Quick Heal
Security Simplified

Android Accessibility

A report by Quick Heal Technologies Limited, India

Whitepaper

Authors: Ghanshyam Ahire, Technical Architect
& Sayali Patil, Software Engineer

Quick Heal

Security Simplified

Table Of Contents

Overview

What was once a trend is now a way of life. Mobile is no.1 for information or media consumption, and the mobile app market is growing faster than a beanstalk. With the rise in mobile apps, there is a significant difference in the mobile app user expectations. Ease, Clarity, Security, and Intuitiveness, and more. Accessibility is also an essential part of apps helping people with hearing loss, deafness, and cognitive differences.

Android offers many accessibility features designed for people with disabilities, but the result is better products that can be helpful for everyone. Accessibility APIs were first introduced in Android 1.6 (API level 4). Implementing accessibility features into your app is straightforward. This paper covers how we can use android accessibility service to develop some advanced features and secure mobile devices.

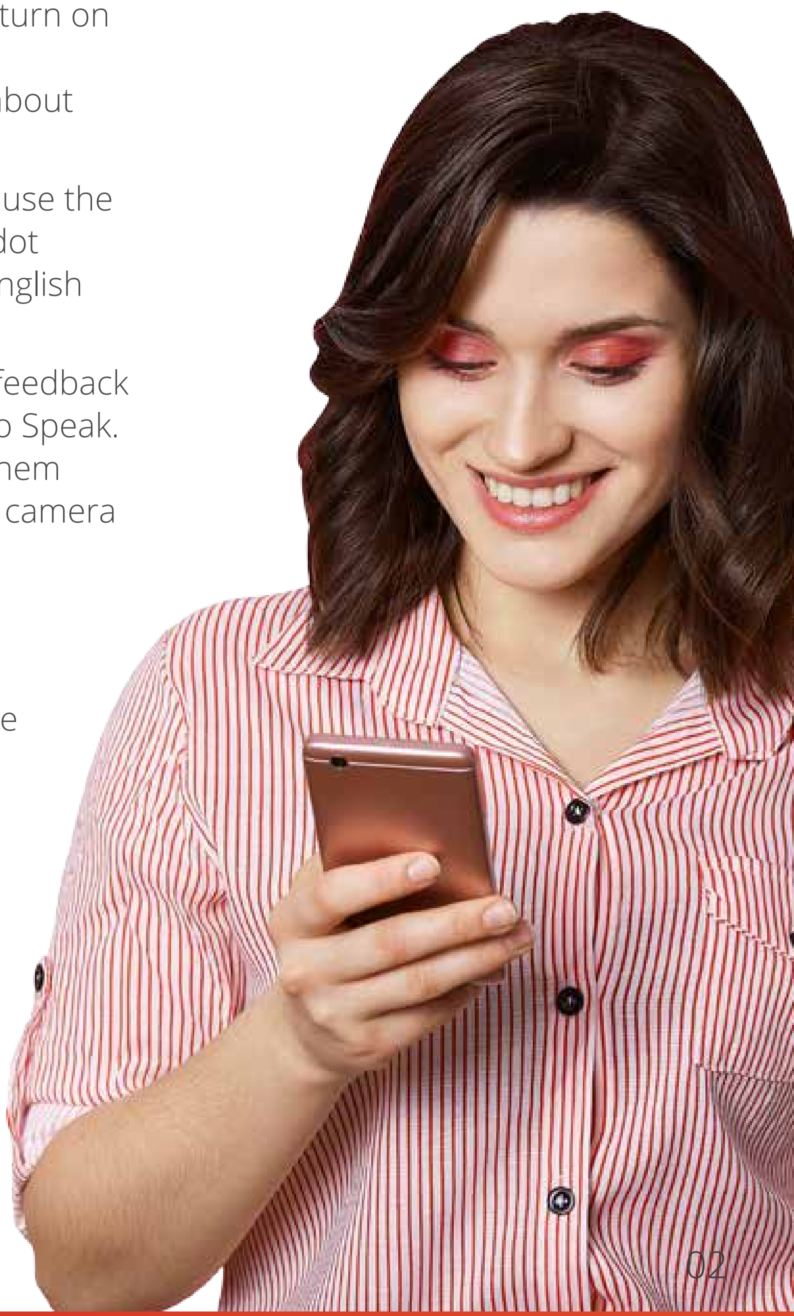
Introduction

Accessibility and assistive technology classify as impactful for people with disabilities. The Android Accessibility Suite is designed to support people to communicate, learn, work, and provide alternative navigation feedback on behalf of applications installed on Android devices.

