Android Accessibility for Native App Developers

@pauljadam
Mobile Accessibility Specialist @Pearson
pauljadam.com and accessw3.com
Developer of a11yTools for iOS and Safari

https://www.icloud.com/keynote/0fpBCz7zrAOwtxcmy53v0b92Q#Android_Accessibility_for_Native_App_Developers

Abstract

- Learn how to build Accessible Native Android Apps! Learn automated and manual accessibility testing techniques. Testing on device, running the simulator, and analyzing project code.

Hands Up If Your Company Has A Mobile App?

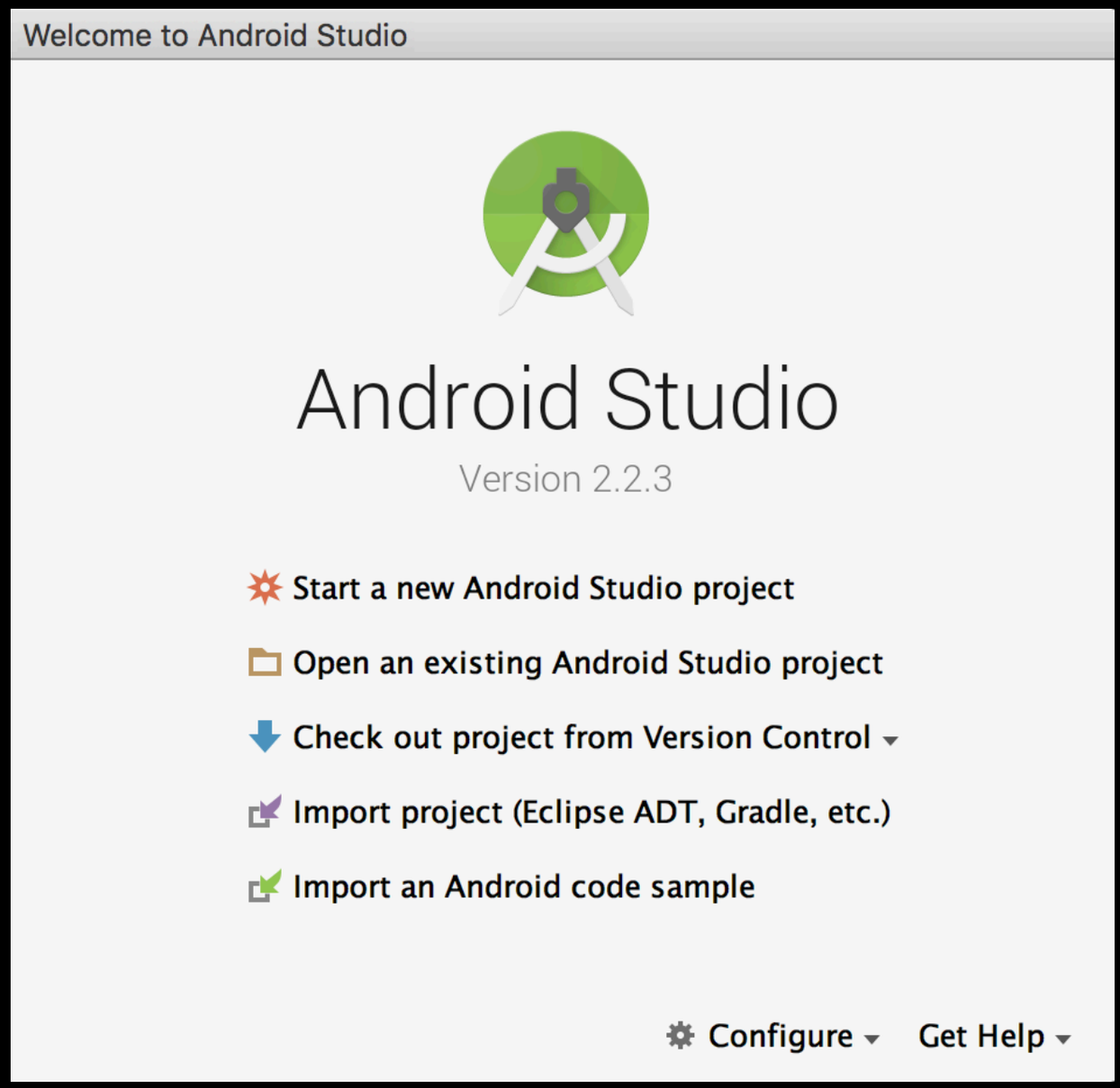
- % of Room with Android App = % ?
- % of Room with both Android and iOS App = %?
- Pure Native Android?
- Hybrid or Web Views?
- Your app is most likely NOT Accessible without a11y design and testing at EVERY stage of development!

To Learn List:

- TalkBack Screen Reader Testing
- Bluetooth Keyboard-Only Testing
- Android Accessibility API & Limitations
- WCAG 2.0 AA
- Android Studio A11y Dev
  - XML Static Fixes & Lint
  - Java Dynamic Fixes
- Google Accessibility Scanner

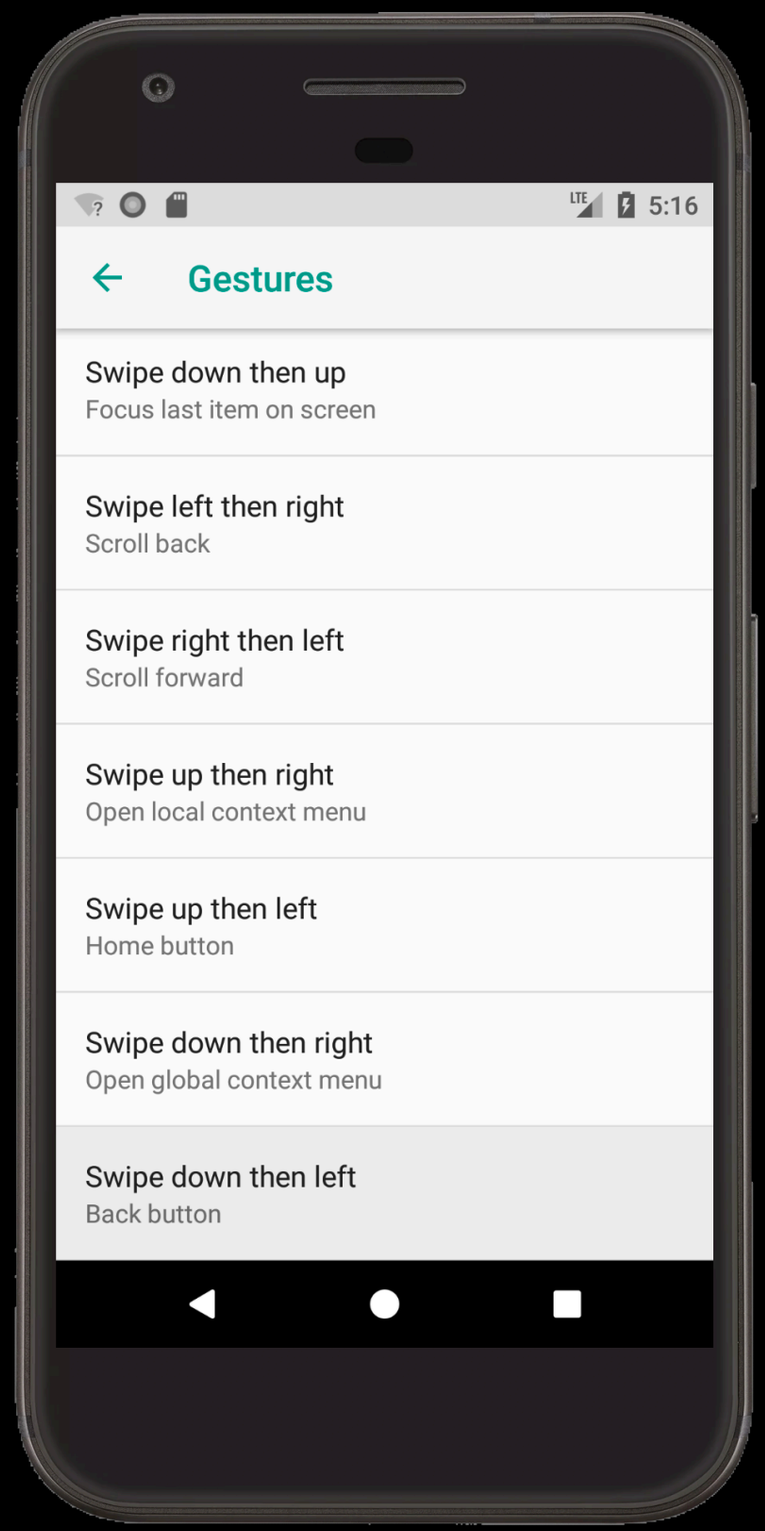
Android A11y Dev on Windows & macOS!

- Download Android Studio to code accessible Android apps
- Works on Windows or Mac



