



# Accessibility in Android world



Nav Singh

Mobile software engineer,  
Manulife



[@navczydev](https://github.com/navczydev)



# Accessibility

It is about making sure that users who have limited vision or other physical impairments can use your application.



# Impact of accessibility

## 1. Increase your app's reach

According to [The World Bank](#), 15% of the world's population has some type of disability. People with disabilities depend on apps and services that support access to communicate, learn, and work.

## 2. Improve your app's versatility

Accessibility can make it easier for all users to interact with your app, not only those with disabilities.



# How Android enables accessibility

One of the main ways that Android enables accessibility is by allowing users to hear spoken feedback that announces the content of user interface components as they interact with applications.



# How to make your application accessible



- **Task flows:** Design well-defined, clear task flows with minimal navigation steps.
- **Action target size:** Make sure buttons and selectable areas are of sufficient size for users to easily touch them, especially for critical actions. How big?.



- **Label user interface controls:** Label user interface components that do not have visible text.
- **Enable focus-based navigation:** Make sure users can navigate your screen layouts using hardware-based or software directional controls (**D-pads, trackballs, and keyboards**).



# Implementation





# Code samples...



```
<ImageButton  
  android:id="@+id/add_note_button"  
  android:src="@drawable/add_note_image"  
  android:contentDescription="@string/add_note_description" />
```



```
<TextView  
  android:focusable="false"  
  android:screenReaderFocusable="true"  
  tools:targetApi="p" />
```



```
<TextView  
  android:accessibilityHeading="true"  
  tools:targetApi="p" />
```



# Code samples...

```
<TextView
  android:id="@+id/tv_hello_world"
  android:nextFocusForward="@id/tv_hello_world_3"
 />

<TextView
  android:id="@+id/tv_hello_world_2"
  android:accessibilityHeading="true"
  android:nextFocusForward="@id/img_decorative"
  android:screenReaderFocusable="true"
  tools:targetApi="p" />

<TextView
  android:id="@+id/tv_hello_world_3"
  android:nextFocusForward="@id/tv_hello_world_2"
  tools:targetApi="p" />

<ImageView
  android:id="@+id/img_decorative"
  android:contentDescription="@null"
 />
```

```
<TextView
  android:id="@+id/usernameLabel" ...
  android:text="@string/username"
  android:labelFor="@+id/usernameEntry" />

<EditText
  android:id="@+id/usernameEntry" />

<TextView
  android:id="@+id/passwordLabel"
  android:text="@string/password"
  android:labelFor="@+id/passwordEntry" />

<EditText
  android:id="@+id/passwordEntry" />
```



# Code samples...



```
<TextView
  android:id="@+id/textToBeChanged"
  android:accessibilityLiveRegion="assertive|none|polite" />
```



```
<LinearLayout
  ...
  orientation="vertical">

  <RelativeLayout
    ...
    android:screenReaderFocusable="true"
    tools:targetApi="p">

    <TextView
      ... />
    <TextView
      ... />
  </RelativeLayout>

  <RelativeLayout
    ...
    android:focusable="true">

    <TextView
      ... />
    <TextView
      ... />
  </RelativeLayout>
</LinearLayout>
```



Test your app's  
accessibility



- **Manual testing:** Interact with your app using Android accessibility services.
- **Testing with analysis tools:** Use tools to discover opportunities to improve your app's accessibility ([Talkback](#), [Accessibility scanner](#))



- **Automated testing:** Turn on accessibility testing in [Espresso](#) and [Robolectric](#).
- **User testing:** Get feedback from real people who interact with your app.
- Pre-launch report on Google Play



# Code samples...

```
init {  
    AccessibilityChecks.enable()  
        .setRunChecksFromRootView(true)  
}
```

```
companion object {  
    @BeforeClass  
    @JvmStatic  
    fun enableAccessibilityChecks() {  
        AccessibilityChecks.enable().apply {  
            setSuppressingResultMatcher(  
                allOf(  
                    matchesCheckNames(`is`("TextContrastViewCheck")),  
                    matchesViews(withId(R.id.textToBeChanged))  
                )  
            ).setRunChecksFromRootView(true)  
        }  
    }  
}
```



Demo







# References

- <https://developer.android.com/guide/topics/ui/accessibility>
- [https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback&hl=en\\_CA&gl=US](https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback&hl=en_CA&gl=US)
- [https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor&hl=en\\_CA&gl=US](https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor&hl=en_CA&gl=US)
- <https://developer.android.com/jetpack/compose/accessibility>
- <https://contrast-ratio.com/#white-on-black>