



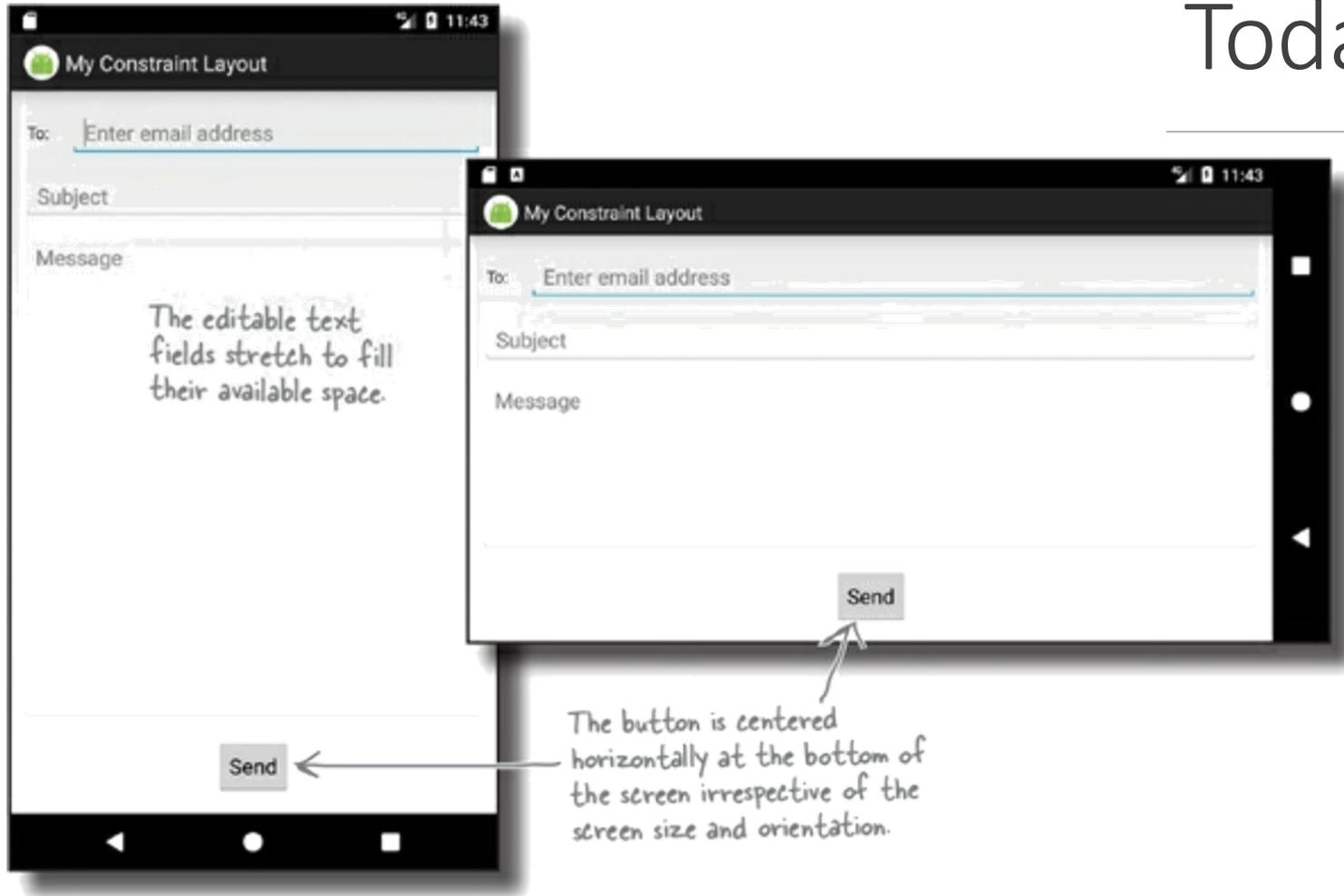
Lab 7 – Constraint Layouts

KUAN-TING LAI

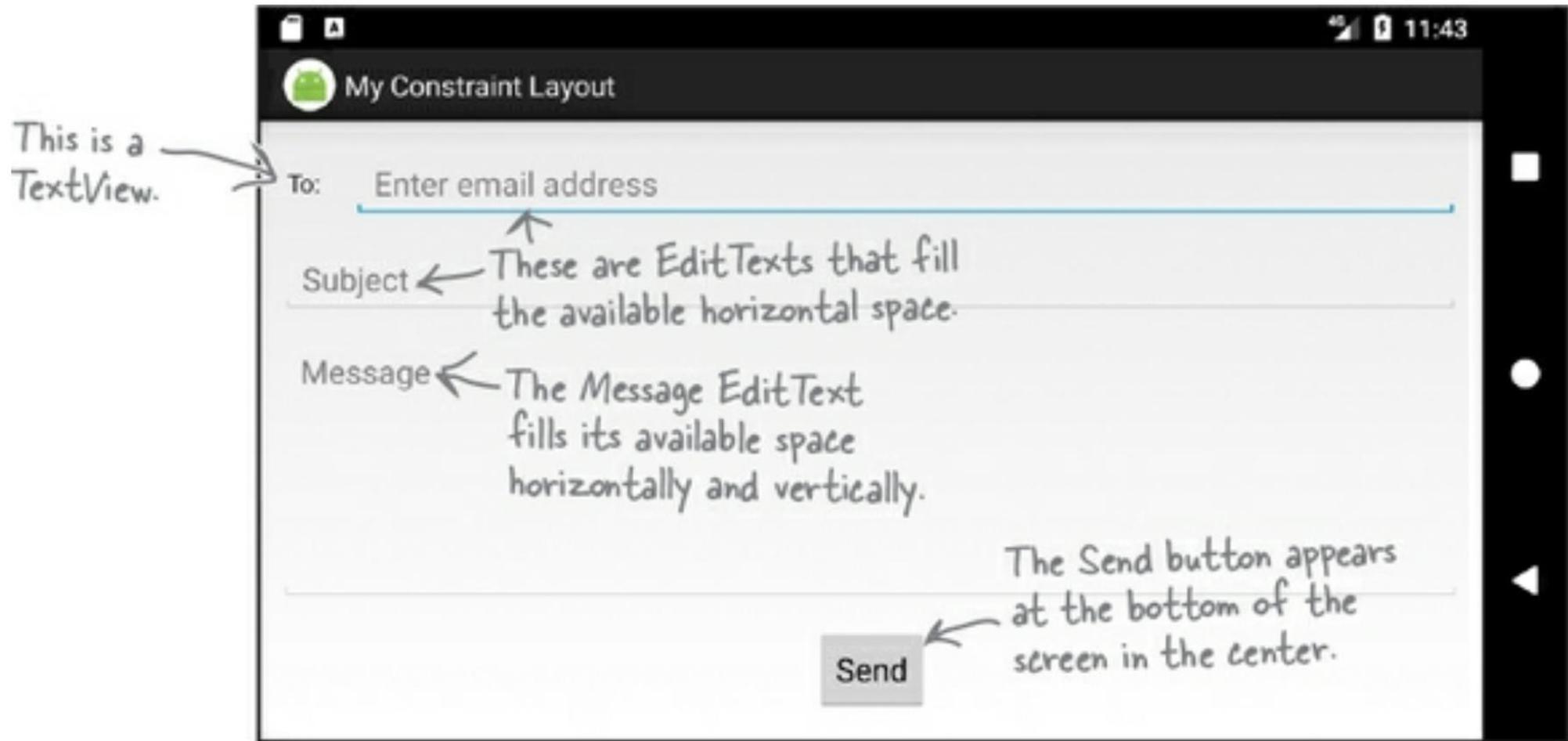
2020/11/1

Today's Lab – Email Form

- Constraint Layout

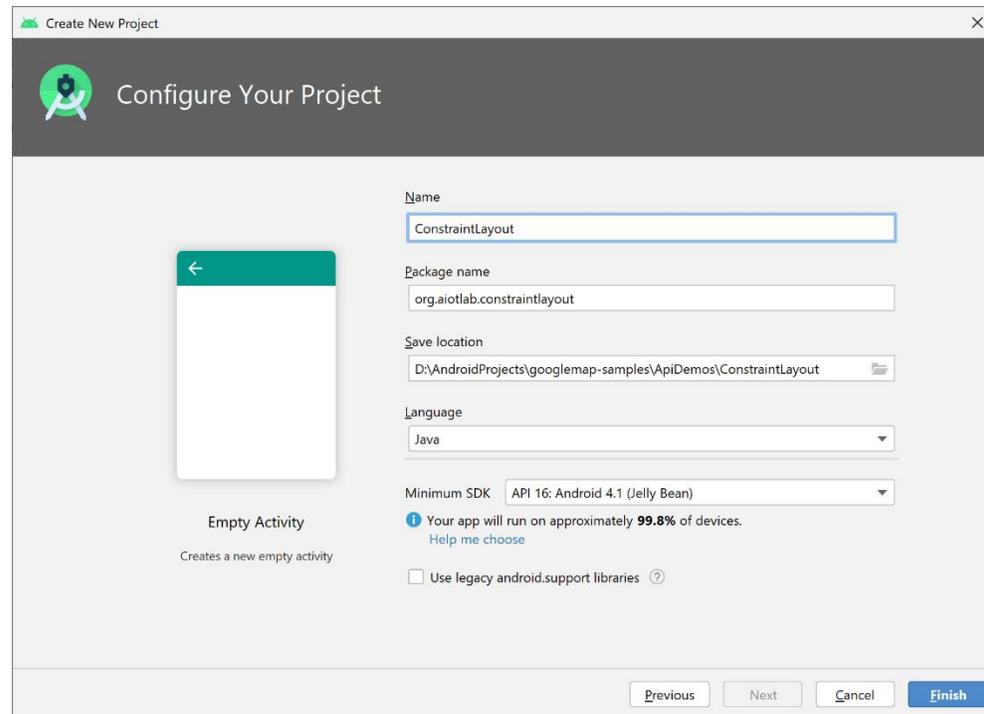


Email Form Layout