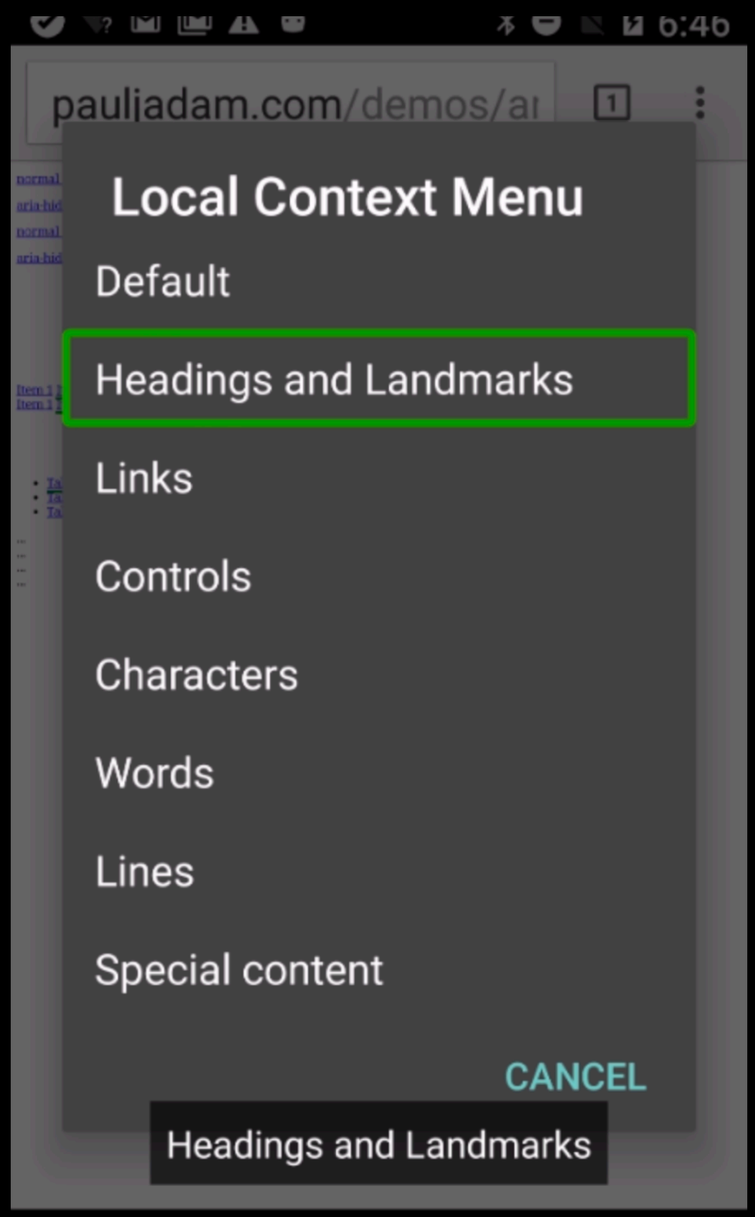
TalkBack Review

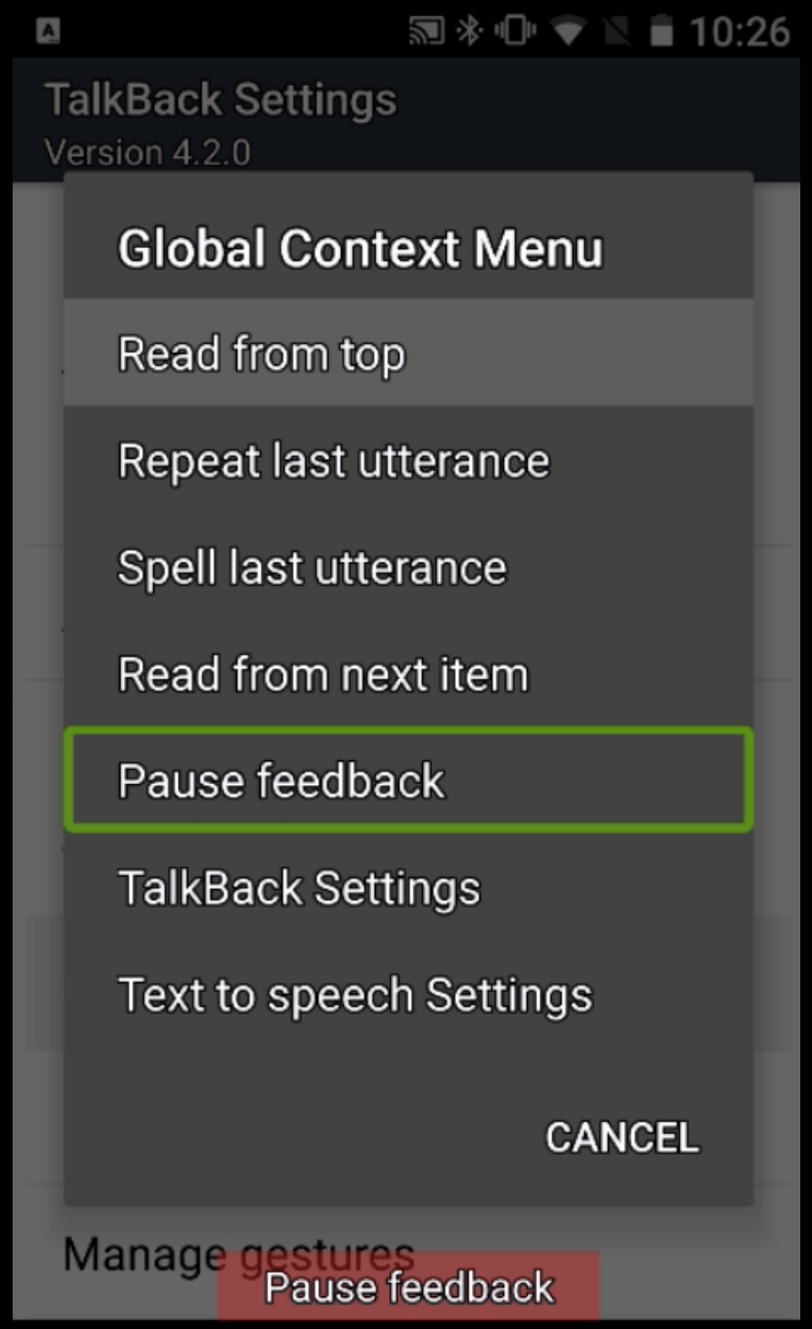
- Explore by Touch
- Gesture Commands
- Global and Local Context Menus
- Android Keyboard Accessibility Testing and Differences from iOS
- Developer settings > Display speech output
- High Contrast Text (contrast false negatives)
- Pause, Enable/Disable Talkback (Accessibility Shortcut)



Global & Local Context Menus

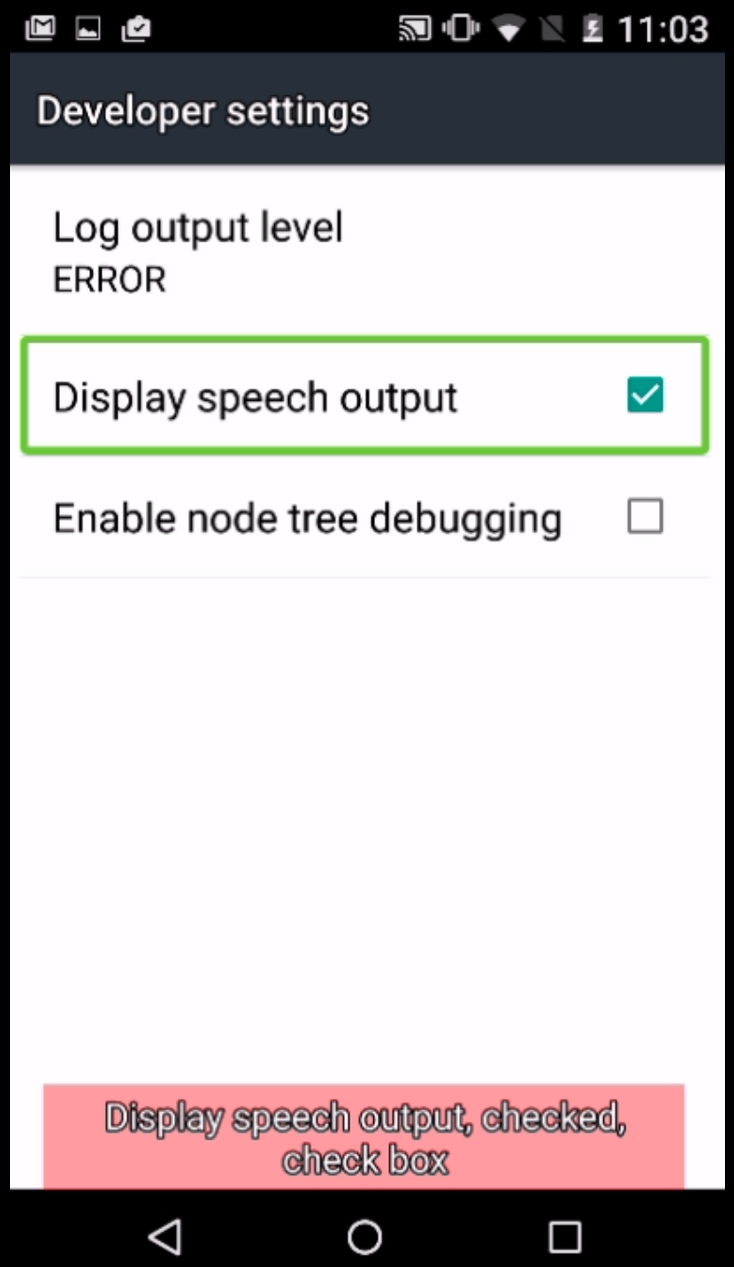
- Use global and local context menus - Android Accessibility Help





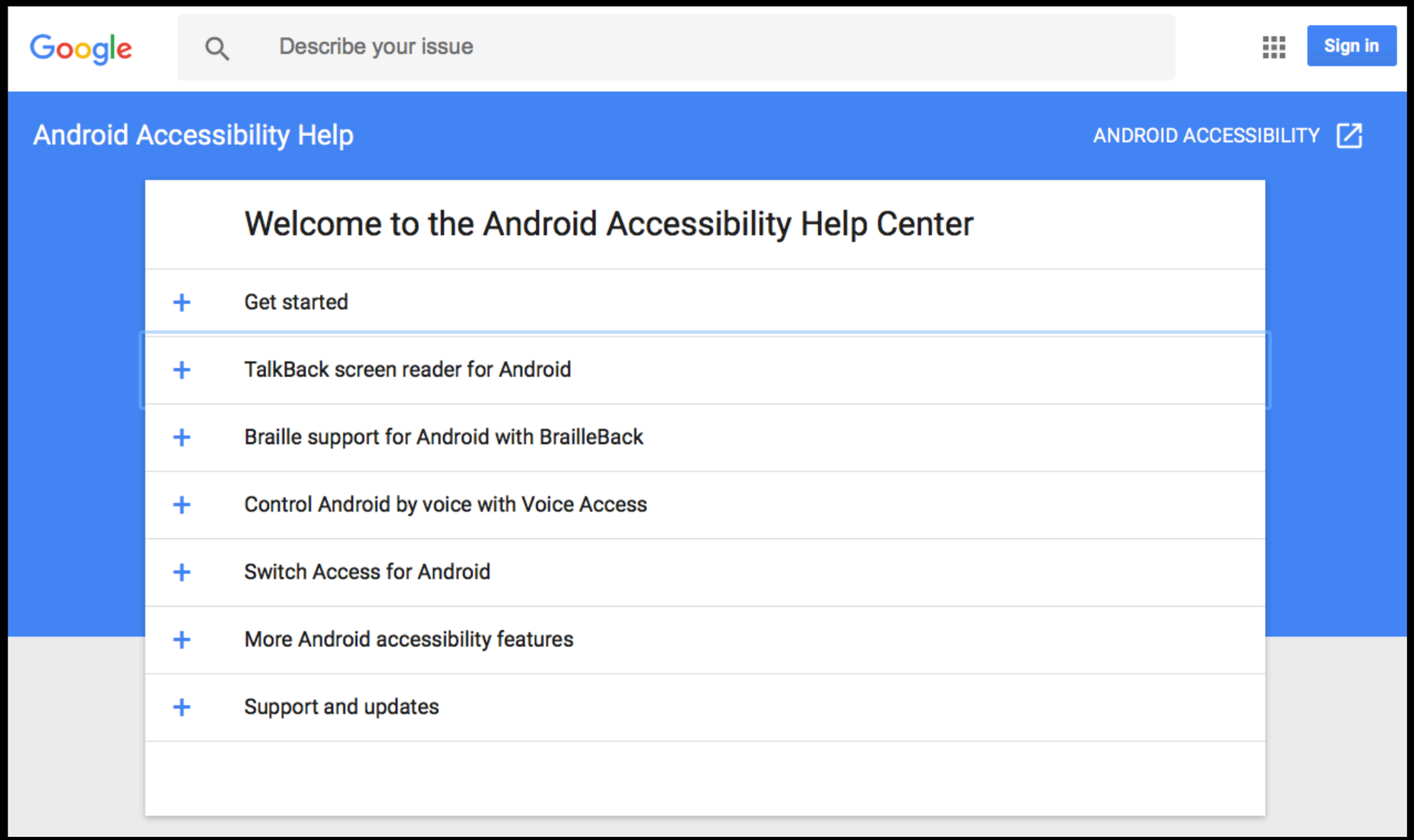
Developer settings > Display speech output