For example, converting text to speech or providing haptic feedback when a user is hovering on an important screen area. Or, if someone is using an app while cooking, they can use voice commands instead of touch gestures to navigate. Accessibility usages are:

1. Use a screen reader

- **TalkBack**: To interact with your device using touch and spoken feedback, you can turn on the TalkBack screen reader. TalkBack describes your actions and tells you about alerts and notifications.
- **TalkBack braille keyboard**: You can use the TalkBack braille keyboard to enter 6-dot braille on your screen. Only Unified English Braille is currently supported.
- **Select to Speak**: If you want spoken feedback only at certain times, you can select to Speak. Select items on your screen to hear them read or described aloud, or point the camera at something in the real world.



2. Change your display

- **Display size and font-size**: To change the size of items on your screen, adjust the display size or font size.
- **Magnification**: To temporarily zoom or magnify your screen, use magnification.
- **Contrast and color options**: To adjust contrast or colors, use high-contrast text, dark theme, color inversion, or color correction.

3. Interaction controls

- **Lookout**: Lookout uses computer vision to assist people who are blind or have a low idea in gaining information about their surroundings.
- **Voice Access**: Voice Access lets you control your device with spoken commands. Use your voice to open apps, navigate, and edit text hands-free.
- **Switch Access**: Switch Access lets you interact with your Android device using one or more switches instead of the touchscreen. You can use a switch or keyboard to control your device.
- **Action Blocks**: Action Blocks makes routine actions easier with customizable buttons on your Android home screen.
- **Time to take action (Accessibility timeout)**: You can choose the duration of the message visibility that ask you to take action.

4. Use a braille display

- **BrailleBack**: You can connect a refreshable braille display to your device via Bluetooth. BrailleBack works with TalkBack for a combined speech and braille experience, allowing you to edit text and interact with your device.

