

# axe DevTools Mobile Release Notes

November 17, 2022

## Component Versions

- axeDevToolsUIKit v3.2.0
- axeDevToolsXCUI v2.2.0
- axe-devtools-android v3.0.0
- axe DevTools Mobile desktop app v0.26.0

## New Features

### axe DevTools Mobile Dashboard and Desktop App

#### Experimental Rules

This release introduces a new concept of experimental rules that we are using to help better refine rules. The results from experimental rules are considered beta and are clearly distinguished from the rest of results in the dashboard. If you suspect inaccurate results, please [report a bug](#) and be sure to include the JSON file for the scan. If you wish not to see results from experimental rules, they can be [filtered out from the settings menu](#).



The screenshot displays the axe DevTools Mobile dashboard interface. At the top, there is a header 'Issues Found' with a refresh icon and a close button. Below this, there is a section for 'Issue Status' with a dropdown menu currently set to 'FAIL (3)'. The main content area is divided into two sections: 'Regular Results' and 'Experimental Results'. The 'Regular Results' section shows a single issue: 'FocusableText' with '1 found'. The 'Experimental Results' section shows two issues: 'InaccessibleAction - experimental' with '2 found'. On the right side of the dashboard, there is a settings panel titled 'Experimental Rules' which is highlighted with a red border. This panel contains the text 'Show results for rules that are considered experimental' and two checked checkboxes: 'iOS' and 'Android'.

Please note that this feature is not supported in the desktop app. Results from experimental rules will be treated the same as existing results. Since filtering is not available, if you wish to not see these results, you can use the ignore rules feature on [iOS](#) or [Android](#).

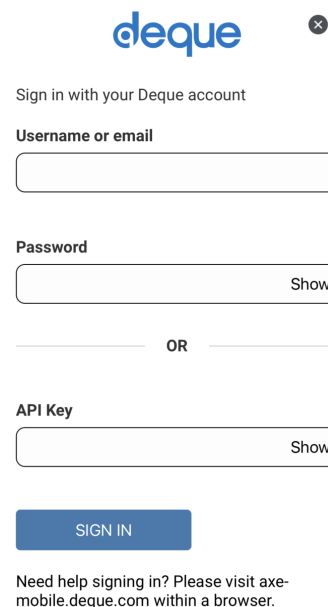
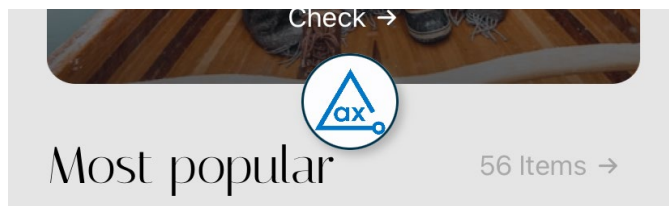
## axe DevTools Mobile for iOS

### Experimental Rules

With this concept of experimental rules, we have introduced a new rule called [Associated Text](#). We have also decided to designate 2 existing rules as experimental, so that they can be further refined: [Inaccessible Action](#) and [Supports Dynamic Type](#). Experimental rules can be ignored the exact same way as standard rules using the [ignore rules feature](#).

### Login Screen

This release contains a new feature that allows users to enter their credentials after the application is built instead of within the codebase. Using this option, axe DevTools Mobile will render a floating action button (FAB) of the axe logo to indicate that the user has not yet authenticated. Tapping on that FAB will launch a login screen where the user can input either their API key or username and password.



deque

Sign in with your Deque account

Username or email

Password Show

OR

API Key Show

SIGN IN

Need help signing in? Please visit [axe-mobile.deque.com](https://axe-mobile.deque.com) within a browser.

Once authenticated, the FAB will appear as the familiar purple accessibility logo meaning axe DevTools Mobile is ready to run scans. This feature is great for driving adoption because users now can generate accessibility scans associated with their credentials without touching any code. Visit [docs.deque.com](https://docs.deque.com) to learn more about how this feature works.

### Optimized Automation Build

In an automated test or CI/CD environment, it may be cumbersome or unreliable to make network requests such as authenticating the axe DevTools Mobile framework. To

overcome such challenges, this release includes a secondary build option that doesn't require authentication or make any network requests to the axe DevTools Mobile service. For this initial release for iOS, the build will support unit testing for UIKit, but the capabilities may be expanded in the future. This optimized build for automation is only available through [Deque's Artifactory](#). Builds that do require authentication and support pushing scans to the server will continue to be available on both Swift Package Manager and Artifactory.

#### Additional Features

- Support for iOS 16
- Added ability to attach tags when using the FAB
- Requests to the axe DevTools Mobile server are now gzip compressed

#### axe DevTools Mobile for Android

##### Experimental Rules

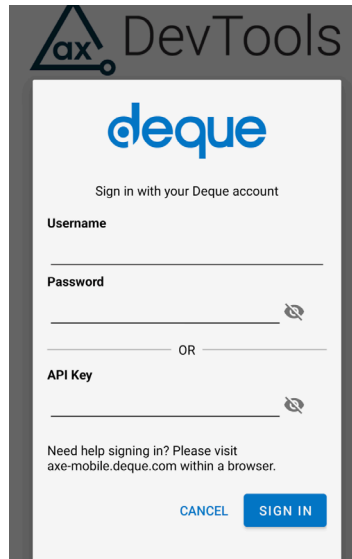
We have released 3 new experimental rules on Android for both XML and Compose:

- [Nested Active Control](#)
- [Nested Element Name](#)
- [Focusable Text](#)
- [Compose Nested Active Control](#)
- [Compose Nested Element Name](#)
- [Compose Focusable Text](#)

To allow for further refinement, we have marked [Inactive Accessible View](#) and [Compose Inactive Accessible View](#) as experimental. Experimental rules can be ignored the exact same way as standard rules using the [ignore rules feature](#).