- http://pauljadam.com/demos/talkbackcheatsheet.html

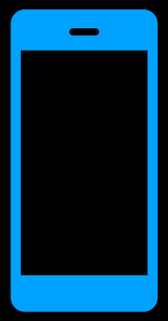


Android Accessibility Help Center

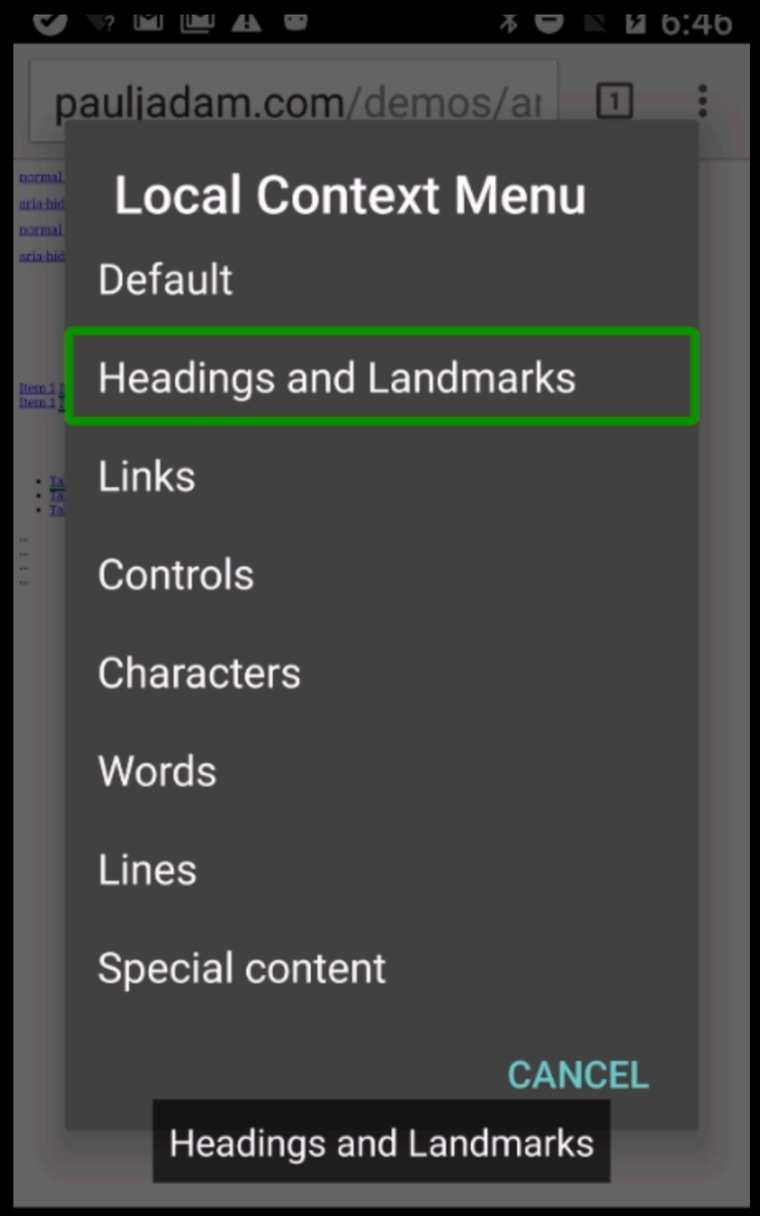
- https://support.google.com/accessibility/android#topic=6007234



Pure Native vs. Hybrid vs. Web-view Apps

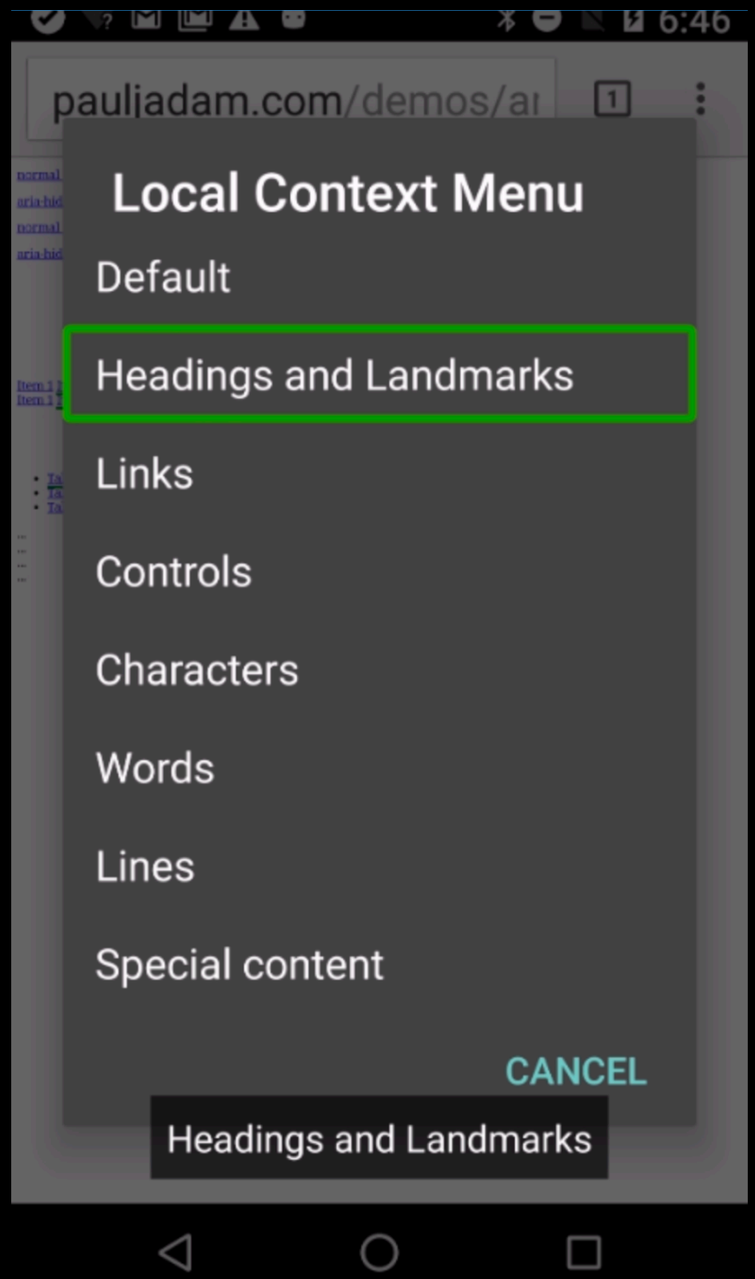


- Web views in a native app have the exact same accessibility as an HTML web page.
- Native Android apps won't have HTML elements in the TalkBack Local Context Menu



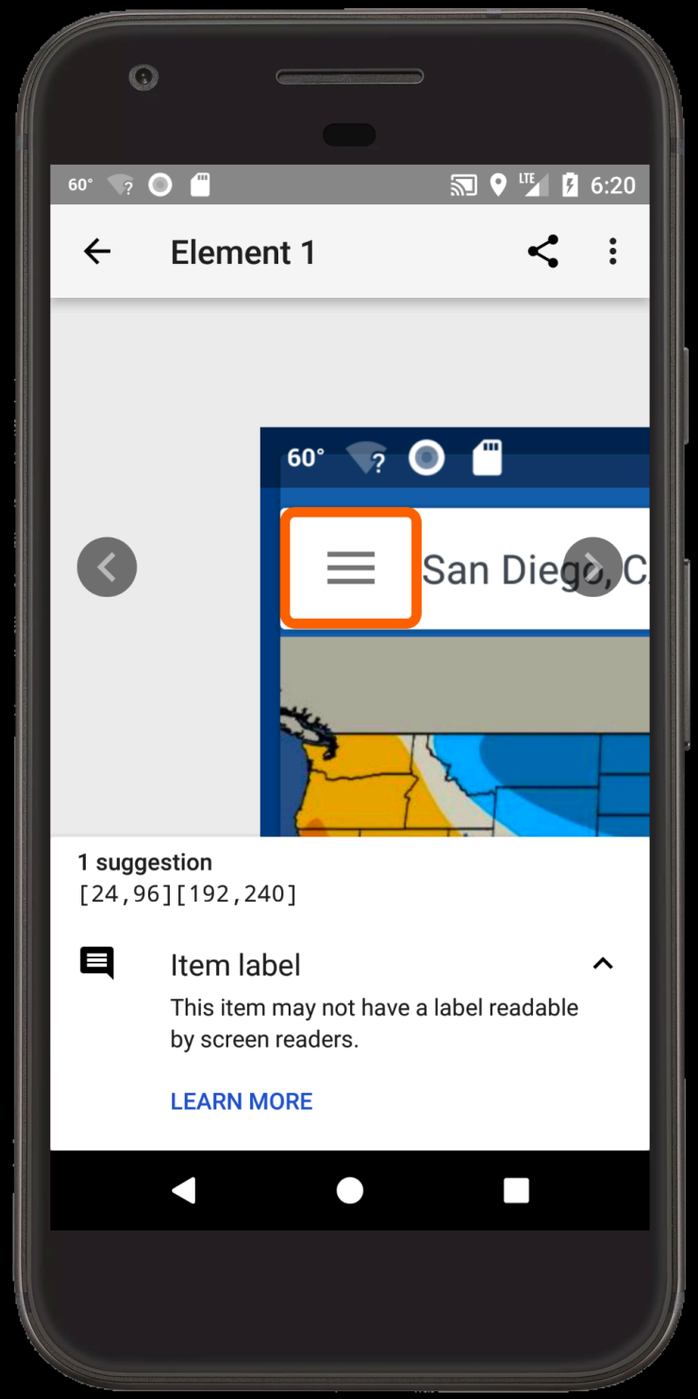
Testing with Screen Readers & Keyboard-Only

- Navigate page by semantic elements
- TAB, swipe, or arrow key through all elements
- Set focus to each element to verify proper accessible name, role, value, and state.
- Activate all dynamic controls using screen reader and then keyboard only
- Check for proper keyboard focus order and focus visibility (not screen reader focus)
- Enter blank and invalid values into forms to test error validation



DEMO Accessibility Problems with Popular Native Apps

- Weather Channel, Instagram, Facebook, Twitter, etc.



