

axe DevTools Mobile Release Notes

June 6, 2022

Component Versions

- axeDevToolsUIKit v3.0.0 (previously Attest-iOS)
- axeDevToolsXCUI v2.0.0
- axe-devtools-android v2.0.1
- axe DevTools Mobile desktop app v0.25.0

New Features

API Refactor

The Android library and iOS frameworks API interfaces are now more intuitive and user-friendly. These updates also remove all remaining references to the previous Attest branding. Please see the updated documentation for [iOS](#) and [Android](#) to start interfacing with the new API today.

API Key Authentication

API key authentication is now fully supported for sending scans directly from the native mobile libraries, offering seamless integration between team members and their CI/CD pipeline(s). Signing in with a username and password remains available as well. Follow the getting started guides for [iOS](#) or [Android](#) to learn how to use this new authentication method. Manage your API keys from your [axe Account page](#).

Refreshed Documentation

Axe DevTools Mobile documentation is now available with our other products at docs.deque.com. Every page was updated and reorganized to offer the utmost clarity in exploring all that axe DevTools Mobile offers. The site is publically available, you no longer have to sign in to your axe Account to view it. This new content is specifically for the refactored Android and iOS APIs. Documentation for previous versions of axe DevTools Mobile will remain available at mobile-docs.deque.com.

New iOS Rules

We've expanded testing coverage for iOS. The [Inaccessible Action rule](#), available in axeDevToolsUIKit, covers WCAG 2.1.1 by testing that any active control has an action available for assistive technologies to activate. The [Meaningful Accessible Name rule](#),

included in both iOS frameworks, is a platform guideline that ensures an element's accessibility label or name does not include its accessibility trait.

Color Contrast Rule Improvements

The color contrast rule now runs locally within axeDevToolsUIKit for iOS and axeDevTools for Android to capture platform-specific views, which improves accuracy. These improvements will reduce both false positives and false negatives.

Note: This may mean seeing new color contrast issues that previously weren't detected.

Public Distribution

The iOS frameworks and Android library are now available through public distribution platforms for each ecosystem. In addition to Agora, you can now pull in the iOS frameworks [using Swift Package Manager](#) and the Android library [from Maven Central](#).

Sample Apps

We've published sample applications that demonstrate integrating axe DevTools Mobile into a project that has existing accessibility issues. They are available as public GitHub repos for [iOS](#) and [Android](#).

Bug Fixes

axe DevTools Mobile for iOS

- Resolve issue with makeKeyAndVisible in UITests [CSD-16882]
- Properly retrieve screenTitle for iOS 15
- Remove "clipped" UIKit views from view hierarchy

axe DevTools Mobile for Android

- Prevent crashing while serializing AxelImage
- Enable compatibility with Java 8

axe DevTools Mobile Dashboard and Desktop App

- Fixed a bug where in some circumstances, the issue grouping by rule was including issues from multiple rules
- Update icons to not make requests to a third party [CSD-18096] note some icons may still be missing and will be fixed in an upcoming release

Known Issues

- axe DevTools Mobile only runs on native iOS and Android views, hybrid 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 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 (previously Attest-iOS.xcframework)

- Banner style views report InScrollView issues, use the ignore rule feature to bypass this rule for known banner views that you know should not be within a scrollable view.
- 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.

axeDevToolsXCUI.xcframework (SwiftUI framework)

- SwiftUI Picker scans may get stuck on simulators or a device running iOS versions lower than 15.0.
- A11yElementFocusBox may show false positives with views partially off-screen, this will be addressed in a later version
- The "Visible to User" filter may not work on views behind custom-built modals, help make this filter more robust by providing an example snippet of how the modal is implemented helpdesk@deque.com
- Landscape orientation is not formally supported at this time, this will be addressed in a later version
- Banner style views report InScrollView issues, use the ignore rule feature to bypass this rule for known banner views that you know should not be within a scrollable view