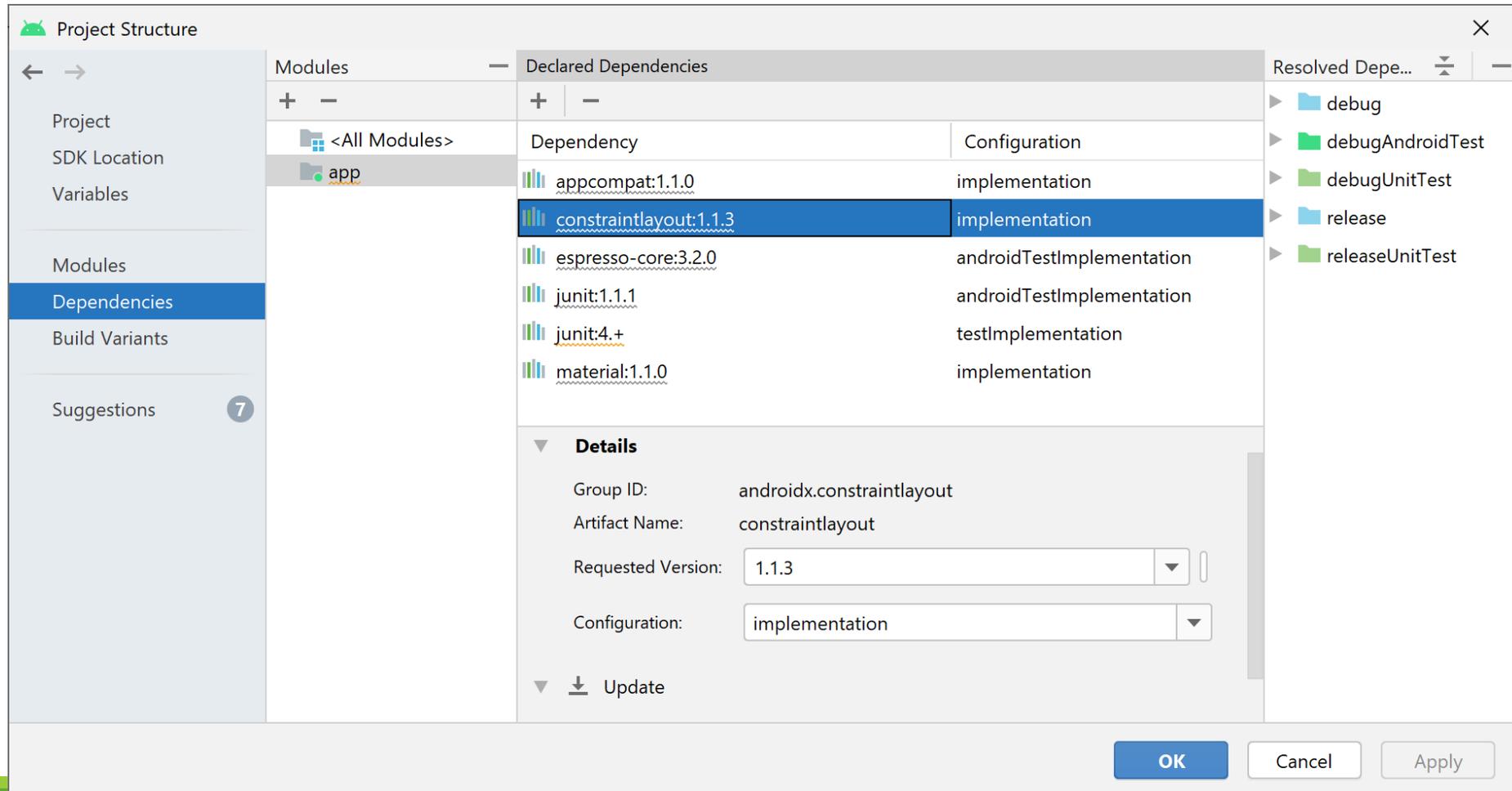
Create a New Project

- Create a new project names ConstraintLayout
- Select **Empty Activity**
- Use default class name “MainActivity”
- Finish



Check if You Have constraintlayout Library

- File -> Project Structure... (Ctrl + Alt + Shift + S)