Native Android App Accessibility

- Android Developers Site
  - https://developer.android.com/guide/topics/ui/accessibility/apps.html
- Labeling UI Elements
- Grouping Content
- Making Touch Targets Large
- Sample Accessibility App
  - https://github.com/googlesamples/android-BasicAccessibility

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:contentDescription="@string/share"
    android:src="@drawable/ic_share" />
```

Android Accessibility API Capabilities/ Limitations/Version Numbers

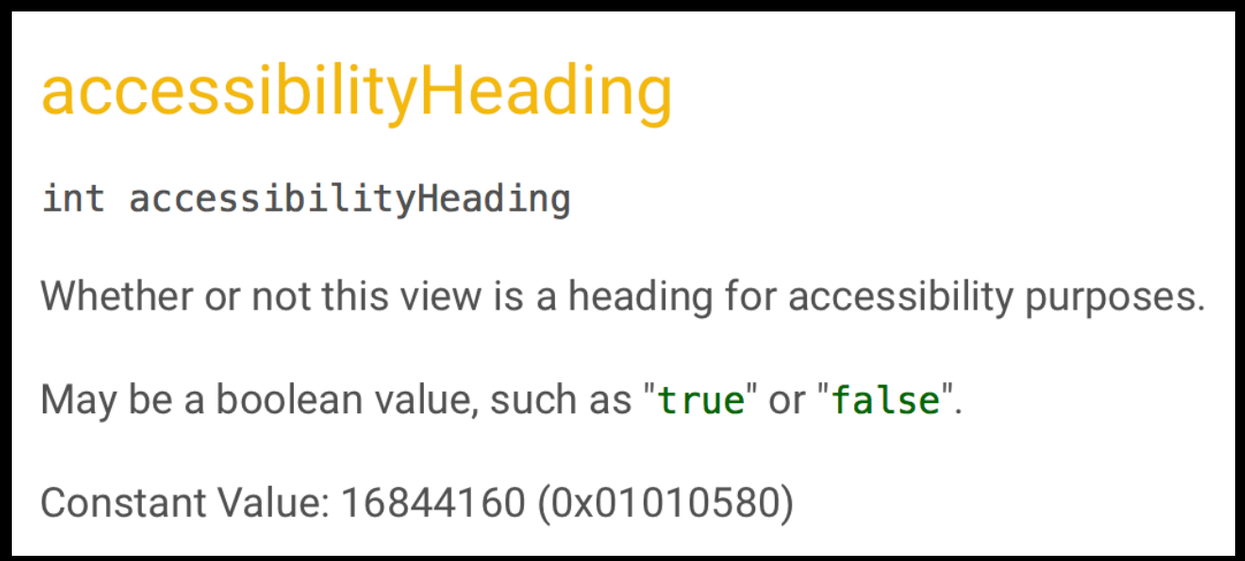
- android:labelFor not in old API levels

- Heading semantics can't be applied to any element easily like iOS with the heading trait.

- New in Android P accessibilityHeading

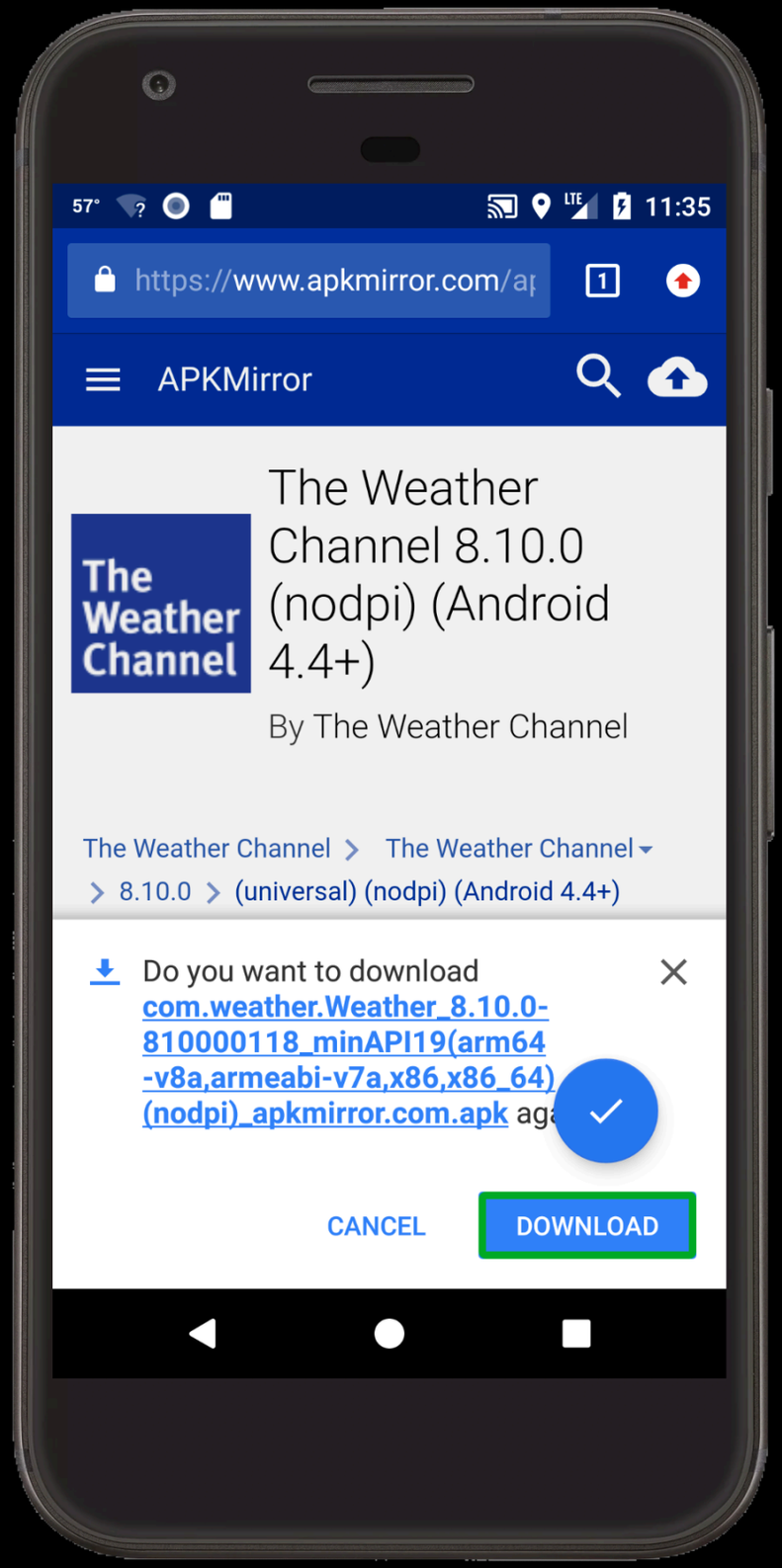
  - If your app displays content that includes logical headers, set the new android:accessibilityHeading attribute to true

  - https://developer.android.com/preview/features.html#a11y



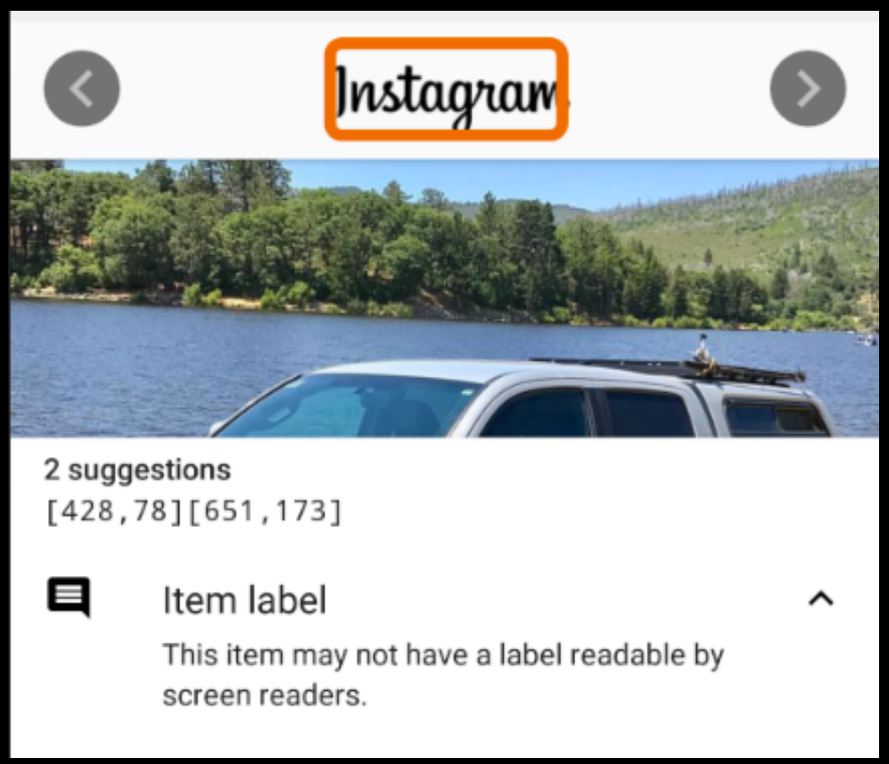
Android Simulator A11y Testing

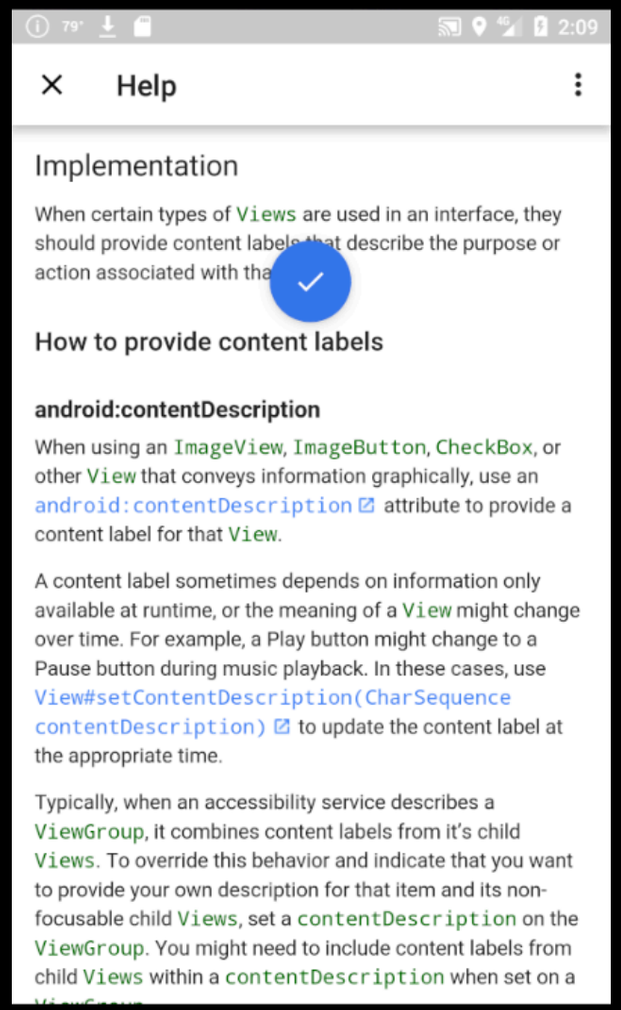
- You don't "need" a real phone!
- Google TalkBack 6.1 APK direct link from a google search.
- Download TalkBack APK inside the Simulator to do screen reader testing without an android device.
- Only Chrome installed in simulator, download Firefox APK in simulator for cross-browser testing.
- I also downloaded Weather Channel APK in Simulator

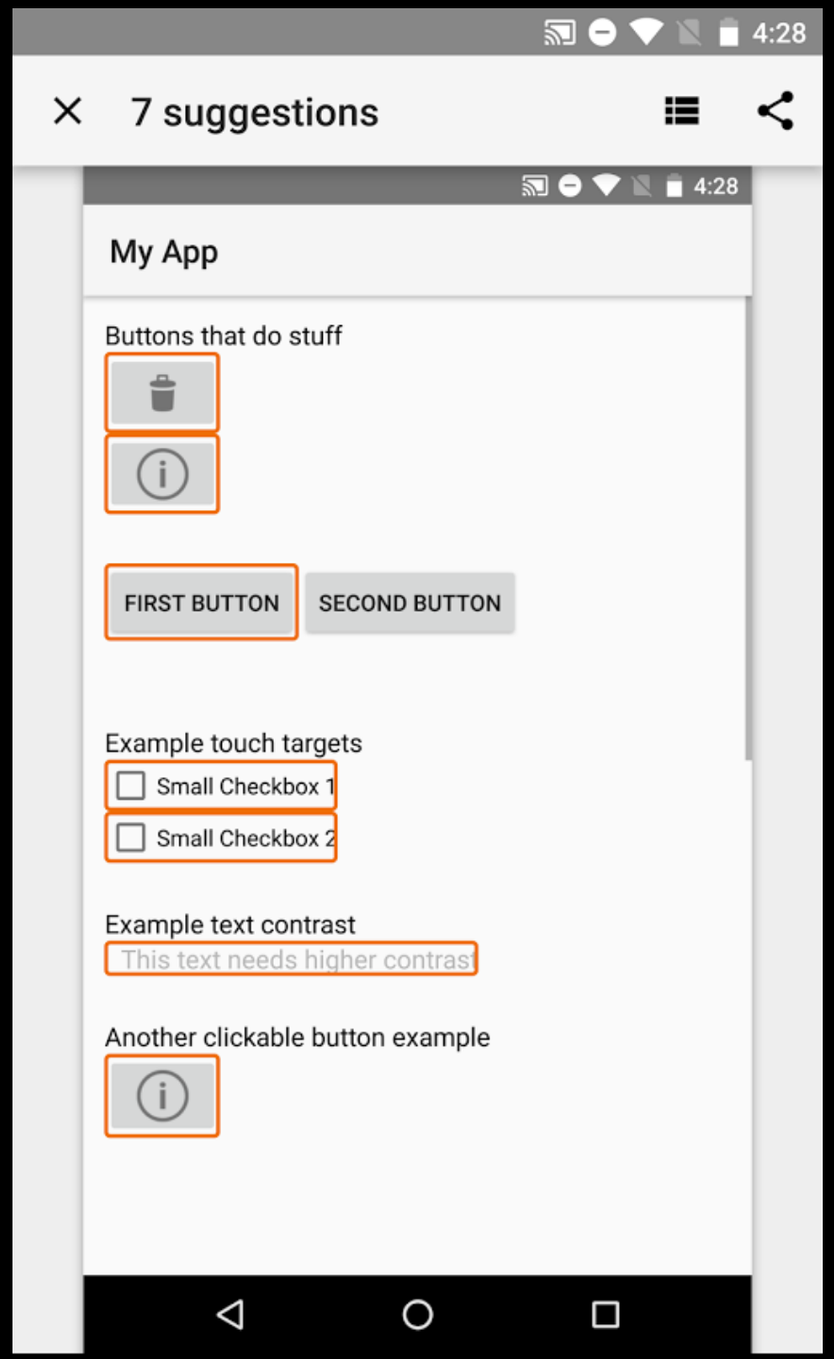


Google Accessibility Scanner