##### Login Screen

This release contains a new feature that allows users to enter their credentials after the application is built instead of within the codebase. Using this option, axe DevTools Mobile will render a FAB of the axe logo to indicate that the user has not yet authenticated. Tapping on that FAB will launch a login screen where the user can input either their API key or username and password.



Once authenticated, the FAB will appear as the familiar purple accessibility logo meaning axe DevTools Mobile is ready to run scans. This feature is great for driving adoption because users now can generate accessibility scans associated with their credentials without touching any code. Visit [docs.dequeue.com](https://docs.dequeue.com) to learn more about how this feature works.

### Optimized Automation Build

In an automated test or CI/CD environment, it may be cumbersome or unreliable to make network requests such as authenticating the axe DevTools Mobile framework. To overcome such challenges, this release includes a secondary build option that doesn't require authentication or make any network requests to the axe DevTools Mobile service. For this initial release for Android, the build will support UI testing using Espresso, but the capabilities may be expanded in the future. This optimized build for automation is only available through [Dequeue's Artifactory](#). Builds that do require authentication and support pushing scans to the server will continue to be available on both Maven Central and Artifactory.

### Improved Appium Integration

The v3.0.0 release of axe-devtools-android introduces an improved experience for testing with Appium. The imports have been simplified and implementation streamlined. Please note that these are breaking changes, so upgrading will require consuming the new imports and utilizing new APIs. Please see the [new setup guide for Appium](#) as well as the updated examples for [Kotlin](#) and [Java](#).

### Additional Features

- [ComposeSwitchName](#) is a new standard rule
- Disconnect method now resets an existing network connection

- Custom Rules are now supported for Compose
- Requests to the axe DevTools Mobile server are now gzip compressed

## Updates

### axe DevTools Mobile Desktop App

#### Deprecation Notice

As we prepare to deprecate the axe DevTools Mobile desktop app, this v0.26.0 release will be the last. We will continue to support patch releases of this version in the case of security vulnerabilities until May 2023. The desktop app option will be replaced by private cloud instance and on-premise service offerings. If you are currently using the desktop application, and a Deque representative has not already reached out to you about these offerings, please submit a helpdesk request at [support.deque.com](https://support.deque.com) or email [helpdesk@deque.com](mailto:helpdesk@deque.com).

### axe DevTools Mobile Dashboard

#### Needs Review

Results reported with the Incomplete status are now displayed as Needs Review. This is purely a change in verbiage to align with other Deque products. The status still indicates that a confident pass or fail could not be automatically determined, so the view should be manually tested.

### axe DevTools Mobile for iOS

#### Deprecated APIs

In the June 2022 release, we marked a handful of APIs as deprecated in the UIKit framework. Those APIs have been removed in this v3.2.0 version. Before upgrading, please ensure that you are either not using any deprecated API or are prepared to update to a stable API.

#### Minimum Supported iOS Version

The minimum supported version has changed from iOS 10 to iOS 13.

# Bug Fixes

## axe DevTools Mobile Dashboard and Desktop App

- Prevent screen reader navigation outside of the Issues Found and Inspect Panels
- The esc button can now be used to close the Issues Found and Inspect panels
- Focus is returned to the Inspect button upon close of the Inspect panel
- Edit tags buttons in the Scans table now have a more descriptive label
- Improved screen reader announcements for the checkboxes in the Scans table
- The upload component is more accurately labeled and has improved focus and hover outlines
- The loader for exporting scans is now labeled
- The offscreen highlight toast message no longer has a close button and disappears automatically
- It's no longer possible to get stuck in a keyboard trap in the Inspect view hierarchy
- All modal dialogs now have an appropriate accessible label
- Improved the screen reader experience for the accessibility score chart

## axe DevTools Mobile for iOS

- No longer report a Label In Name issue on TextFields that contain a button
- Fix false positives with search bars for Inaccessible Action
- Report needs review result status for Screen Title if there is no nav bar
- Allow Screen Title to pass for an image with an accessibility label
- Supports Dynamic Type is now inapplicable for headers and footers
- Fix a false negative in Supports Dynamic Type for collection cells
- Improve accuracy of Supports Dynamic Type results for table view cells
- Reduce false positive results from Supports Dynamic Type for UITextFields [CSD-17038]
- Return needs review status instead of fail for Supports Dynamic Type when text uses UIFontMetrics [CSD-16500]
- Fixed an issue where the accessibility icon in the FAB would drift position
- Reduced false positives for multiple rules for PickerViews and table view cells
- Ensure that buttons within text fields are appropriately marked as focusable
- Improve understandability of properties reported in the dashboard for several rules

## axe DevTools Mobile for Android

- Add the ability to delete scans using AxeDevToolsCompose
- Added Java overloads to the connect method [CSD-21376]
- Active View Name rule now reports inapplicable for views marked not important for accessibility [CSD-21376]
- Screen Title rule now reports a failure when activity title is null and there's no fragments
- When there is an authentication failure with credentials specified in the code, the FAB with the axe logo will appear instead of the accessibility icon FAB
- Fixed false positive for Inactive Accessible View when the parent view is clickable
- Reduce false positives in Label In Name by ignoring special characters

## 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 Dashboard

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

- 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](mailto: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

## axe DevTools Mobile for Android

- 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](mailto:helpdesk@deque.com) for assistance. Under certain conditions, there may be two Compose root nodes existing at the same time.