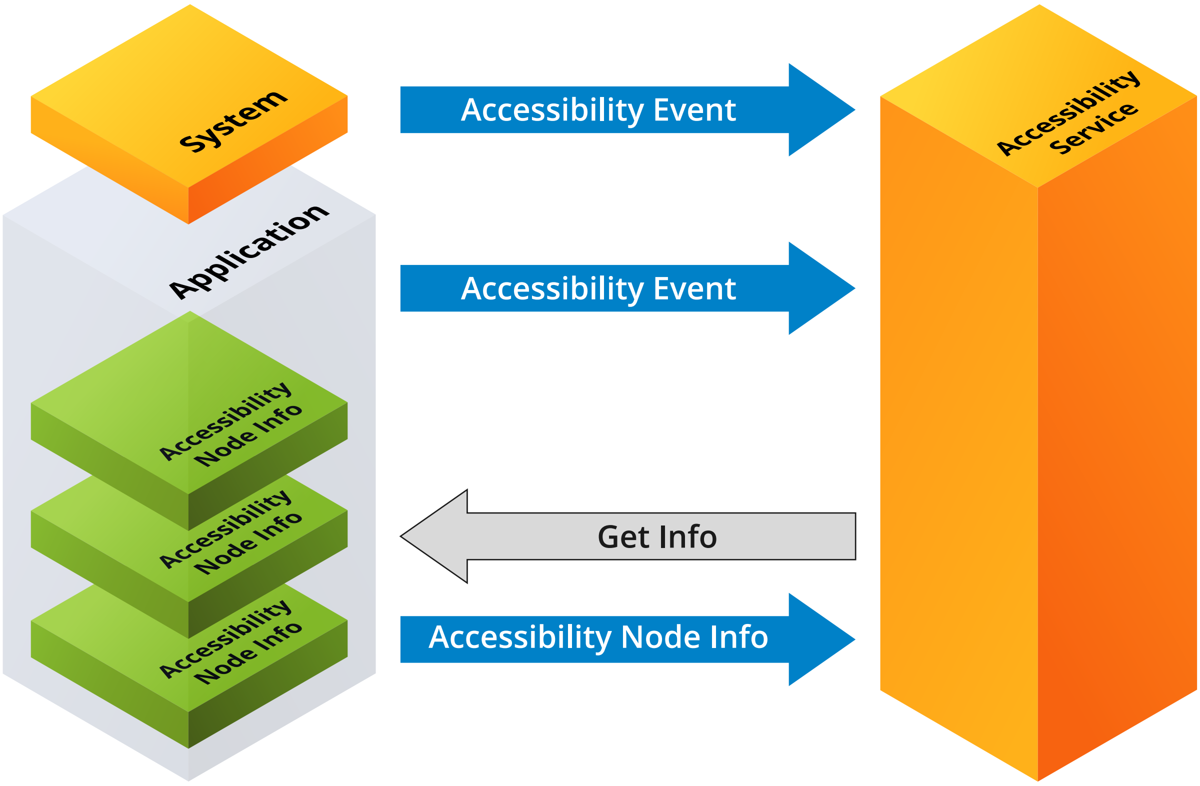
5. Audio & on-screen text

- **Captions**: You can choose caption preferences (language, text, and style) for your device.
- **Live Caption**: Live Caption automatically captions speech on your device.
- **Live Transcribe & Sound Notifications**: You can use Live Transcribe to capture speech and sound and see them as text on your screen. Sound Notifications help you know what's happening in your home, like when a smoke alarm beeps or a doorbell rings.
- **Sound Amplifier**: You can use a Sound Amplifier with wired or Bluetooth headphones to filter, augment, and amplify the sounds in your environment or on your Android device.
- **Hearing aid support**: You can pair hearing aids with your Android device to hear more clearly.
- **Real-time text (RTT) during calls**: You can use text to communicate during a phone call with RTT.

Lifecycle and working

The lifecycle of an accessibility service is managed exclusively by the android system and follows the established service life cycle. After the system binds to a service, it calls AccessibilityService#onServiceConnected(). This method can be overridden by clients/Apps that want to perform post binding setup.

An accessibility service stops either when the user turns it off in device settings or when it calls AccessibilityService#disableSelf().



OnServiceConnected:

This function is called when the user switches on the Accessibility Service for our app from the phone settings. Also, in this Function, we specify the type of events that can be handled by accessibility service.

onAccessibilityEvent(AccessibilityEvent event)

This function is called whenever there is an event on the screen. We mention these events under @xml/service config or the ones we say in the nonservice-connected Function.

FAQs:

This function is called when the user switches on the Accessibility Service for our app from the phone settings. Also, in this Function, we specify the type of events that can be handled by accessibility service.

onAccessibilityEvent(AccessibilityEvent event)

This function is called whenever there is an event on the screen. We mention these events under @xml/service config or the ones we say in the nonservice-connected Function.

1. What are the types of events that occur on the screen?

The events that occur on screen are:

TYPE_VIEW_LONG_CLICKED, TYPE_VIEW_SCROLLED, TYPE_VIEW_SELECTED, TYPE_VIEW_TEXT_CHANGED, TYPE_WINDOWS_CHANGED, TYPE_WINDOW_CONTENT_CHANGED, TYPE_WINDOW_STATE_CHANGED, TYPE_VIEW_CLICKED, TYPE_NOTIFICATION_STATE_CHANGED
and as such many more events

2. How do we read these events from our onAccessibilityEvent?

Well, the parameter of AccessibilityEvent type passed to the function onAccessibilityEvent contains information about the AccessibilityEvent. We can retrieve this information using the function getEventType, which returns a variety of events occurring.

3. What are the type of actions that we can perform user's behalf using AccessibilityService?

The event that we can perform on user's behalf include

ACTION_SET_TEXT, ACTION_CLICK, ACTION_LONG_CLICK, ACTION_CUT, ACTION_COPY, ACTION_SCROLL_FORWARD, ACTION_SCROLL_BACKWARD, ACTION_SELECT.