- https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor&hl=en







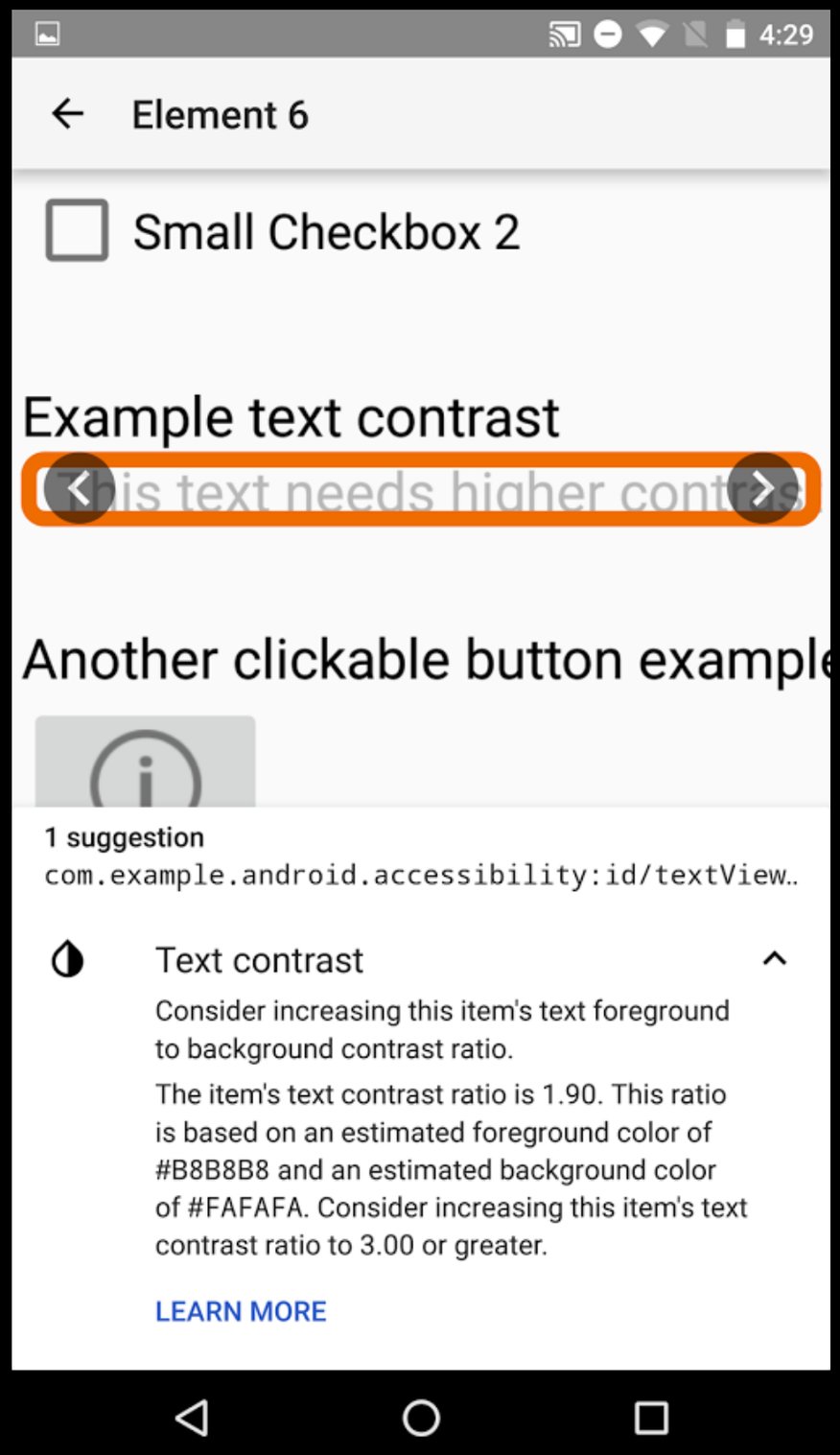
Using Accessibility Scanner

1 Settings > Accessibility
2 Turn on Accessibility Scanner
3 Open the app that you want to scan then tap the Accessibility Scanner button

[Google Accessibility Scanner - Android Apps on Google Play]

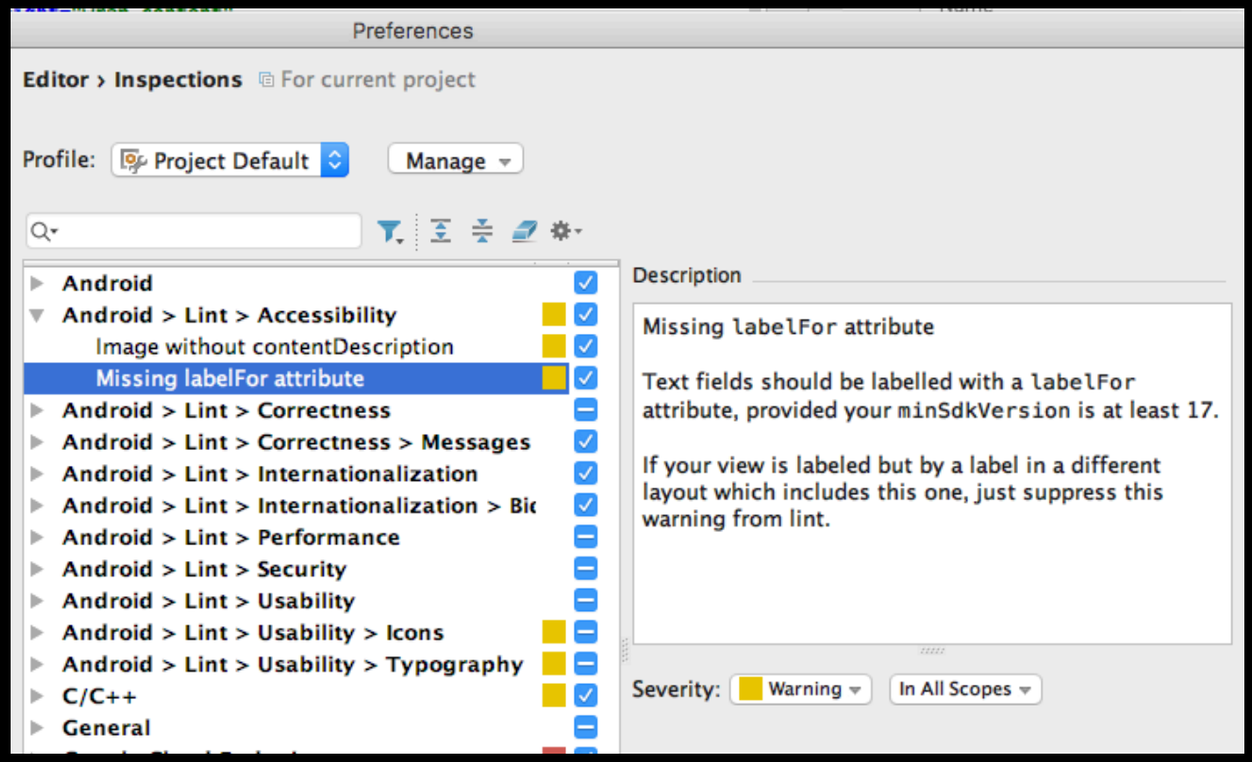
[Get started with Accessibility Scanner]

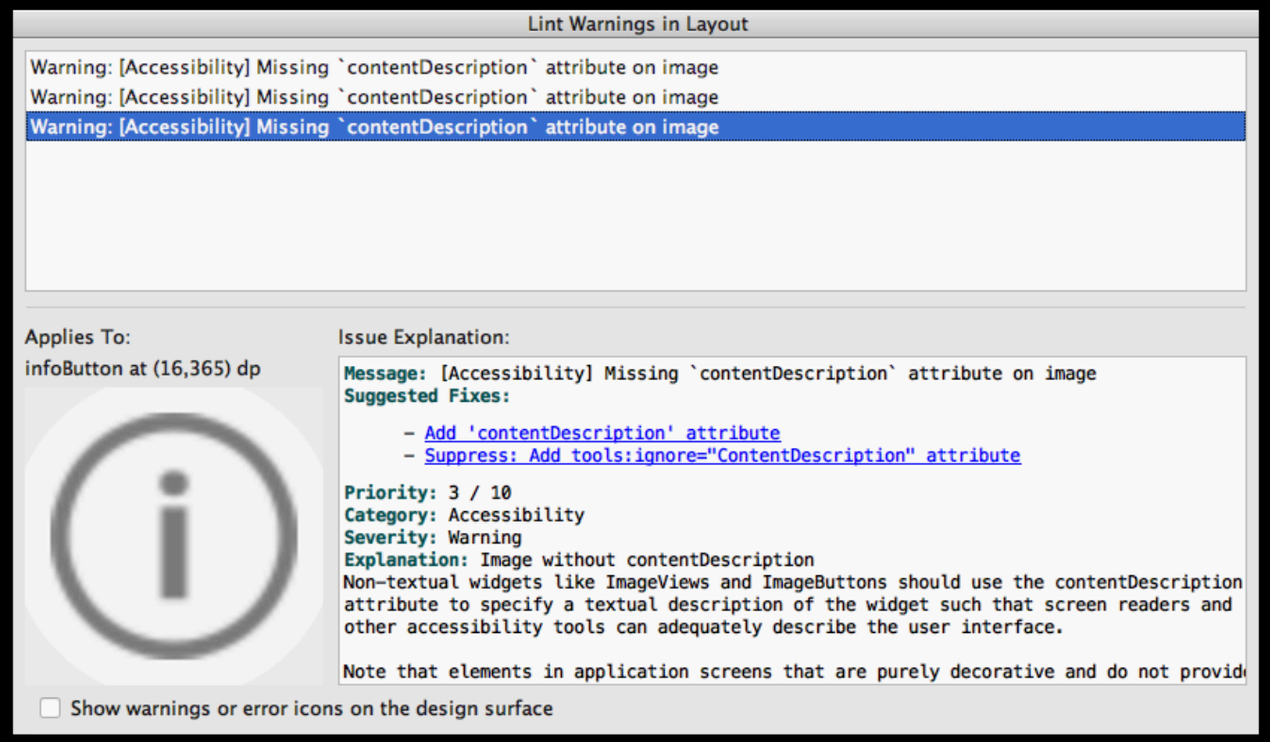
[How to read Accessibility Scanner results]



Android Lint

- Included in Android Studio
- Missing labelFor attribute
  - Text fields should be labelled with a labelFor attribute, provided your minSdkVersion is at least 17.
- Explanation: Image without contentDescription
  - Non-textual widgets like ImageViews and ImageButtons should use the contentDescription attribute to specify a textual description of the widget such that screen readers and other accessibility tools can adequately describe the user interface.

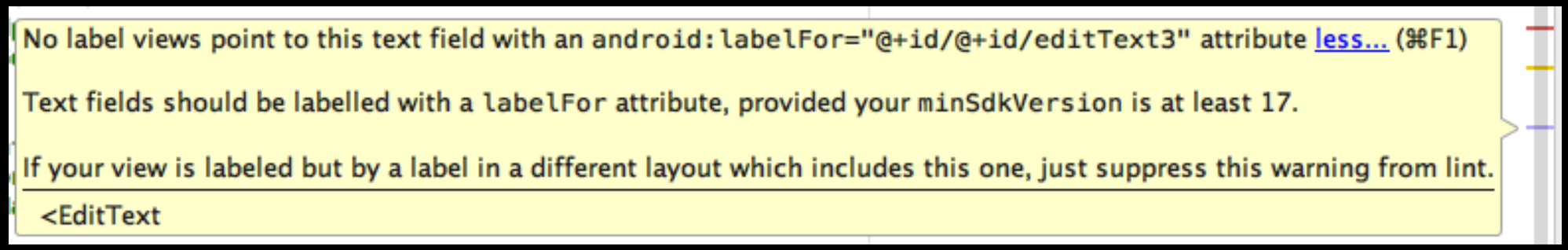


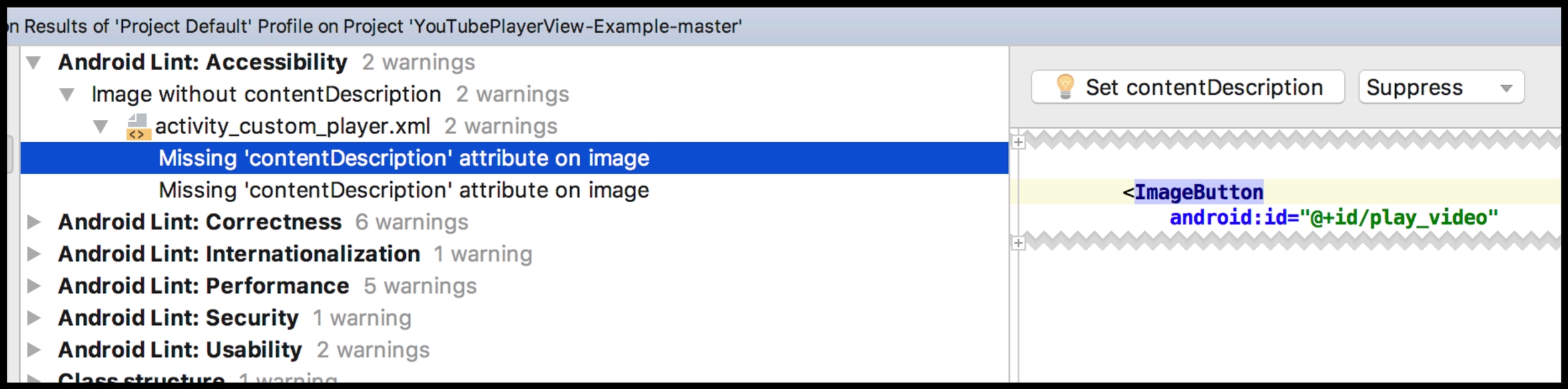


Android Lint Accessibility Tests

Lint misses issues like missing labels if minSdkVersion not 17.

Demo App To Use For Testing
StackTipsLab/YouTubePlayerView-Example





Fixing Accessibility Issues in Design View, XML View, Java Code View

activity_main.xml - FormValidation - [~/Dropbox/AndroidA11y/AndroidStudioProjects/FormValidation]

app

layout > activity_main.xml

activity_main.xml | MainActivity.java

RelativeLayout | TextView