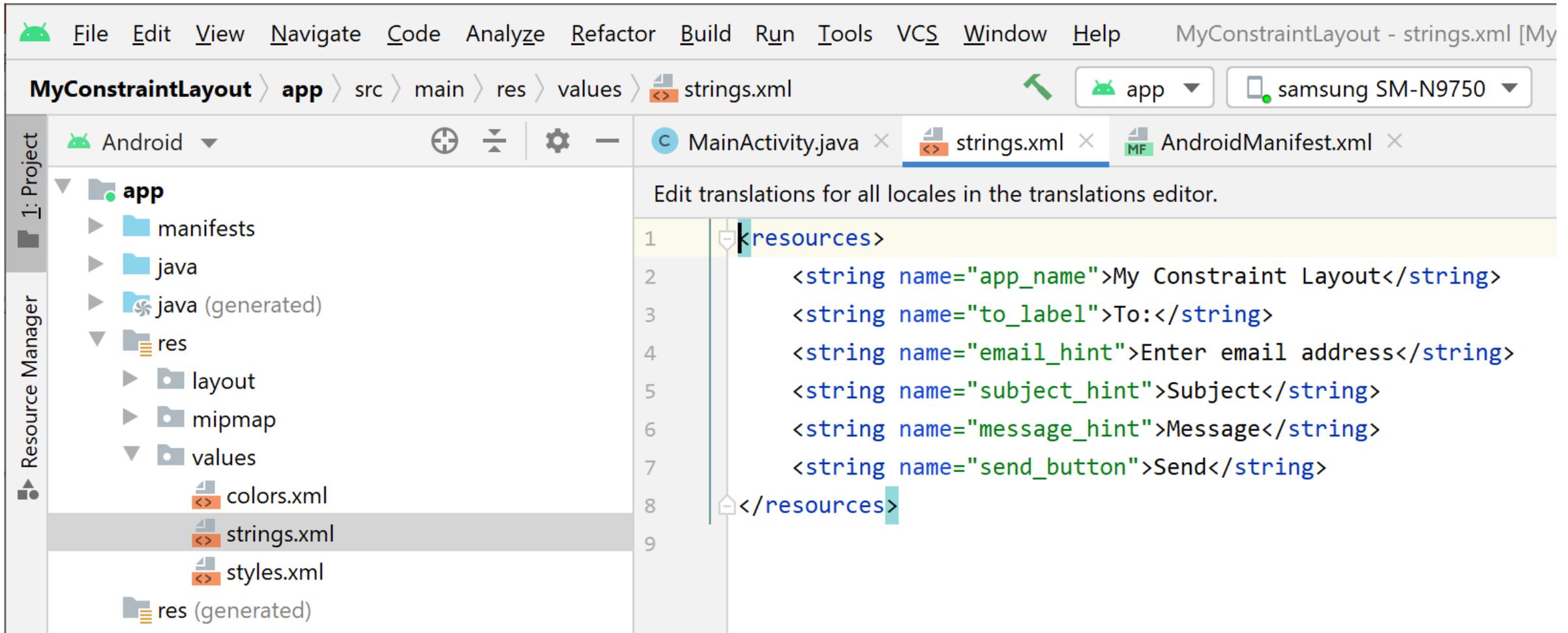


The screenshot shows the 'Project Structure' dialog in Android Studio. The 'Dependencies' tab is selected for the 'app' module. The 'Declared Dependencies' table lists several dependencies, with 'constraintlayout:1.1.3' highlighted. The 'Details' section for this dependency shows the following information:

Property	Value
Group ID	androidx.constraintlayout
Artifact Name	constraintlayout
Requested Version	1.1.3
Configuration	implementation

The 'Resolved Dependencies' section on the right shows the build variants: debug, debugAndroidTest, debugUnitTest, release, and releaseUnitTest.

Add the String Resources to strings.xml



The screenshot shows the Android Studio IDE with the following components:

- Menu Bar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help.
- Toolbar:** MyConstraintLayout > app > src > main > res > values > strings.xml. Includes a back arrow, a device selector (app), and a simulator selector (samsung SM-N9750).
- Project Explorer (Left):** Shows the project structure under '1: Project' and 'Resource Manager'. The 'values' folder is expanded, showing 'colors.xml', 'strings.xml' (selected), and 'styles.xml'. A 'res (generated)' folder is also visible.
- Editor (Right):** Contains the XML code for strings.xml. The text reads: "Edit translations for all locales in the translations editor." The code is as follows:

```
1 <resources>
2     <string name="app_name">My Constraint Layout</string>
3     <string name="to_label">To:</string>
4     <string name="email_hint">Enter email address</string>
5     <string name="subject_hint">Subject</string>
6     <string name="message_hint">Message</string>
7     <string name="send_button">Send</string>
8 </resources>
```

Use Constraint Layout in “activity_main.xml”

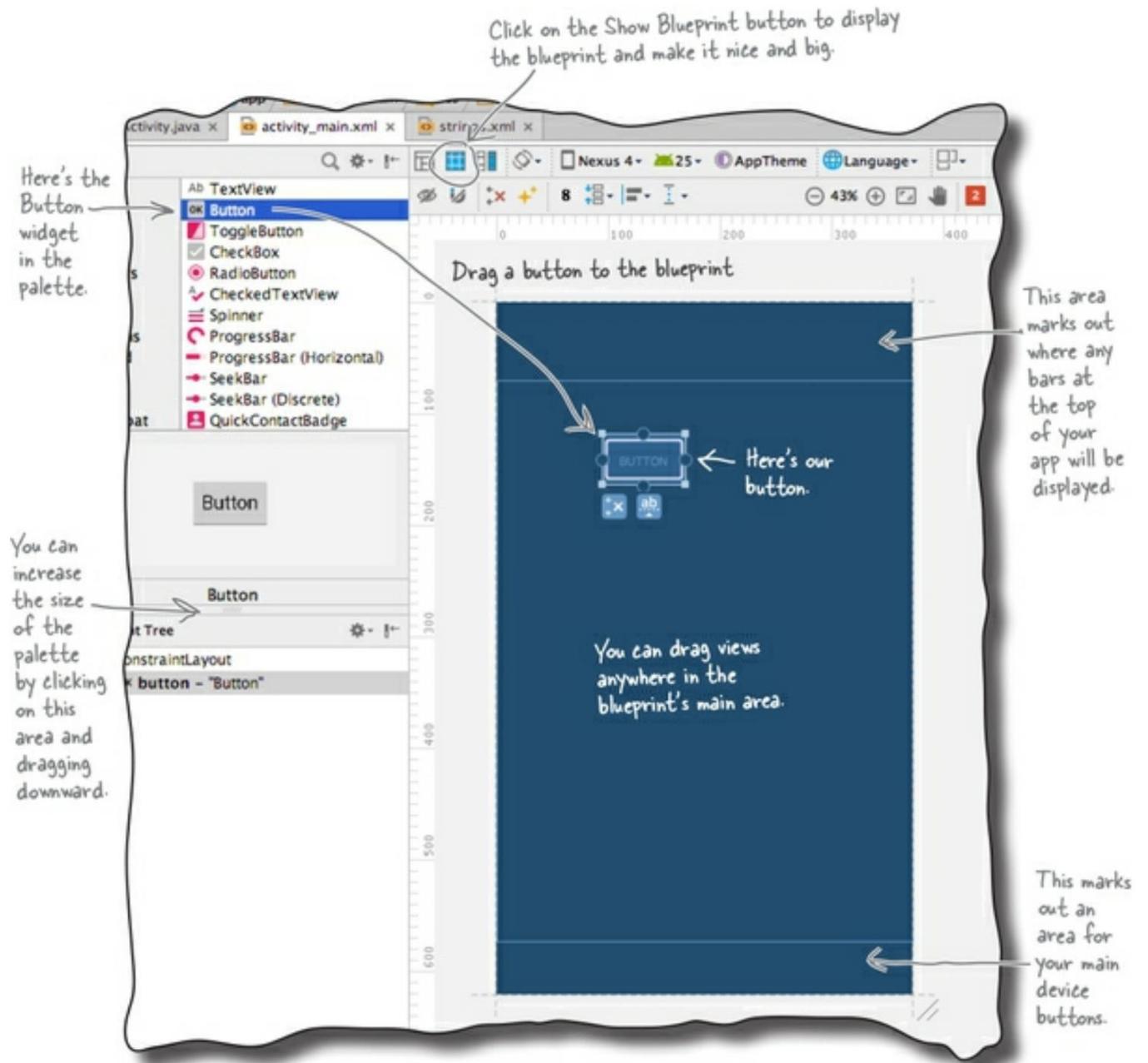
The screenshot displays the Android Studio interface with the following components:

