

Accessibility as a First Class Citizen

Sophie Au (she/they/he)

Frontend-y Fullstack Developer

Token Diversity Person

sophieau.com

@SolviAu



Accessibility in General

Accessibility in React Native

Who is Accessibility for?

Who is Accessibility for?

- Blind / visually-impaired

Who is Accessibility for?

- Blind / visually-impaired
- Deaf / hearing-impaired
- Motor-impaired
- Neurological disorders
- Learning disabilities
- Mental disorders

Who is Accessibility for?

- Blind / visually-impaired
- Deaf / hearing-impaired
- Motor-impaired
- Neurological disorders
- Learning disabilities
- Mental disorders
- Slow / no internet
- Old / slow device
- Small / large device
- Personal sensibilities
- Left-handedness
- Small hands

Why Should We Care?

Why Should We Care?

- It's the right thing to do
- Not being accessible is expensive in the long run

Why Is Bad Accessibility Expensive?

- Risk of getting sued
- Fixing accessibility issues later on takes time
- (Potential) employees/users value accessibility
- Untapped user base (15%+ of world population)

<https://www.who.int/health-topics/disability>

How Do We Become More Accessible?

Build in Accessibility From the Start

Feature Discovery

- Consider accessibility implications
- Estimate with accessibility in mind

Design

- High contrast colors
- Accomodate zoom levels
- Large tap areas
- Loading and error states
- Simple and short flows
- ...

