

axe DevTools Mobile Release Notes

February 16, 2023

Component Versions

- axeDevToolsUIKit v3.3.0
- axeDevToolsXCUI v2.3.0
- axe-devtools-android v3.1.0

New Features

Analytics

We have incorporated analytics into axe DevTools Mobile using Amplitude so that we can gather data on how users engage with the tool. This will allow us to make more informed decisions on the future development of the product. The implementation of these analytics adheres to the [Deque Privacy Policy](#), personally identifiable information will not be collected, and the data is anonymized. You will notice a banner in the lower left corner of the axe DevTools Mobile Dashboard that allows you to opt in or out of the analytics tracking and associated browser cookies.

Usage Service

axe DevTools Mobile is launching Usage Service support allowing you to track adoption within your organization. Usage Service is an opt-in model and is disabled by default. If you would like to start collecting usage metrics, check out the guides to get started on iOS or Android.

Color Contrast Vision Enhancement for iOS

This release contains some exciting updates to the Color Contrast rule for the axeDevToolsUIKit framework. We are now leveraging [Apple's Vision Framework](#) to improve our detection of text color as well as enable the rule to work on images of text. We expect these enhancements to increase the accuracy of the Color Contrast rule and allow the rule to detect issues that it previously couldn't.

Android Dialog Scanning

Up until now, there was a limitation for scanning dialogs with axe-devtools-android. That's no longer the case because this release contains a new API for both manual and automated testing of XML layouts that enables scanning a dialog. Please see the documentation for [instructions on how to scan dialogs](#) with this new method. [21547, 17303, 20497]

Updates

axe DevTools Mobile Dashboard

We've made some updates to the high level scan details page to make it more valuable. The breakdown charts are still available, but we've changed the section below it to "Get More Out of Your Results". This now displays which rules are detecting Critical issues to help you prioritize fixing those first. It also provides a link to documentation for users who may be viewing a scan for the first time and want more information on how axe DevTools Mobile works.

axe DevTools Mobile for Android

In v2.1.0 of axe-devtools-android released in July 2022, we decided to mark results from the Hidden Active View Focus and Hidden Informative View Focus rules as "IGNORED" by default. After working to reduce false positives generated by those rules, we have reinstated them to report issues as normal in this release. We still think there is further room for improvement of these rules, so they are considered experimental, but due to a known issue, they will not show up as such in the dashboard. This will be resolved in an upcoming dashboard release, but the rules can be disabled via the Custom Rulesets or Ignore Rules features if you wish not to see results from them.

Bug Fixes

axe DevTools Mobile for iOS

- Fix an issue that was causing isVisibleToUser to be true for controls behind a modal in some cases [21547]
- Reduce false positive in the Inaccessible Action Rule by making UIDimmableViews inapplicable
- Add support for Dark Mode on the login screen
- Reduce false positives for several rules by considering 'toolbar' a header/footer
- Update the boundsInScreen for Inaccessible Action results to be that of the control that can't be focused instead of the parent view that is focusable
- Reduce false positive in the Inaccessible Action rule related to TableCells
- Fix an issue causing the keyboard to cover the API key text field on the login screen

axe DevTools Mobile for Android

- Once a scan has been successfully pushed to the dashboard, the link to that scan will now be printed to the logs

- Fixed an issue that caused the FAB to be positioned offscreen to the left on API 33
- The FAB will no longer cause a crash while TalkBack is running [20453]

axe DevTools Mobile Dashboard

- Truncate long scan names that would overflow into the issues column in the scans table
- Ensure that the focus outline for nodes in the Inspect hierarchy is not cut off or obstructed
- Add hover states to the controls that display issues grouped by rule, so it is clear that they are buttons
- Make the filtering by tags feature more clear that you can add multiple tags to the list [#17958]
- The copy button in the share modal now changes to a checkmark to indicate to the user that the link was successfully copied
- The user dropdown menu no longer requires scrolling to get to the “Sign out” option
- Fixed an issue with URI encoding that was affecting the “Copy Link to Issue” button and share modal link copying for apps that have spaces in the package name
- Add external link icons to the help modal because the documentation links are set to open in a new tab or window
- Changed the “Issues Found” heading to “Results” so that it makes more sense in the context of viewing the results that have a status other than “FAIL”
- Remove the “more details” expand/collapse button in the issues, so users can get to the full contents of an issue quicker with fewer clicks
- Update the icons for the OS column of the table and scan detail pages to be more modern