- Top Bar:** Menu items (File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help) and the title bar "ConstraintLayout - activity_main.xml [ConstraintLayout.app] - Android Studio".
- Toolbar:** Navigation icons for back, forward, search, and other actions.
- Project View (Left):** A tree view showing the project structure: `app` (manifests, java, res, mipmap, values, themes, Gradle Scripts). The `res/layout` folder is expanded, and `activity_main.xml` is selected.
- Code Editor (Center):** Displays the XML code for `activity_main.xml`. The code is as follows:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity">
8
9 </androidx.constraintlayout.widget.ConstraintLayout>
```
- Right Panel:** Includes the "Run 'app' with Coverage" button, "Code", "Split", and "Design" tabs, and the "Layout Validation" section.

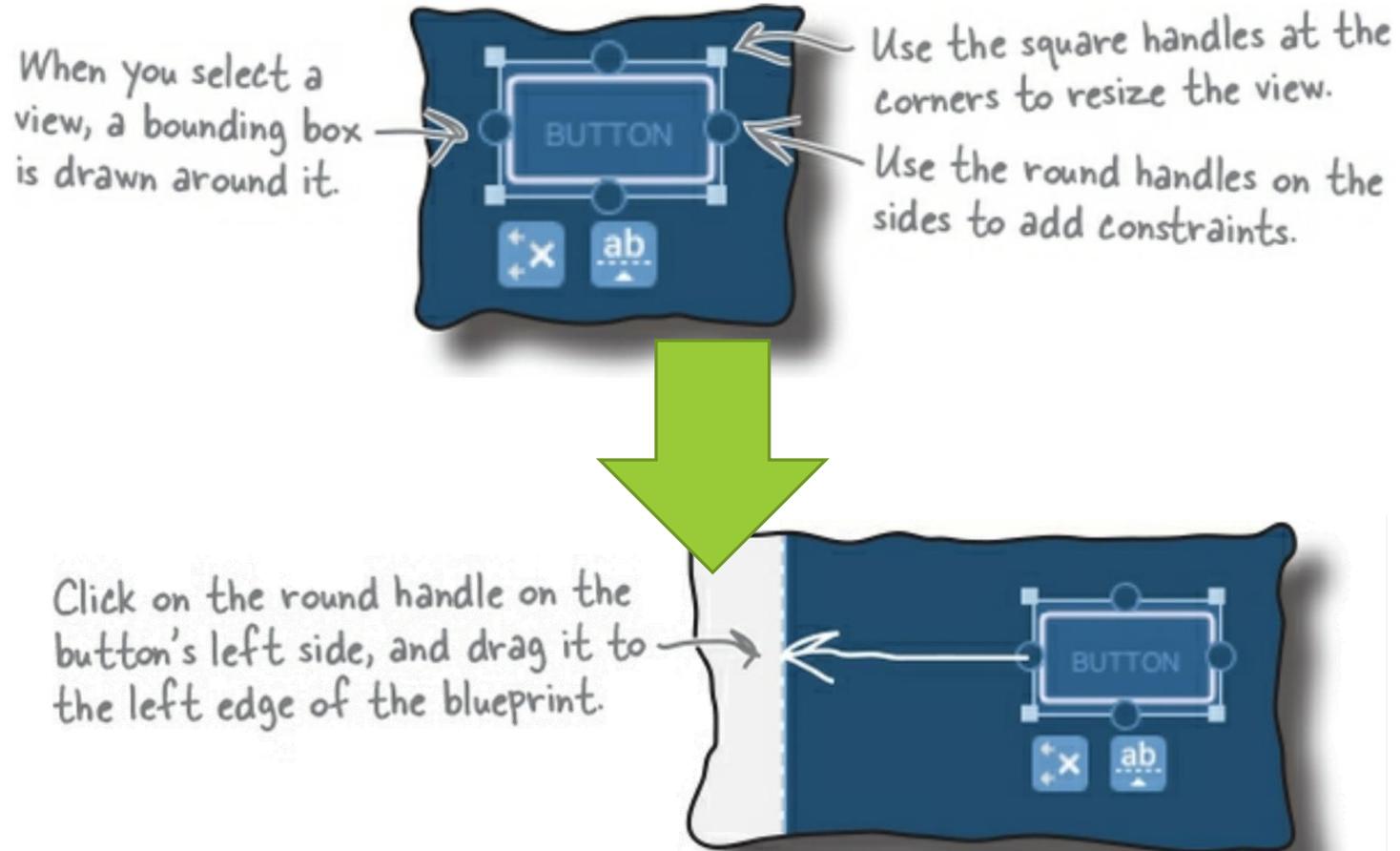
Use Blueprint

- Click Show Blueprint Button
- Drag & drop a Button

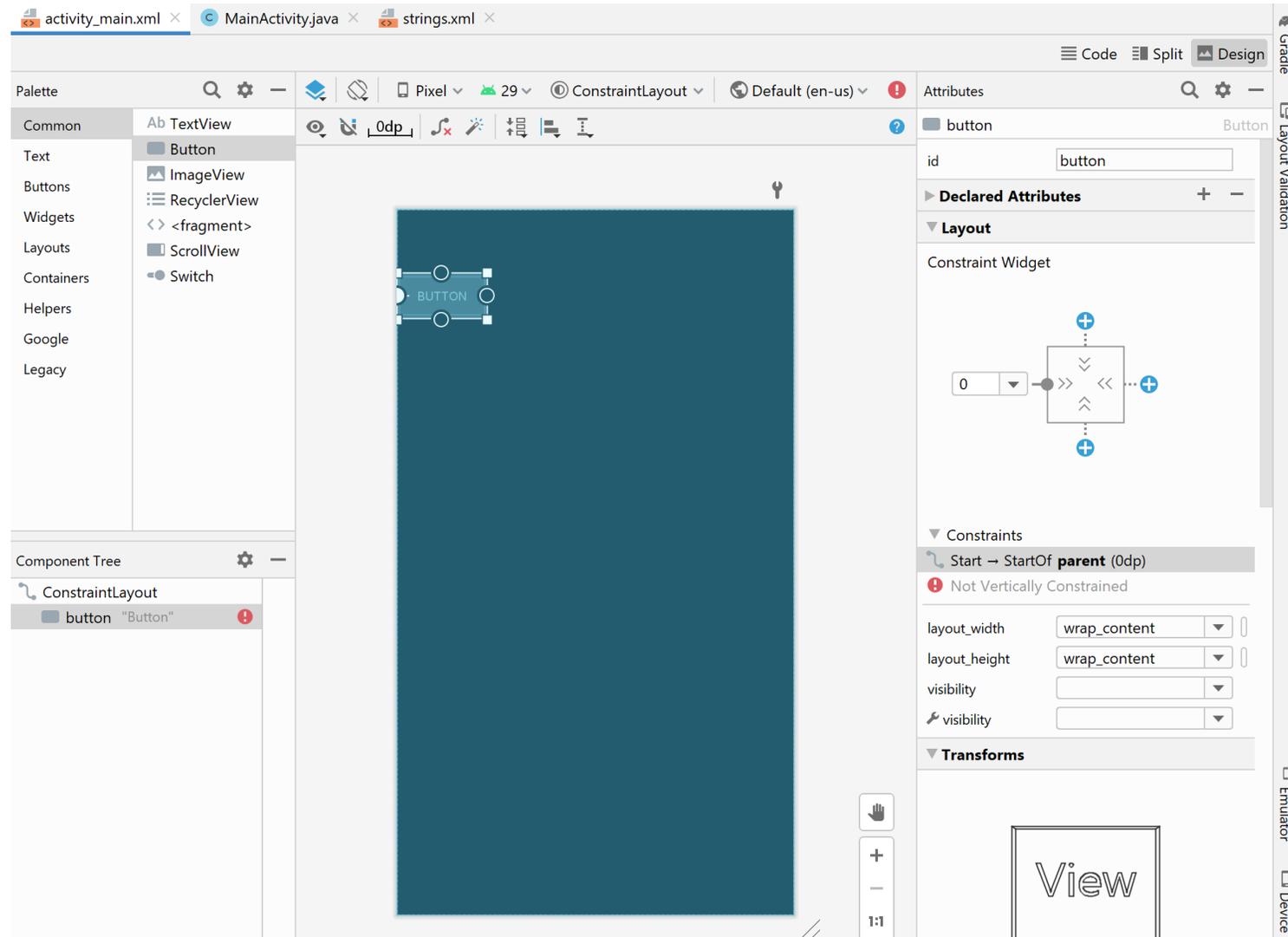


Add Horizontal Constraint

- Add left constraint

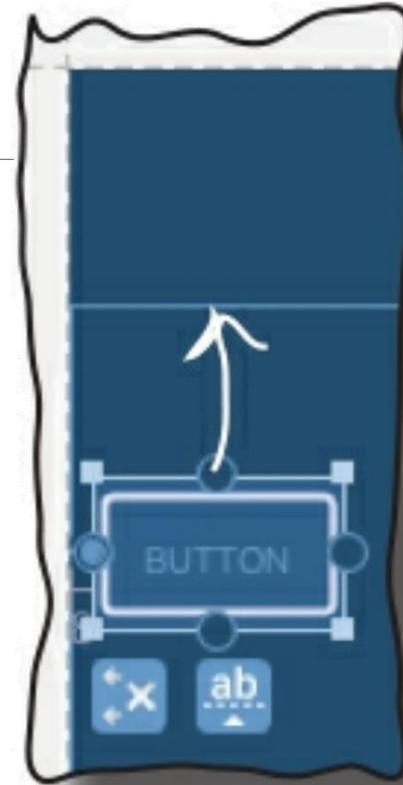


Result: Button slides to the Edge



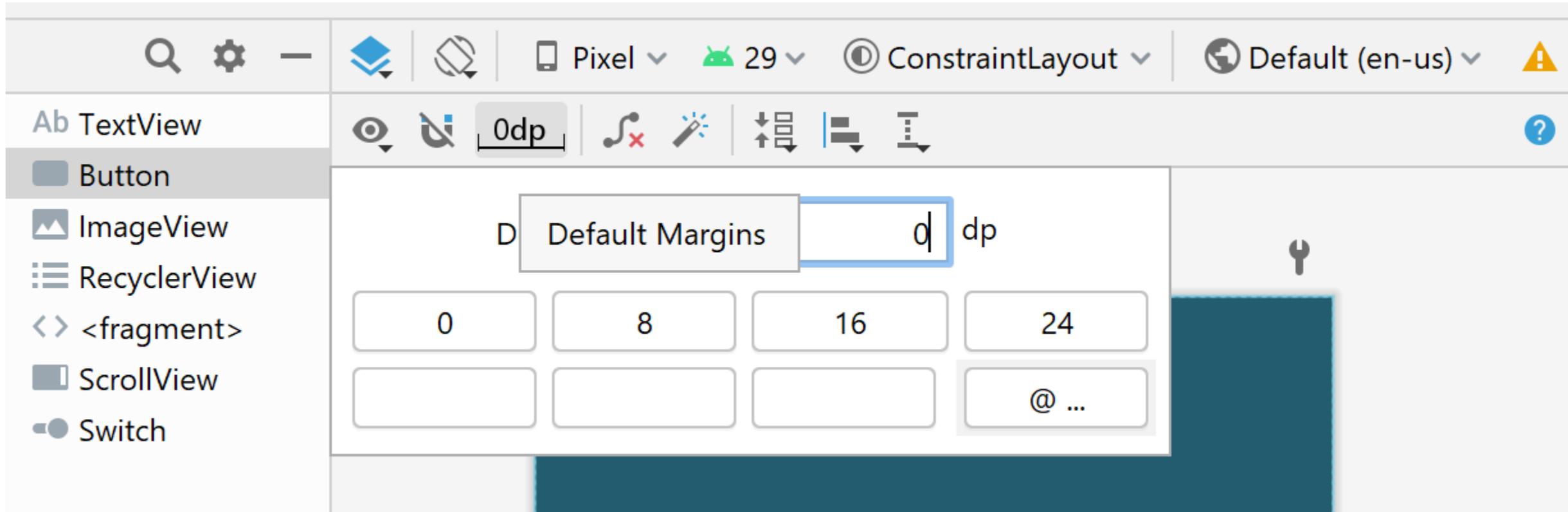
Add Vertical Constraint

- Button slides to the top

