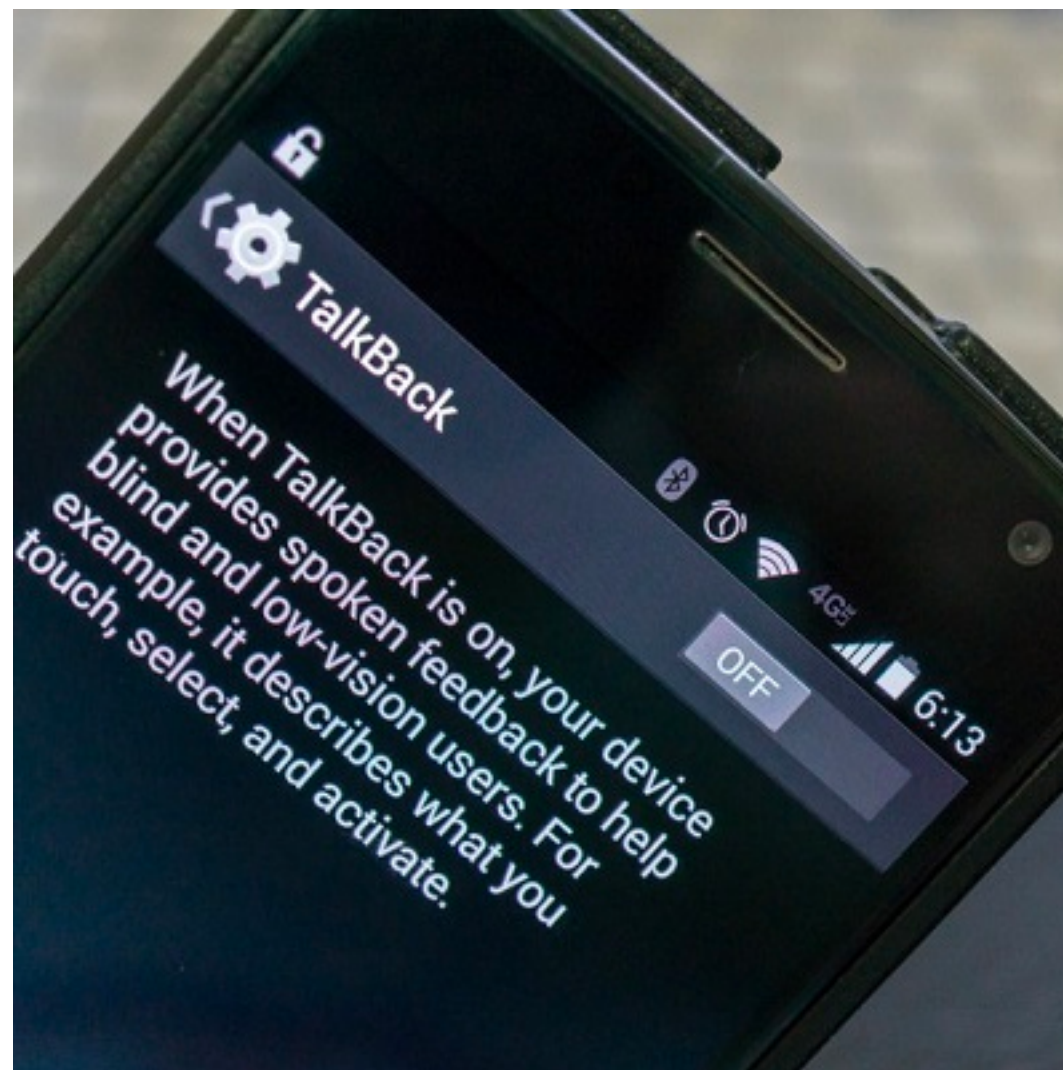
- Blindness
- Low vision.
- Color blindness.
- Deafness or hearing loss.
- Restricted motor skills.