Content

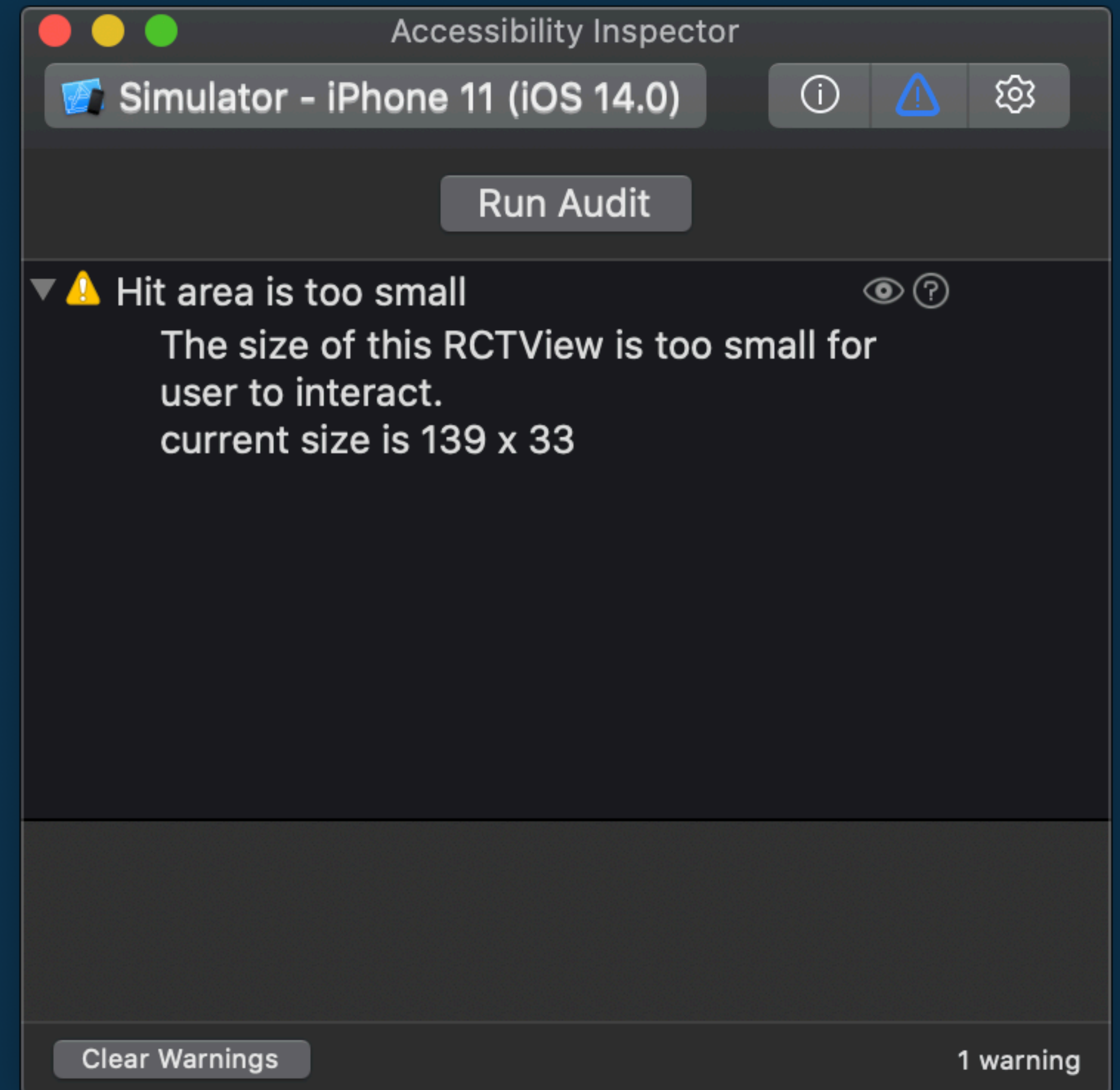
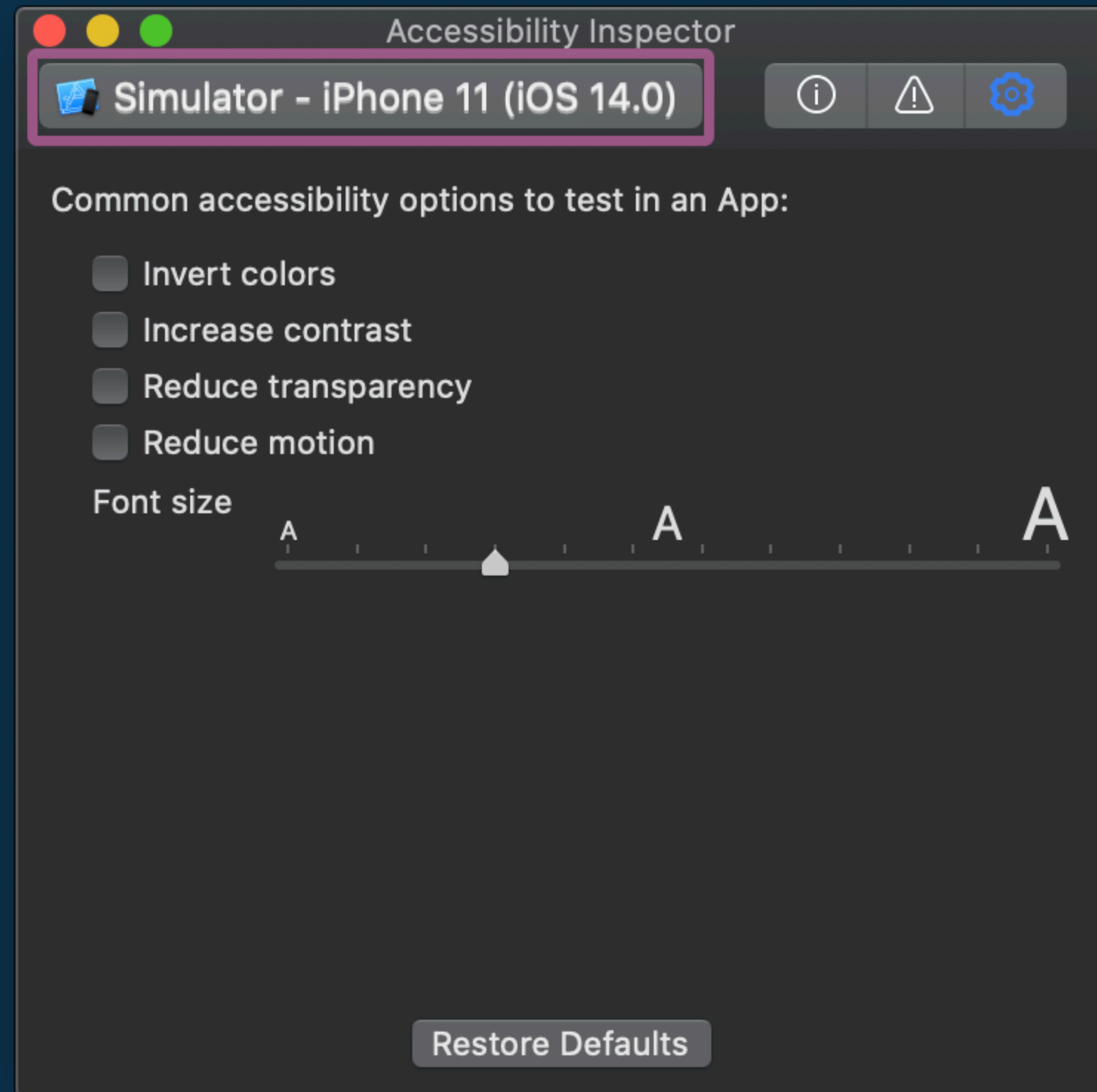
- Adapt language level to target user
- Avoid gendered, religious, offensive language
- Avoid (accidentally) condescending language