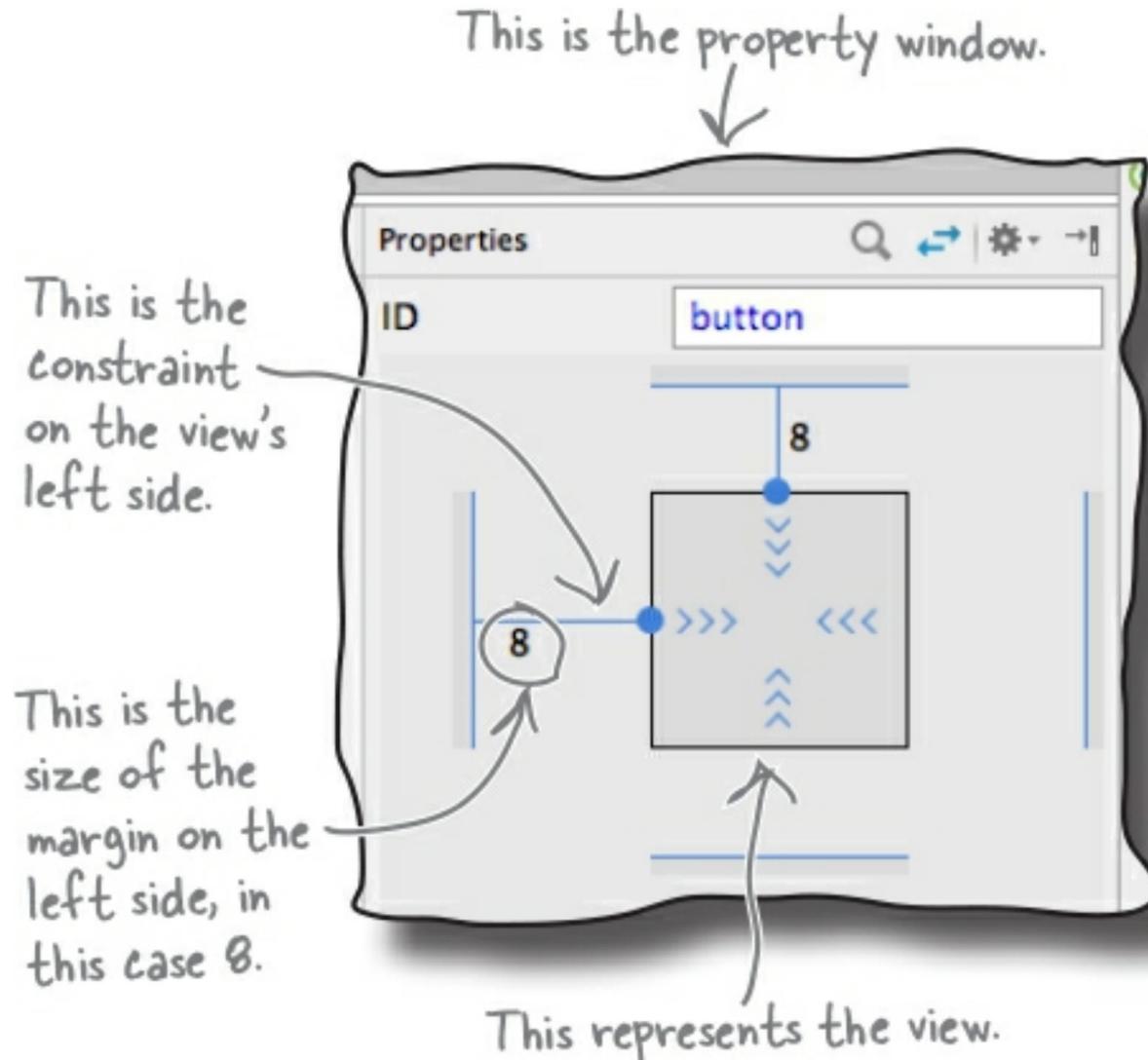


Click on the round handle on the button's top edge, and drag it to the top of the blueprint.

Set Default Margin to 8



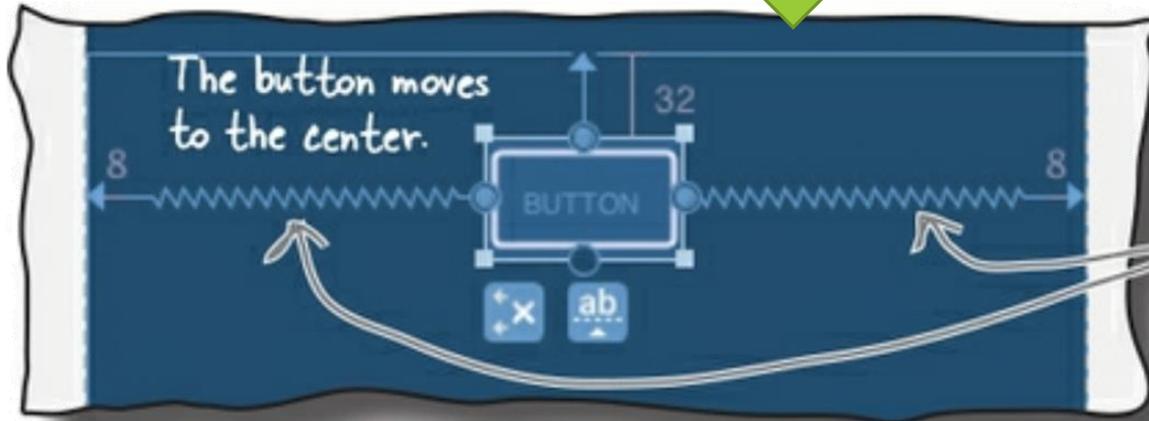
Set Button Margins



Add Right Constraint

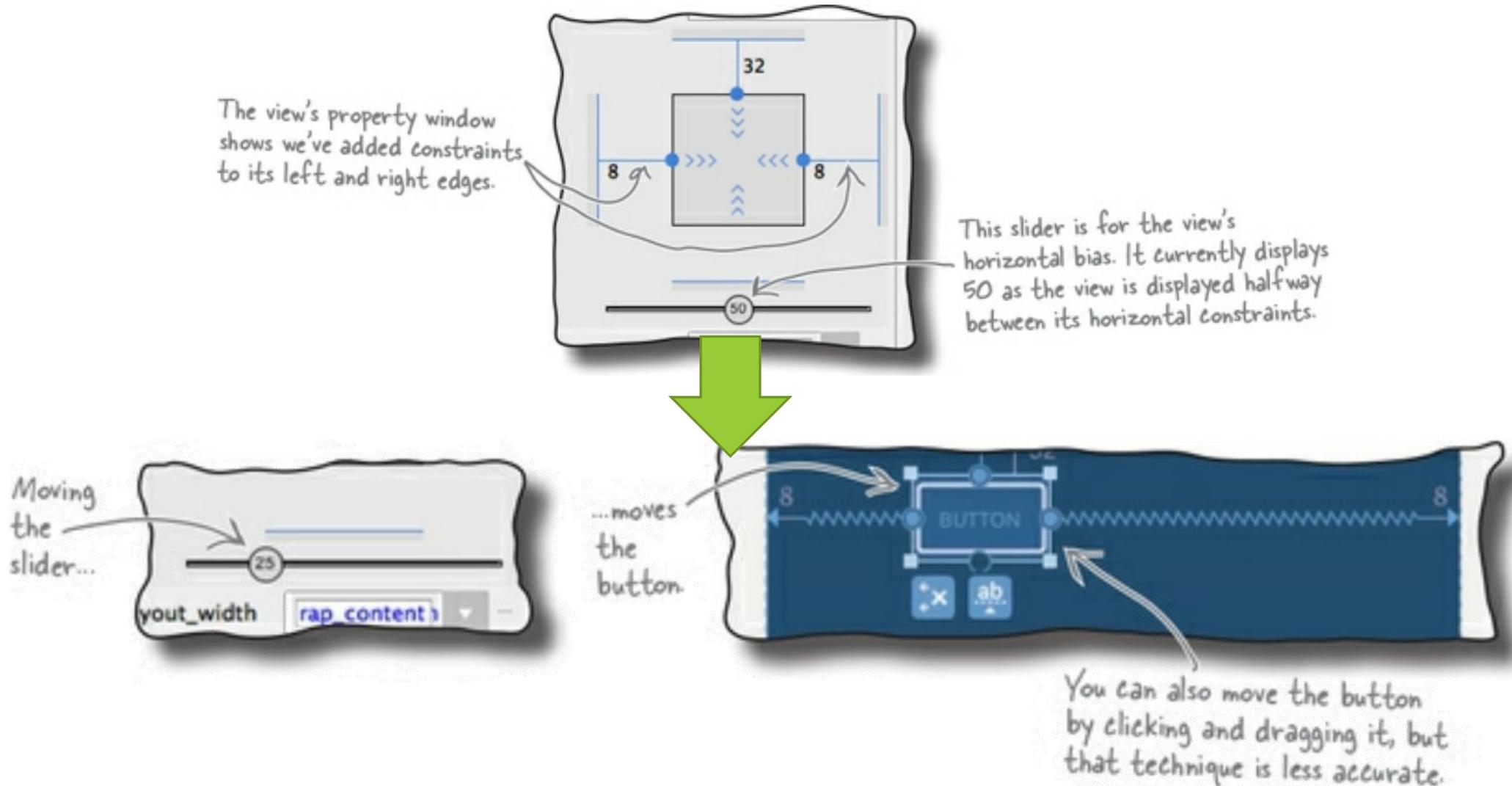


Click on the constraint handle on the button's right edge, and drag it to the right edge of the blueprint.



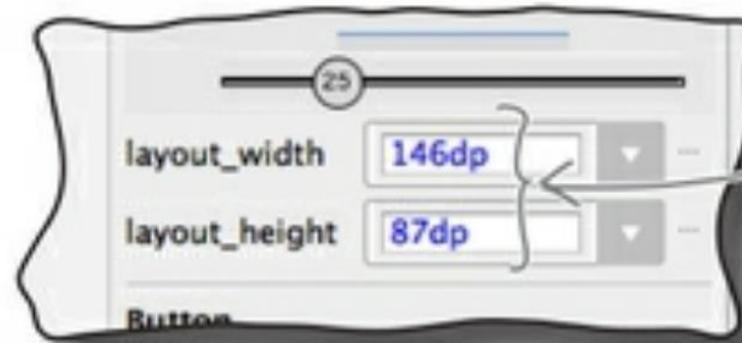
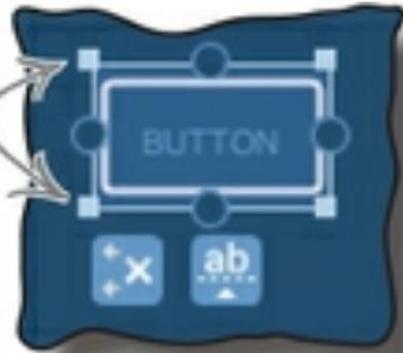
Constraints on opposite sides of a view are displayed as springs.

Adjust a view's Position by Updating its Bias