```
47          android:inputType="textPassword" />
48
49      <Button
50          android:id="@+id/button"
51          android:layout_width="wrap_content"
52          android:layout_height="wrap_content"
53          android:text="Login"
54          android:layout_alignTop="@+id/editText2"
55          android:layout_toRightOf="@+id/editText2"
56          android:layout_toEndOf="@+id/editText2" />
57
58      <TextView
59          android:id="@+id/textView3"
60          android:labelFor="@+id/editText3"
61          android:layout_width="wrap_content"
62          android:layout_height="wrap_content"
63          android:text="Username"
64          android:textAppearance="@android:style/TextAppearance.Material.Medium"
65          android:layout_below="@+id/button"
66          android:layout_alignParentLeft="true"
67          android:layout_alignParentStart="true"
68          android:layout_marginTop="28dp" />
69
70      <EditText
71          android:id="@+id/editText3"
72          android:layout_width="wrap_content"
73          android:layout_height="wrap_content"
74          android:layout_alignParentLeft="true"
75          android:layout_alignParentStart="true"
76          android:layout_below="@+id/textView3"
77          android:ems="10"
78          android:inputType="textPersonName" />
79
80      <TextView
81          android:id="@+id/textView4"
```

Design | Text

MainActivity.java - FormValidation - [~/Dropbox/AndroidA11y/AndroidStudioProjects/FormValidation]

app

com > example > pauljadam > formvalidation > MainActivity

activity_main.xml | MainActivity.java

MainActivity | onCreate() | new OnClickListener | onClick()

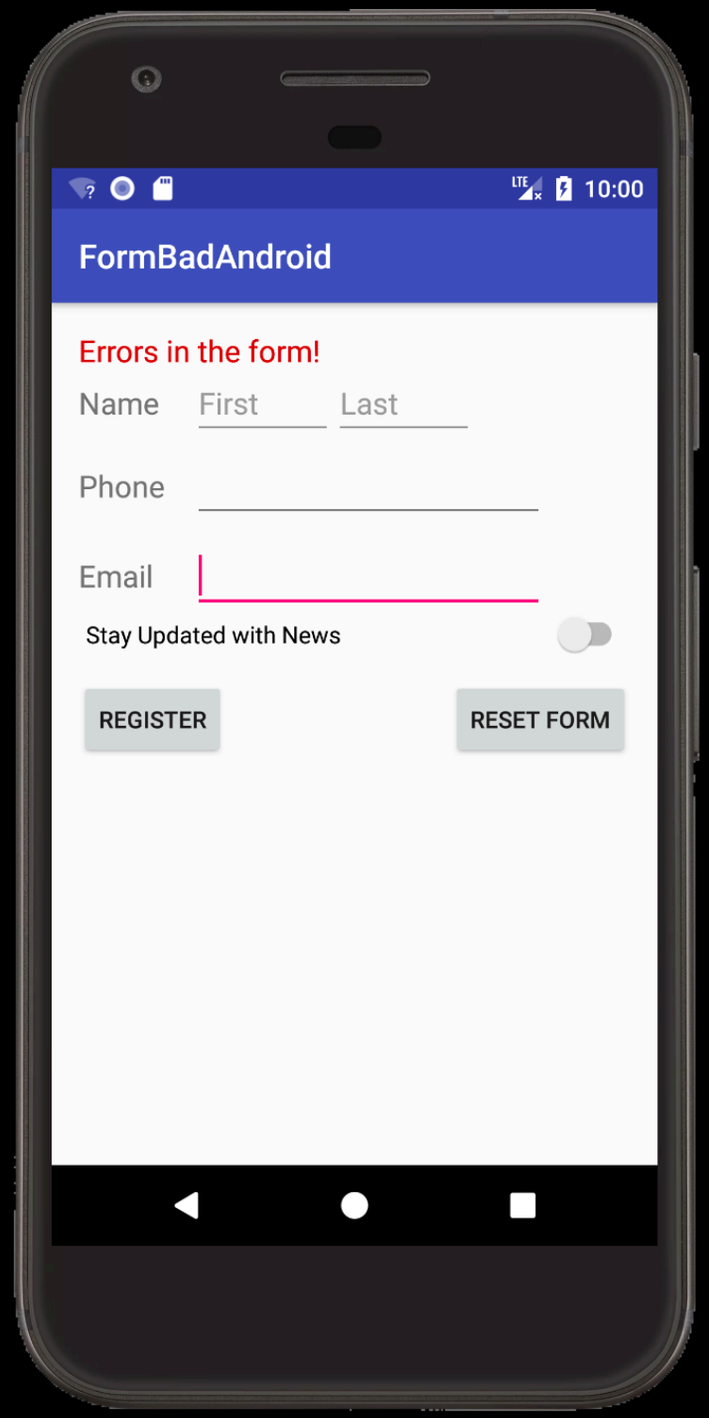
```
34              }
35          });
36          final Button login2 = (Button) findViewById(R.id.button2);
37          final TextView textView = (TextView) findViewById(R.id.textView3);
38          final TextView textView2 = (TextView) findViewById(R.id.textView4);
39          final EditText password = (EditText)findViewById(R.id.editText4);
40          final EditText username = (EditText)findViewById(R.id.editText3);
41
42          login2.setOnClickListener(new View.OnClickListener() {
43              @Override
44              public void onClick(View view) {
45
46                  if( password.getText().toString().length() == 0 ) {
47                      password.setError( "Password is blank! Please enter your password." );