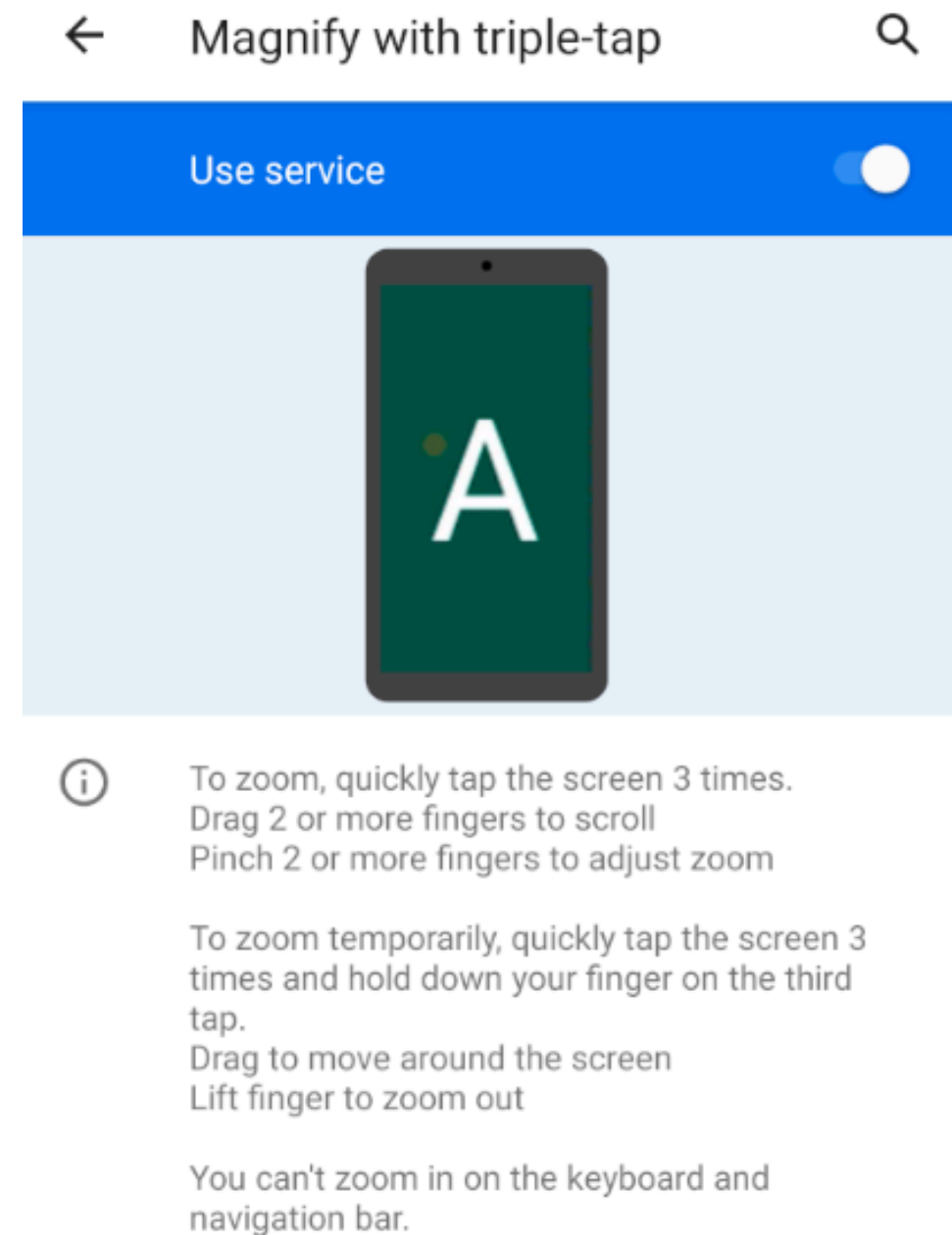
# TalkBack

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



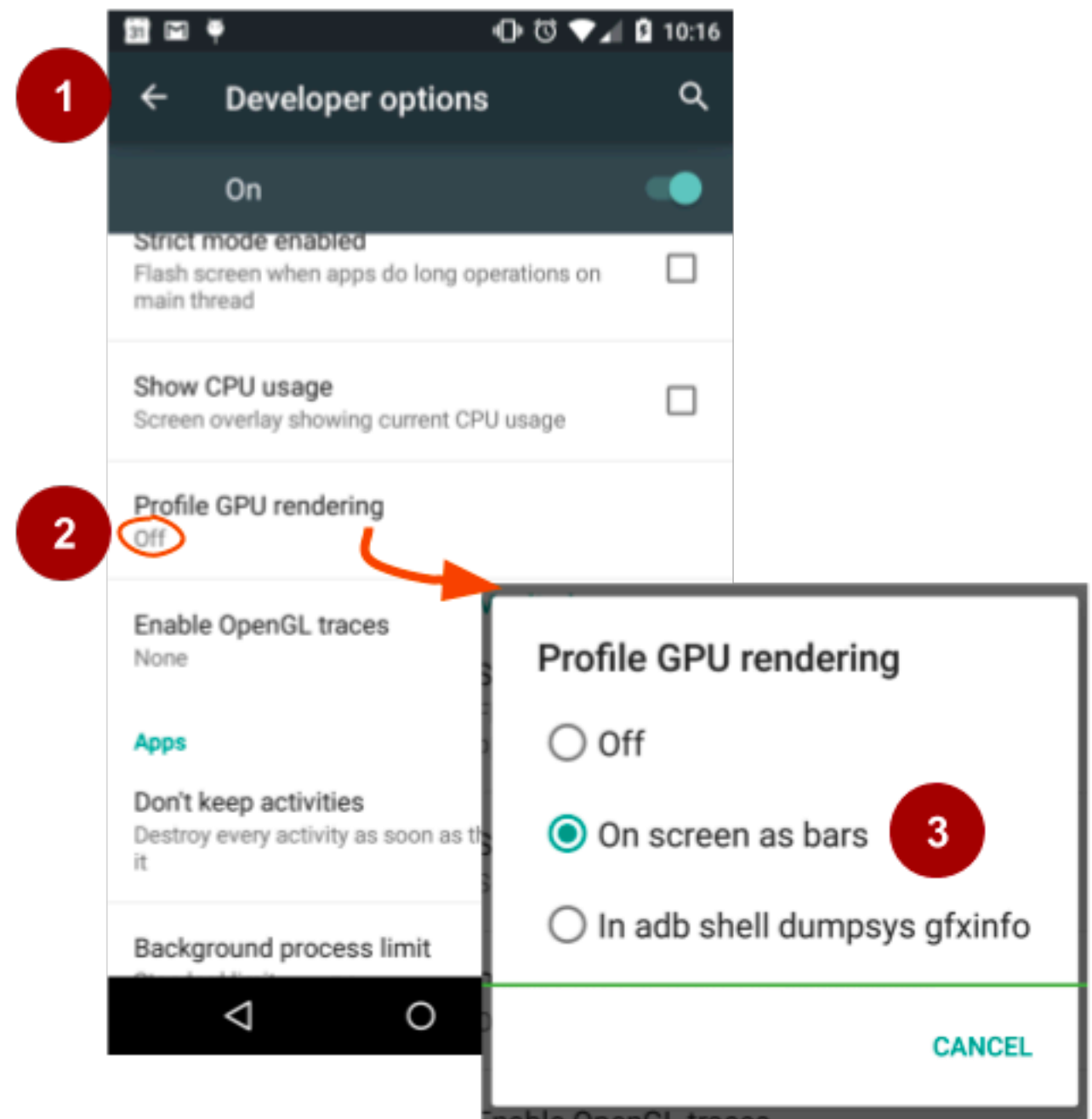
# Font and Color

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



# GPU Profiling

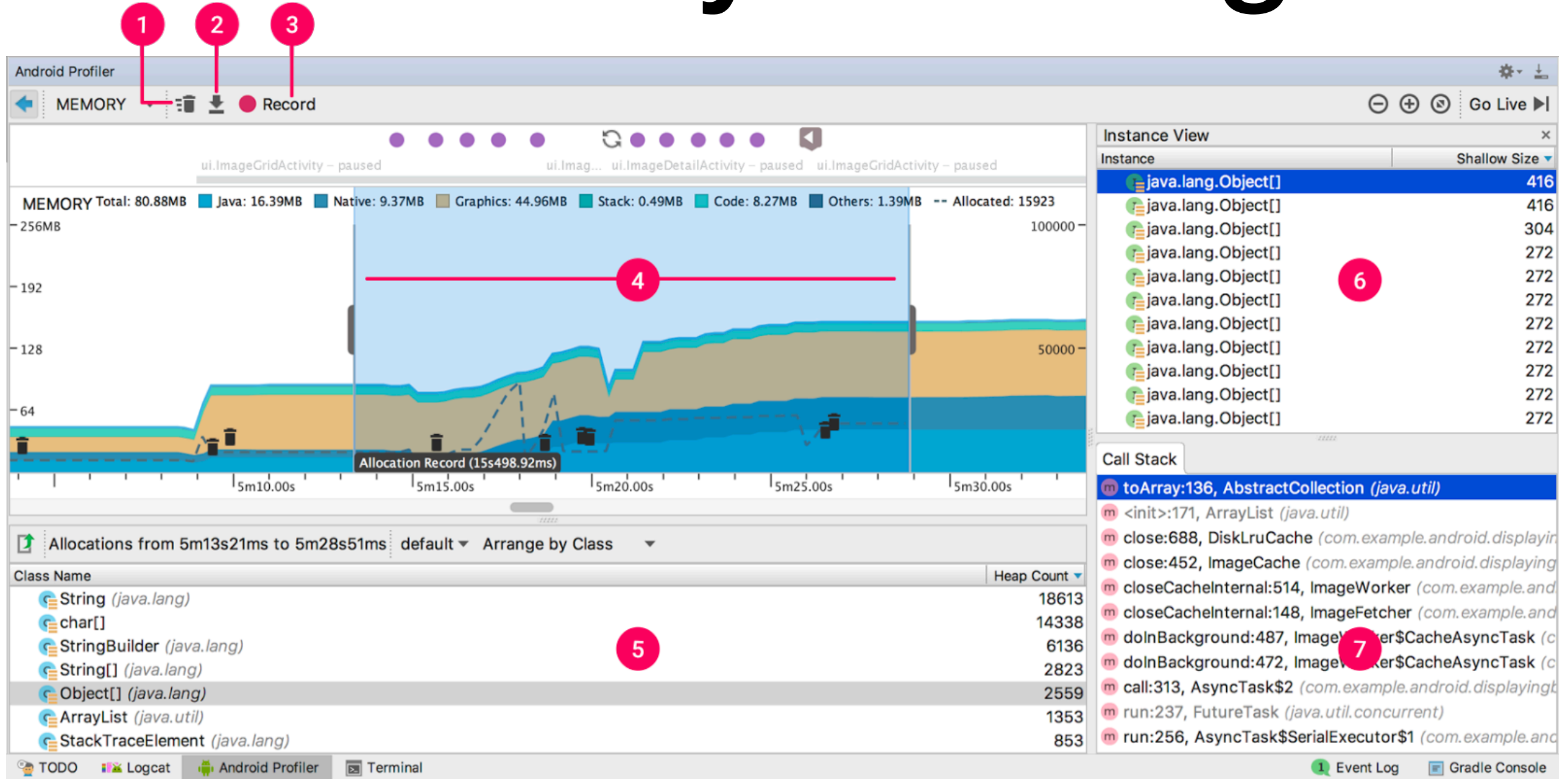
- Settings
  - Developer options
    - Monitoring
      - Profile GPU rendering



# GPU Profiling



# Memory Profiling



# 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

```
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)


        mainTitle.text = "This is my main title"
        subTitle.text = "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 kotlin 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

Android  
Open Source Project

CHANGES ▾DOCUMENTATION ▾BROWSE ▾

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

UpdatedJan 30, 2019

Owner Jakub Gielzak

Assignee

Reviewers Treehugger Robot  
 Florina Muntenescu  
 Jelle Fresen

CC Nikita Fruk

Repo[platform/frameworks/support](#)

Branch[androidx-master-dev](#)

Parent[2a9664f](#)

TopicNo 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 kotlin synthetic with findViewById

kotlinx.android.synthetic is no longer a recommended practice. Removing in favour of explicit findViewById.  
~~XX~~

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 Canary 11+.

**app/build.gradle:**