Resize a view

You can resize a view using the square resizing handles on its corners.



You can also hardcode the width and height in the view's property window.

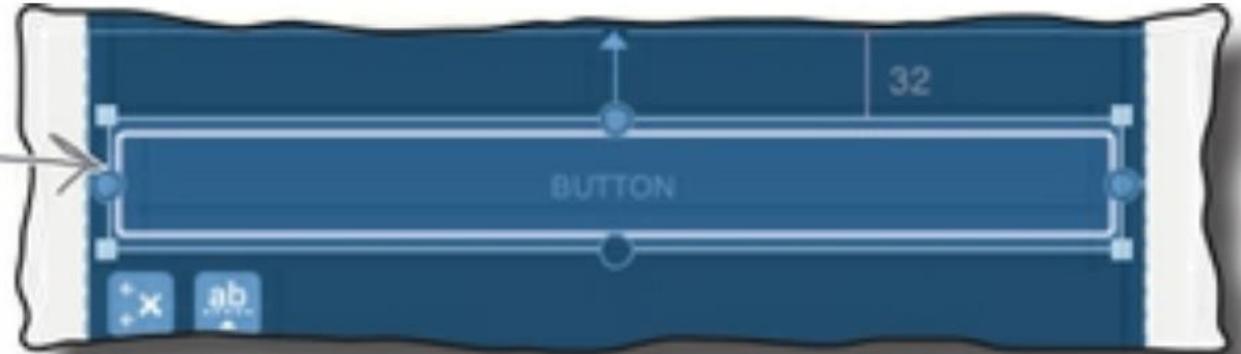


Setting the width and height to "wrap_content" makes it just large enough to display its contents, just as it does in other layouts.

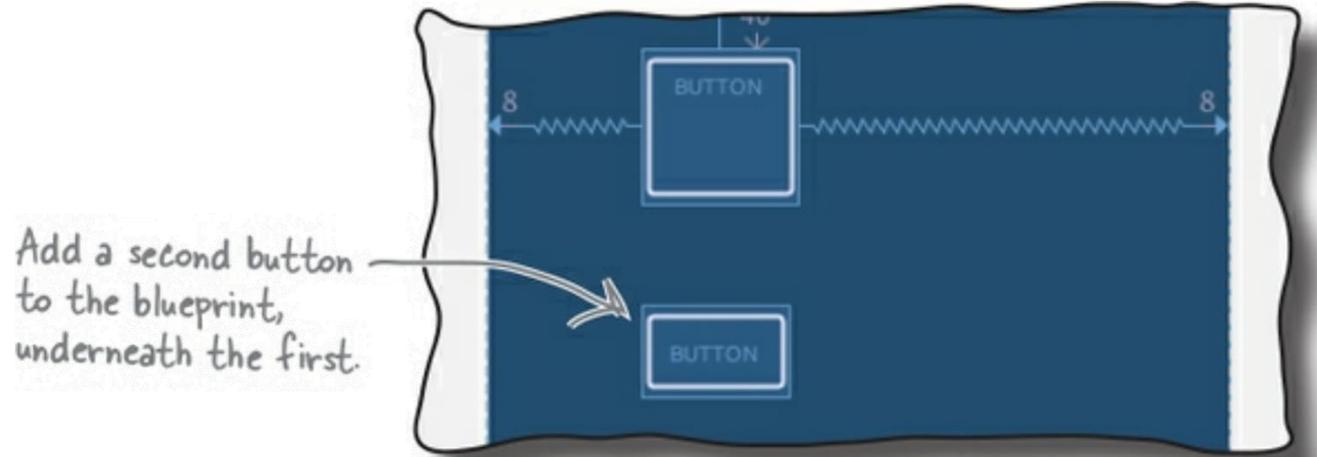
Match the view's Constraint



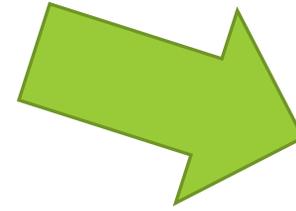
Set the width to 0dp, and the button is sized to match its constraints.



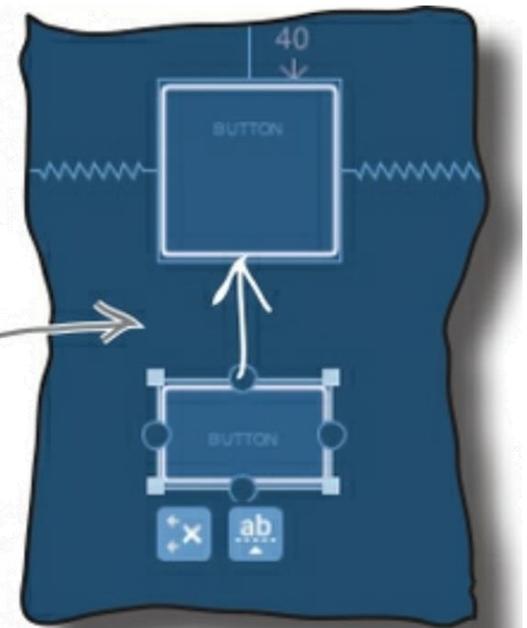
Add 2nd Button

