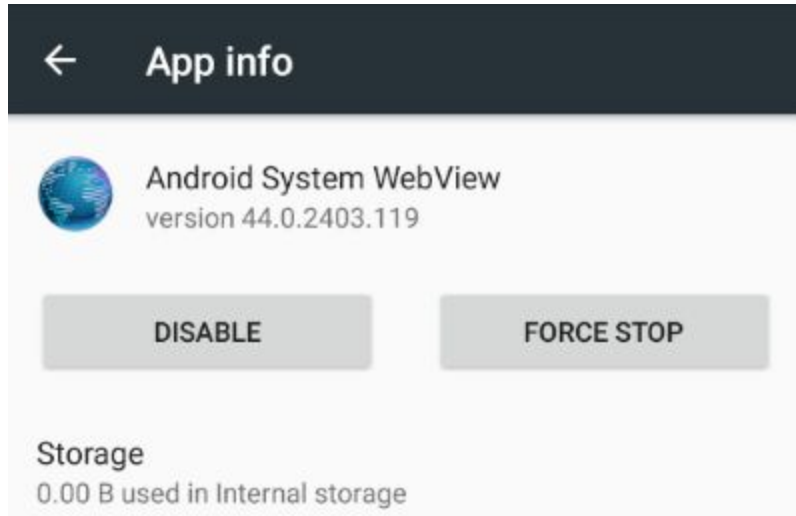


Creating compatible field plug-ins

Max S. Haberman

SurveyCTO field plug-ins use Android webview. For the field plug-in you are creating, make sure the JavaScript features used are compatible with the version of Android the field plug-in will be used on. For example, Android 6 often runs webview 44.



To access this page, go to *Settings > Apps*, tap on the three-dot icon on the upper-right, tap "Show system apps", then open *Android System WebView*. Exact location of the version number can vary; for example, in Android 10, the version number is on the bottom.

This means Android 6 can only run JavaScript features that are compatible with webview version 44 and below. For example, the "[let](#)" feature was introduced in Android webview 49, so it will not work in field plug-ins running on Android 6 or below.

If you are considering using newer JavaScript features, check their compatibility with the collection device(s) that will be used. The [Mozilla MDN](#) includes charts that show the compatibility of each feature (for an example, go to the "[let](#)" page, and scroll down to *Browser compatibility*: you'll see that when "let" is used in Android webview, it requires at least version 49). You can also use [CanIUse](#) (Android webview versions are usually the same as the Chrome versions of the same number; for example, Android webview 49 is very similar to Chrome 49). Avoiding newer features also allows other SurveyCTO users who are using older devices to use your field plug-ins.

Android webview versions

Below is a chart that approximates the version of webview running on each version of Android.

Android	Webview version
---------	-----------------

version	
4.4 (KitKat)	30
5 (Lollipop)	37
5.1 (Lollipop)	39
6 (Marshmallow)	44
7.1.1 (Nougat)	55
8.1 (Oreo)	61
9 (Pie)	69
10	74

Note: Besides version 4.4, this chart of webview versions was retrieved using Android Studio emulators. The actual webview version and features can vary. For example, method definitions are supported starting in Android webview 39, but they still do not work in Android 5.1 (but they do in Android 6).

iOS compatibility

iOS devices use WKWebView for their webview, which unfortunately is not open about version numbers and which JavaScript features are allowed. However, this is usually not as much of an issue, since iOS 13 (as of this writing, the latest version of iOS) is used on 70% of all iPhones ([source](#), updated January 2020), compared to about 40% of Android devices running Android 9 or 10 ([source](#), updated April 2020). It is still a good idea to make sure your field plug-in works on the collection devices.

Alternative features

Below is a list of popular JavaScript features that are typically not compatible with older versions of webview, and what features you can use instead.

[let](#) and [const](#)

The "let" statement is rarely supported on older devices. The "const" statement is usually supported, but not in all situations (for example, it cannot be reassigned using a different iteration of a loop). To be safe, use "[var](#)" instead.

[includes\(\)](#)

This method for the String class was not introduced until webview 41. Instead, use [indexOf\(\)](#), like this:

appearance.indexOf('minimal') !== -1

Method definitions

You might be tempted to define includes() as a method of the String class, but this is not always supported either. This is typically bad practice anyway.

for...of

While convenient for iterating over arrays and NodeList objects, instead, you can use a normal for statement.