48                      textView2.setContentDescription(textView2.getText() +", Invalid input,
49                      password.requestFocus();
50                      password.sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_FOCUSED);
51                  } else if( username.getText().toString().length() == 0 ) {
52                      username.setError( "Username is blank! Please enter your username." );
53                      textView.setContentDescription(textView.getText() +", Invalid input, "+
54                      username.requestFocus();
55                      username.sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_FOCUSED);
56                  }
57              }
58          });
59
60
61
62
63      }
64  }
65
```

Most Native Controls Get Accessibility For Free!

- iOS & Android e.g. a Button with proper visible text is already accessible by default!
- Most controls are accessible but many need accessibility work.
- E.g. android:text on checkbox

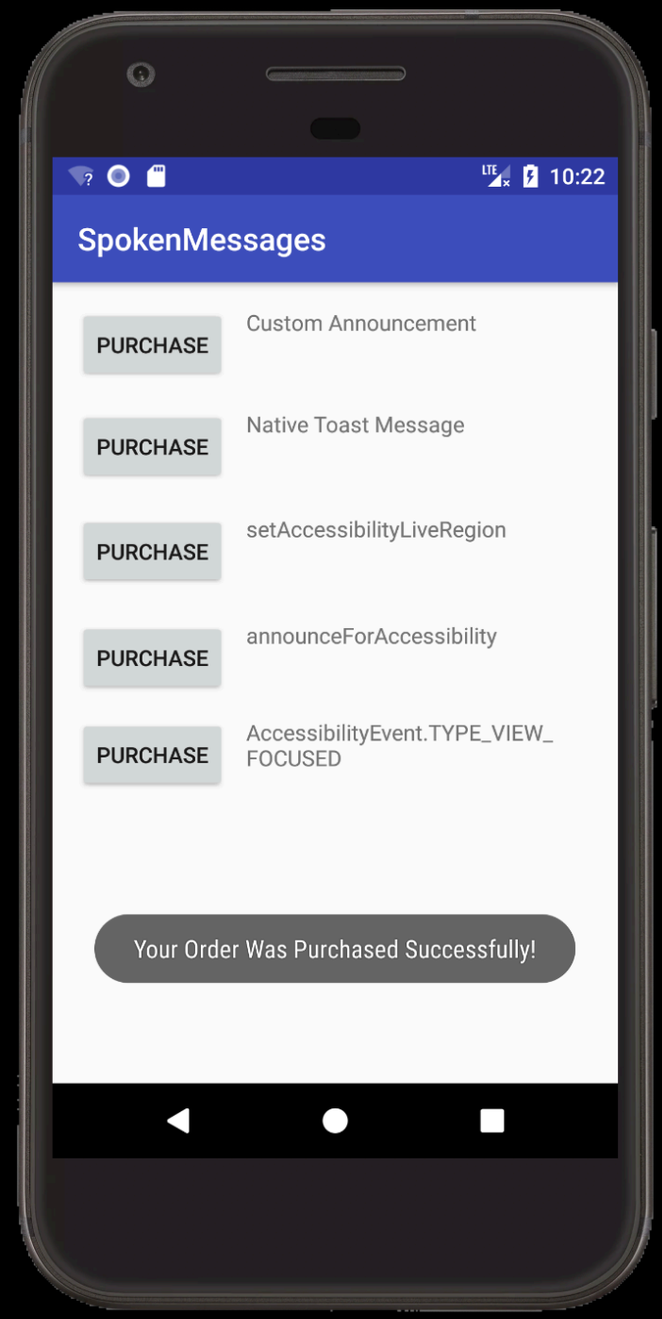
Common Android Accessibility Problems

- Form Labels
- Error Validation
- Focus Management
- Keyboard Operability & Focus Outline
- Unlabelled Icon Buttons
- Missing Role or State
- Low Contrast Text



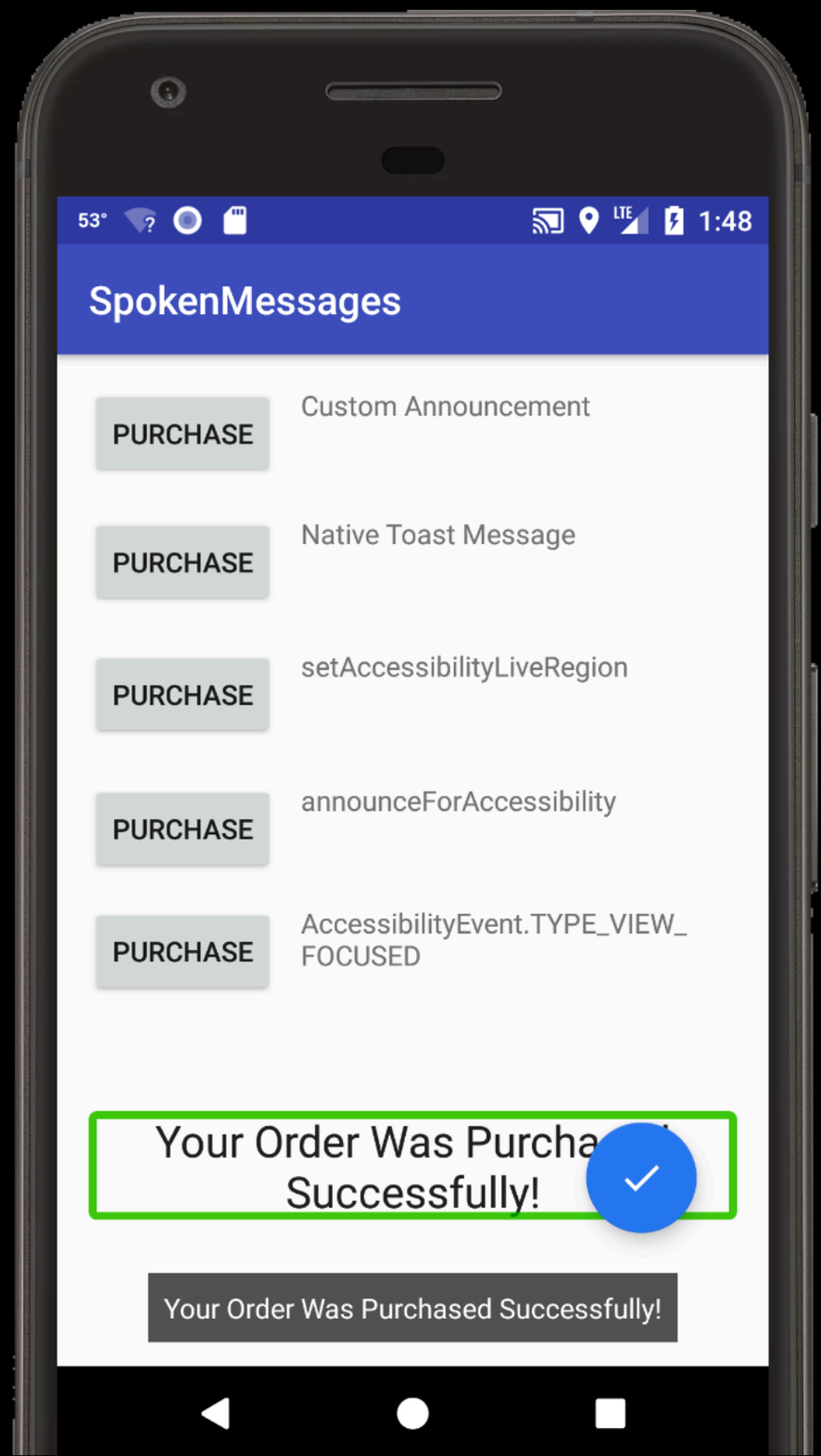
Announcement Notifications

- Live Regions
  - setAccessibilityLiveRegion(View.ACCESSIBILITY_LIVE_REGION_ASSERTIVE)
- .announceForAccessibility
  - announceForAccessibility("Your Order Was Purchased Successfully!")
- Toast Message Spoken Aloud by TalkBack Automatically
  - Toast.makeText(MainActivity.this, "Your Order Was Purchased Successfully!", Toast.LENGTH_SHORT).show();



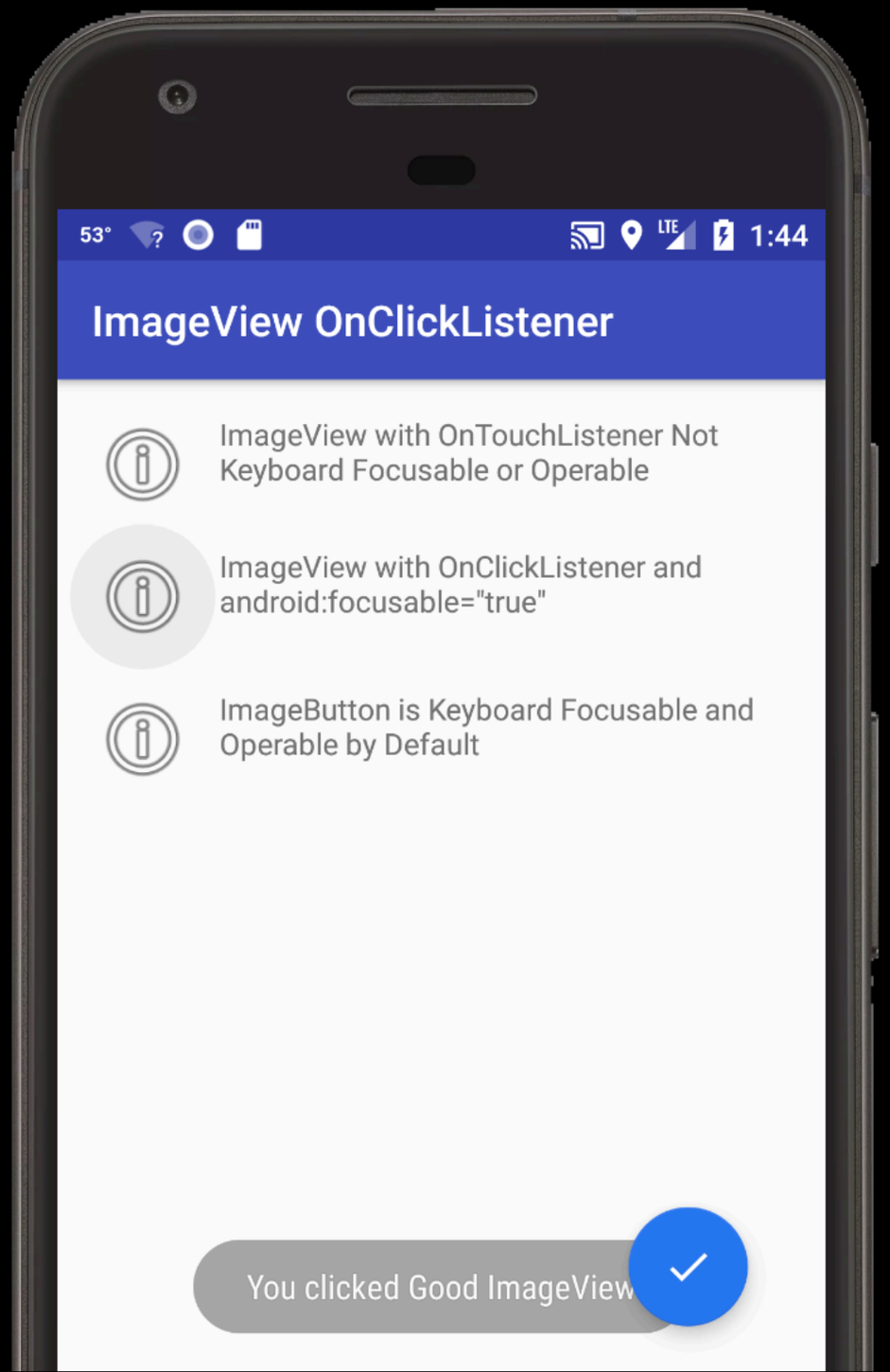
Focus Management

- Move TalkBack screen reader focus
  - sendAccessibilityEvent(AccessibilityEvent.TYPE_VIEW_FOCUSED)



Keyboard Accessibility

- Ensure Keyboard Operability via Bluetooth Keyboard-Only Navigation
- Cannot be tested with the screen reader running.
- Test using Keyboard ONLY!
- Focus visibility required.



Advanced Accessibility Remediation Methods for Android Native

- Using `setLabeledBy` on Android for RadioButton group label like <fieldset><legend>

- https://gist.github.com/renatoi/c496c825445c555ba8e5f1cbe444d531

Activity/Fragment

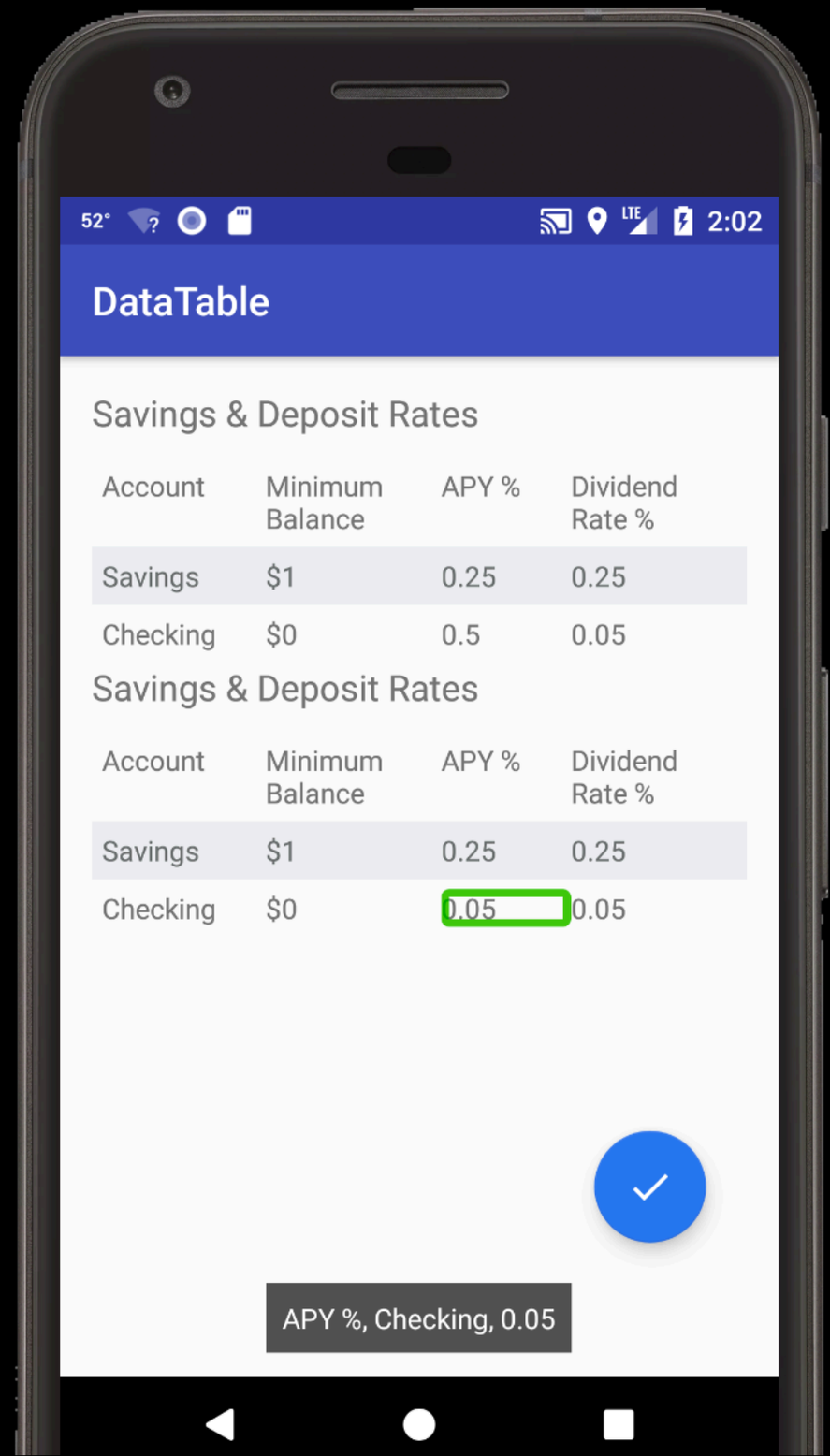
```
// radio buttons
final TextView radioLabel = (TextView) findViewById(R.id.choice_controls_radio_label);
final RadioGroup radioGroup = (RadioGroup) findViewById(R.id.choice_controls_radio_group);