Known Issues

- axe DevTools Mobile only runs on native iOS and Android views, so hybrid or code once platforms such as Cordova, Xamarin, and Flutter are not supported.
- Rules will not run in web views or rendered PDFs. For testing against HTML content, please utilize axe DevTools HTML.

axe DevTools Mobile Dashboard

- The table of scans height is no longer adjustable. This will be fixed in a future release. The width of the table remains adjustable.

axe DevTools Mobile Desktop App

- Unable to connect to the database on Java 14+, please continue to use an earlier version of Java.

axe DevTools Mobile for iOS

axeDevToolsUIKit.xcframework

- Views built with SwiftUI cannot be tested using this framework, use the axeDevToolsXCUI framework in UITesting for SwiftUI views
- Custom rule support is available only in Swift, the custom rule API is not open to Objective-C.
- The "Visible to User" filter may not work properly on side menus and alerts. We are actively working on a fix and are expecting it in the next release.
- We are working on fixes for the following false positives and will update this list as fixes are released:
 - Currently the CollidingViews rule utilizes the size of the UILabel frame. When the frame is larger than the visible text, it may incorrectly flag text as colliding with another element on the screen.
 - InScrollView may report issues for text within banner-behaving elements. To make these elements available to those that require larger text, use UILargeContentViewer.
 - You may see SupportsDynamicType false positives in the following scenarios:
 - If using the UIContentSizeCategoryDidChange notification, the framework may incorrectly report that text does not respond to Dynamic Type in UITests.
 - If using NotificationCenter.default.addObserver(forName:..., object:..., queue:..., using:...) to respond to the UIContentSizeCategoryDidChange notification.
 - If using traitCollectionDidChange on a text's parent view or view controller to respond to Dynamic Type.

axeDevToolsXCUI.xcframework (SwiftUI framework)

- Picker views in SwiftUI and UIKit may cause the framework to get stuck on simulators or devices.
- The "Visible to User" filter may not work properly on side menus and alerts. We are actively working on a fix and are expecting it in the next release.
- Landscape orientation is not formally supported at this time, this will be addressed in a later version.

- We are actively working on fixes for the following false positives and will update this list as fixes are released:
 - For views that are off-screen, the A11yElementFocusBox rule may incorrectly report the VoiceOver focus box is not encapsulating the view.
 - Currently the CollidingViews rule utilizes the size of the UILabel frame. When the frame is larger than the visible text, it may incorrectly flag text as colliding with another element on the screen.
 - InScrollView may report issues for text within banner-behaving elements. To make these elements available to those that require larger text, use UILabelContentView.
 - Results for LabelAtFront may not be correct on picker elements in iOS 16+

axe DevTools Mobile for Android

- If your debug or test build is utilizing Proguard, follow the steps to [ignore Deque in your Proguard settings](#).
- A build with r8 enabled may attempt to minify the axeDevTools library resulting in an error similar to:

Caused by: java.lang.NullPointerException: throw with null exception
 at g.b.b.a\$a.a(Unknown Source:1)
 at g.b.b.a\$a.a(Unknown Source:0)
 at g.b.b.a.a(AccessToken.java:190)

To resolve this error add the following line to your proguard file to keep axeDevTools classes: `keep class com.deque.** { *; }`

- If you encounter an error along the lines of `Expected exactly '1' node but found '2' nodes that satisfy: (isRoot)`, please contact us at helpdesk@deque.com for assistance. Under certain conditions, there may be two Compose root nodes existing at the same time.
- axe-devtools-android v3.1.0 is compatible with pure java8 projects, so you may see errors while compiling the library. To resolve, add the following imports:

```
implementation 'androidx.core:core-ktx:1.9.0'
implementation 'org.jetbrains.kotlin:kotlinx-coroutines-core:1.6.4'
implementation 'org.jetbrains.kotlin:kotlinx-coroutines-android:1.6.4'
```

If you see errors related to minSDK version for core-ktx library, try adding `<uses-sdk tools:overrideLibrary="androidx.core" />` to you project's Android Manifest