```
android {  
    ...  
    viewBinding {  
        enabled = true  
    }  
}
```

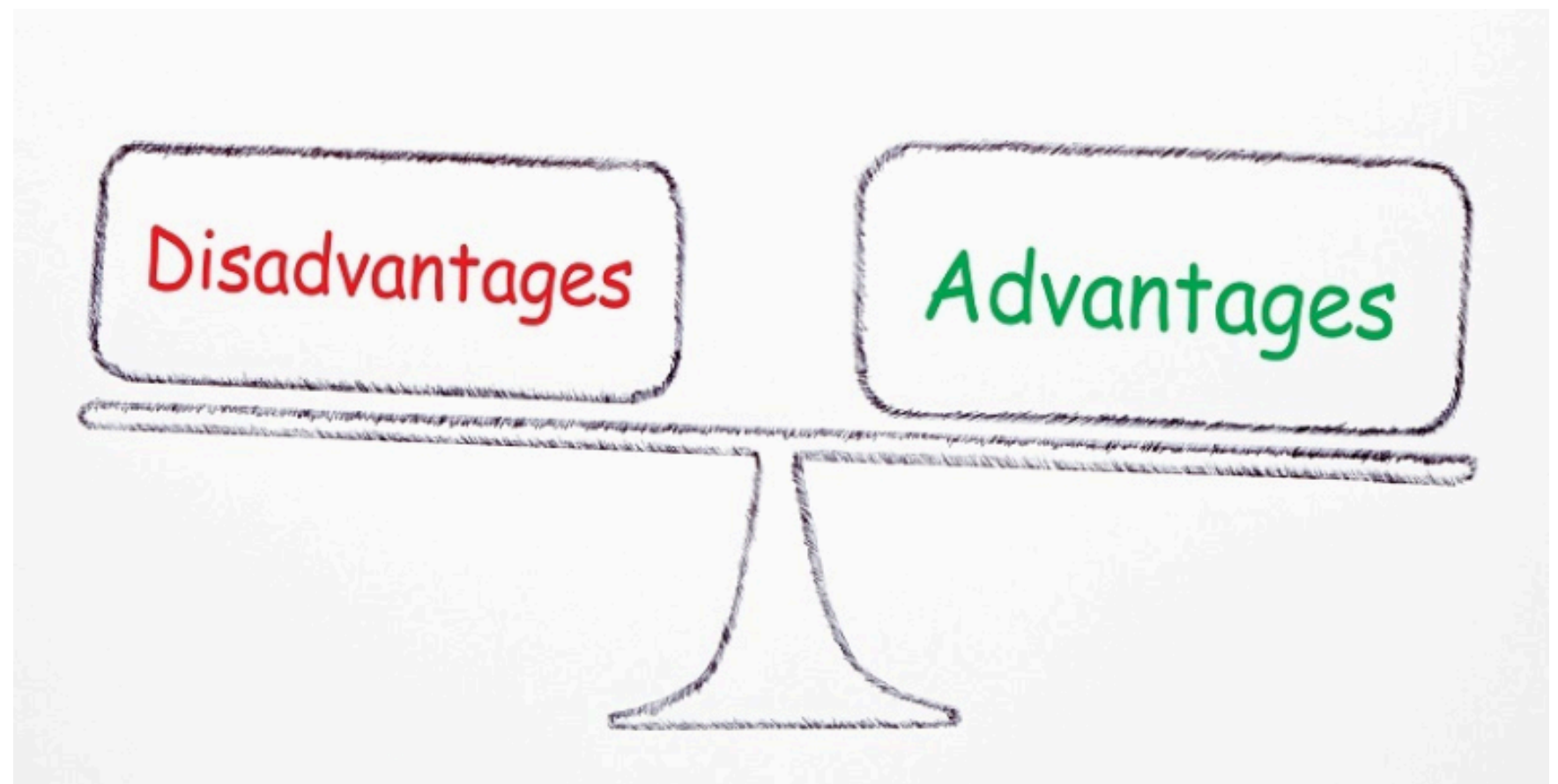
# ViewBinding

**build.gradle:**

```
buildscript {  
    ext.kotlin_version = '1.3.61'  
    repositories {  
        google()  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:3.6.0-rc01'  
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
    }  
}
```

# ViewBinding

- 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.