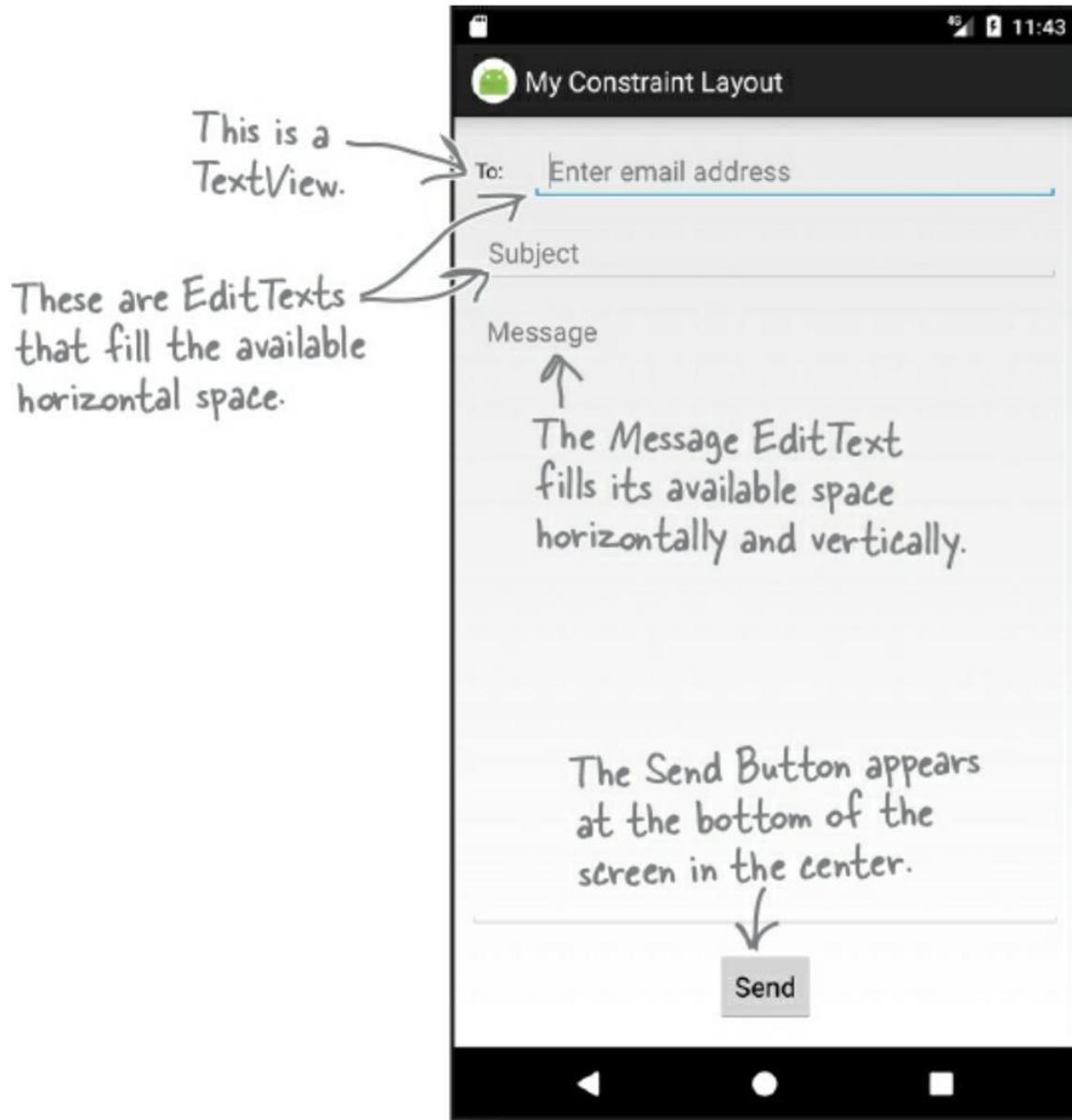


Add a second button to the blueprint, underneath the first.



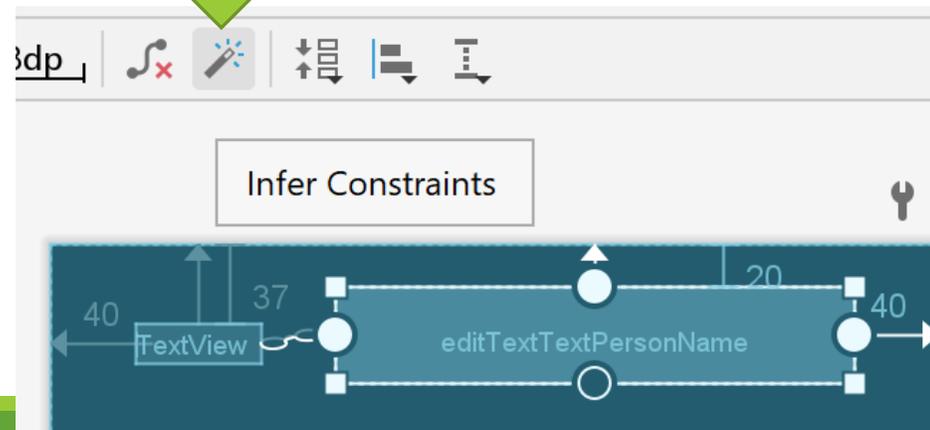
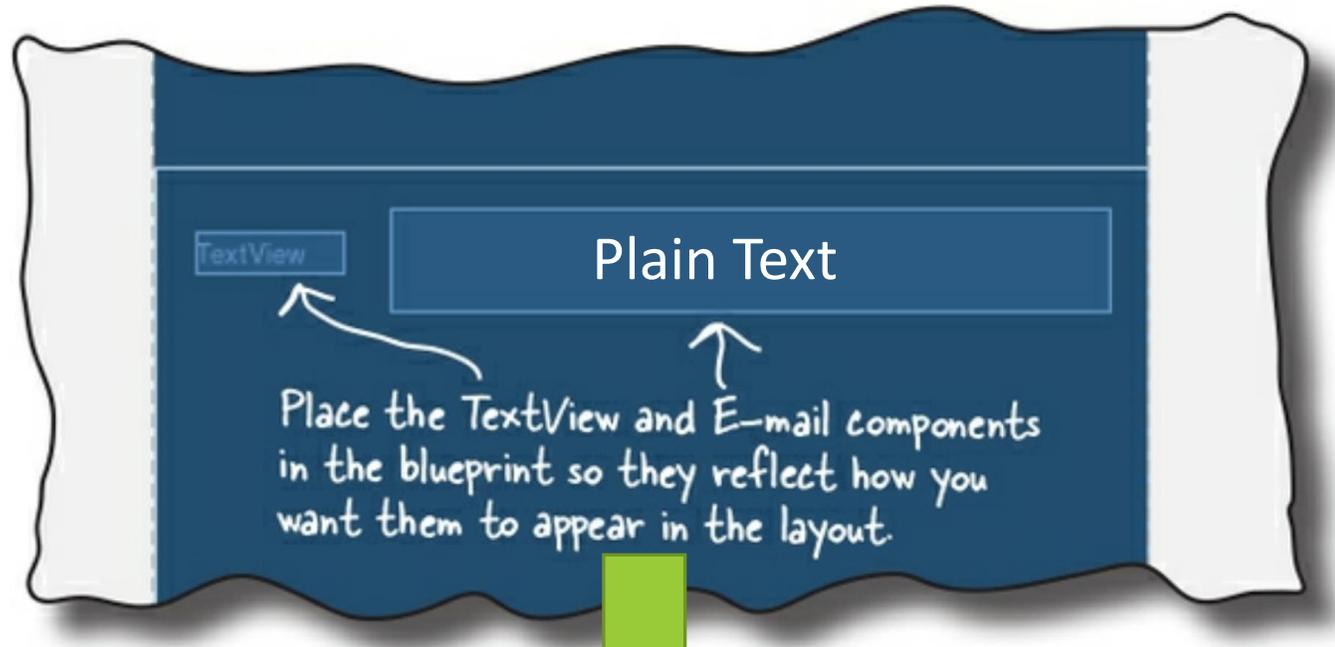
This adds a constraint attaching the top of one button to the bottom edge of the other.





Build the Email Layout

Use “Infer Constraints Button”



Change the Text Value

Update this property to change the TextView's text value.



```
android:id="@+id/to_label"  
android:text="@string/to_label"
```

Android Studio adds these lines of code when you change the view's ID and text value.

Final Result