4. How do we perform these actions using AccessibilityService?

Before understanding this, we need to know how we parse through the XML layout. For these, we need to understand the concept of nodes. All XML UI's contains a hierarchy of views and layouts. The basic Layout can be considered a node, and all the sub-views or sub-layouts can be regarded as its child's. Similarly, each of these sub-views or sub-layouts, in turn, contain their child's this creates a hierarchy which can be used to parse the XML.

5. How Accessibility does returns these XML elements?

For accessing the XML we need to have the parent node which represents the layout of the current window. This can be stored in an AccessibilityNodeInfo Object. Using, the following code:

AccessibilityNodeInfo currentNode=getRootInActiveWindow();

Now this currentNode object contains the parent layout and we can get it's child using the function getChild(<index>). Several operations that can be performed on this currentNode object or it's childs are: getChildCount(), getClassName(), getContentDescription(), getText(), getParent(), getInputType().

6. How to perform actions on users behalf?

Now since we know that the object currentNode holds a reference to our views and layouts. We can use this object to perform actions, and for doing this, we call the function to perform step (<type of action>) on the current node object.

How to leverage security features

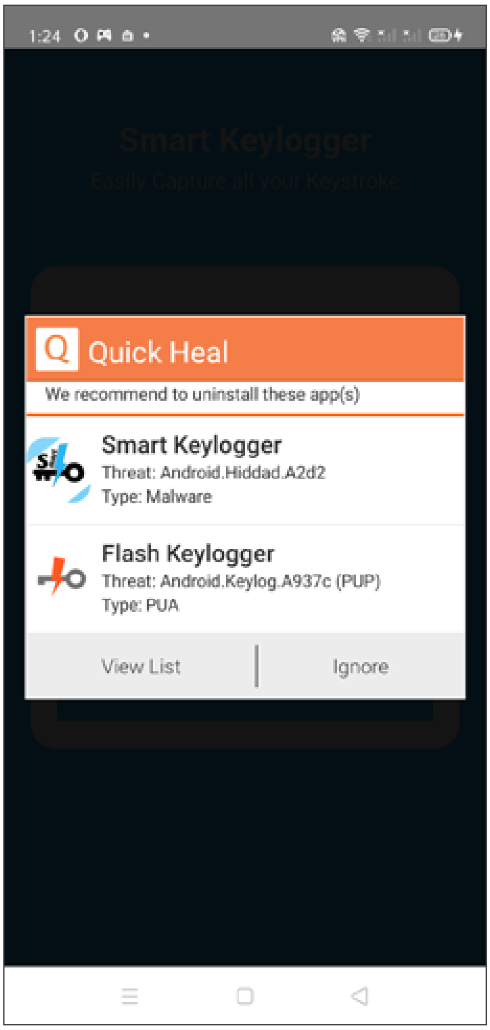
Following are some features are implemented for security purpose using accessibility -

a) Anti-key logger

Anti-Key logger monitors behavior of all currently ongoing key actions on mobile device and allows you to prevent if any application try to capture Key logging.

Implementation Details

- Enable the accessibility service
- Add observer for foreground keyboard app
- If keyboard in use, verify is any app enabled for KEY_TYPE event or using KEY_TYPE event
- If yes then prompt user for keylogger



Smart Keylogger

Easily Capture all your Keystroke

Quick Heal

We recommend to uninstall these app(s)

Smart Keylogger
Threat: Android.Hiddad.A2d2
Type: Malware

Flash Keylogger
Threat: Android.Keylog.A937c (PUP)
Type: PUA

b) Scan Before Download Scan

Automatically scans apps even before you download them from the Google Play Store. It also informs you whether an app is safe to install or not. If it's not, the feature displays the risks of installing the app.

c) Parental Control & Web Security

The Parental Control feature helps you block malicious and potentially dangerous websites. With this feature, you can block a particular website's URL or websites based on their category, such as adult, gambling, violence, etc. Offers multi-browser support for enhanced Web Security. Enable this feature to block infected, malicious, fraudulent, and phishing websites.

Conclusion

Accessibility removes a barrier to print, audio, and visual media that can be much more easily overcome through web technologies. Accessibility empowered peoples have a disability and trying to make day-to-day life easier. At various mobile as well web application accessibility verify all web content are reachable to all people without any barrier and mobile device should be secure for everyone.