Engineering

- Build for screen reader
- Assume low computing power
- Assume slow internet
- Double-check design and content with tooling

React Native Time

Accessibility Inspector



Accessibility Props

- accessible
- accessibilityLabel
- accessibilityHint
- accessibilityRole
- accessibilityState
- accessibilityValue
- onAccessibilityTap
- onAccessibilityEscape
- accessibilityIgnoresInvertColors (iOS only)
- accessibilityViewIsModal (iOS only)
- accessibilityElementsHidden (iOS only)
- onMagicTap (iOS only)
- accessibilityLiveRegion (Android only)
- importantForAccessibility (Android only)

Accessibility Props

- **accessible**
- accessibilityLabel
- accessibilityHint
- accessibilityRole
- accessibilityState
- accessibilityValue
- onAccessibilityTap
- onAccessibilityEscape
- accessibilityIgnoresInvertColors (iOS only)
- accessibilityViewIsModal (iOS only)
- accessibilityElementsHidden (iOS only)
- onMagicTap (iOS only)
- accessibilityLiveRegion (Android only)
- importantForAccessibility (Android only)

Accessible

- Groups all children into a single selectable component
- Label is set by concatenating all text children
- 'Touchable' and 'Pressable' core components are accessible by default

```
<View
  style={[styles.root, !isFound && {opacity: 0.7}]}
  accessible={true}>
  <View>
    <Text style={styles.title}>{title}</Text>
    <Text style={styles.owner}>{owner}</Text>
  </View>
  <Image
    source={keyIcon}
    style={[styles.image, {tintColor: isFound ? '#ffd700' : '#aaaaaa'}]}
  />
</View>
```

AccessibilityLabel

- Describe what the component contains / is used for
- Should be used to rearrange component content, not as a replacement

```
<View
  style={[styles.root, !isFound && {opacity: 0.7}]}
  accessible={true}
  accessibilityLabel={`${title}, held by ${owner}. ${isFound ? 'Retrieved' : 'Not yet retrieved'}`}>
  <View>
    <Text style={styles.title}>{title}</Text>
    <Text style={styles.owner}>{owner}</Text>
  </View>
  <Image
    source={keyIcon}
    style={[styles.image, {tintColor: isFound ? '#ffd700' : '#aaaaaa'}]}
  />
</View>
```


AccessibilityRole

- Describe the role of the component (e.g. button, link, header)
- Need to always be set manually, never set by default

```
• • •  
<TouchableOpacity  
  onPress={onPress}  
  disabled={disabled}  
  accessibilityRole="button">  
  <Text>Refresh Key Status</Text>  
</TouchableOpacity>
```

AccessibilityState

- The state the component is in (e.g. loading, disabled)

```
<TouchableOpacity
  onPress={onPress}
  disabled={disabled}
  accessibilityState={{disabled}}
  accessibilityRole="button">
  <Text>Refresh Key Status</Text>
</TouchableOpacity>
```


Thank You