int childCount = radioGroup.getChildCount();
for (int i = 0; i < childCount; i++) {
    View v = radioGroup.getChildAt(i);
    if (v instanceof RadioButton) {
        ViewCompat.setAccessibilityDelegate(v, new AccessibilityDelegateCompat() {
            @Override
            public void onInitializeAccessibilityNodeInfo(View host, AccessibilityNodeInfoCompat info) {
                super.onInitializeAccessibilityNodeInfo(host, info);
                info.setLabeledBy(radioLabel);
            }
        });
    }
}
```

Data Tables in Native?

- There is no row and column headers for accessible data tables in native Android or iOS Accessibility APIs.

- android:contentDescription can be used to prepend row and column header text manually to the accessible names of each cell



Screen Reader Language Output

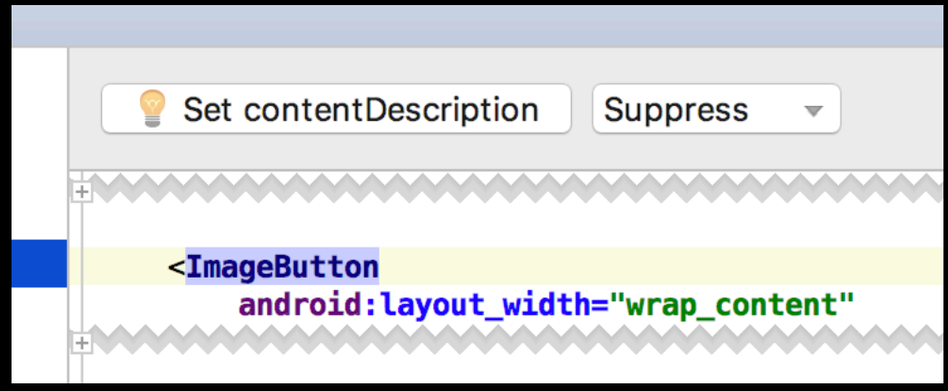
```
TextView localeWrappedTextView = (TextView) findViewById(R.id.textView2);
localeWrappedTextView.setText(wrapTextInLocaleSpan("Bonjour!", Locale.FRANCE));


private SpannableStringBuilder wrapTextInLocaleSpan(
        CharSequence originalText, Locale loc) {
    SpannableStringBuilder myLocaleBuilder =
            new SpannableStringBuilder(originalText);
    myLocaleBuilder.setSpan(new LocaleSpan(loc), 0,
            originalText.length() - 1, 0);
    return myLocaleBuilder;
}
```

https://developer.android.com/guide/topics/ui/accessibility/services.html#multilingual-tts

Test & Fix Bad Image Demo App

- https://github.com/pauljadam/csunmobile



Test & Fix Bad Reading Order Demo App

- android:accessibilityTraversalBefore="@id/button"

Test & Fix Bad Form Demo App

Test & Fix Bad Keyboard Android Demo App

Test & Fix Bad Checkbox Demo App