

Titanium SDK 8.1.0.GA Release Note

Titanium SDK 8.1.0.GA - 13 August 2019

- [About this release](#)
- [New features](#)
 - [Android platform](#)
 - [iOS platform](#)
 - [Windows platform](#)
 - [Multiple platforms](#)
- [Community credits](#)
- [Fixed issues](#)
 - [Android](#)
 - [iOS](#)
 - [Windows](#)
 - [Multiple platforms](#)
- [Improvements](#)
 - [Android](#)
 - [iOS](#)
 - [Windows](#)
 - [Multiple platforms](#)
- [API changes](#)
 - [New APIs](#)
- [SDK modules versions](#)
 - [Android and iOS](#)
 - [CommonJS](#)
 - [Hyperloop](#)

About this release

Titanium SDK 8.1.0 is a minor release of the SDK, addressing high-priority issues from previous releases.

As of this release, Titanium SDK 8.0.x will not receive updates more than six months after the release of 8.1.0 (current) (2020-02-13). Any needed fixes will be in 8.1.x or later supported releases within the 8.x branch. See [Axway Appcelerator Deprecation Policy](#) and [Nominal Lifetimes](#) documents for details.



With the release of Titanium SDK 9.0.0, we will no longer support Node.js 8.X. Node.js 10.13.0 will be the new minimum supported version with SDK 9.0.0.

New features

Android platform

- [TIMOB-4041](#) - Android: Add "separatorStyle" property to `Ti.UI.ListView`
 - Added support for `ListView.separatorStyle`

ListView.separatorStyle

> [Expand](#)

```
var win = Ti.UI.createWindow({ backgroundColor: 'black', fullscreen: true
});
var listView = Ti.UI.createListView({
    separatorStyle: Ti.UI.TABLE_VIEW_SEPARATOR_STYLE_NONE,
});
var section = Ti.UI.createListSection();
var sectionData = ;
var i = 25;
for (var k = 0; k < 25; k++) {
    sectionData.push({
        properties: {
            title: 'Row ' + (k + 1)
        }
    });
}
section.setItems(sectionData);
listView.sections = section;
win.add(listView);
win.open();
```

- [TIMOB-14460](#) - Android: Add `Ti.Platform.canOpenURL()` support
 - Added support for `Ti.Platform.canOpenURL()`
- [TIMOB-26793](#) - Android: Hide radio buttons `OptionDialog`
 - Added support to use `OptionDialog` without radio buttons

OptionDialog

> [Expand](#)

[source](#)

```
var win = Ti.UI.createWindow({
  title: 'Click window to test OptionDialog',
  backgroundColor: 'white'
});
var opts = {
  title: 'Select an option',
  options: 'Option 1', 'Option 2', 'Option 3', 'Option 4',
  buttonNames: 'Cancel'
}
var dialog;
win.addEventListener('click', function() {
  dialog = Ti.UI.createOptionDialog(opts);
  dialog.addEventListener('click', onSelectDialog);
  dialog.addEventListener('cancel', function(e) {
    alert('Dialog canceled! e.cancel = ' + e.cancel + ', e.index = ' +
e.index);
  })
  dialog.show();
});
function onSelectDialog(e) {
  console.log(e);
}
var btn = Ti.UI.createButton({
  title: "toggle selected index",
  bottom: 5
});
btn.addEventListener("click", function() {
  if (opts.selectedIndex == 0) {
    opts.selectedIndex = -1;
  } else {
    opts.selectedIndex = 0;
  }
});
win.add(btn);
win.open();
```

iOS platform

- [TIMOB-18421](#) - iOS: Add animation properties "springVelocity" and "dampingRatio"
 - Added `dampingRatio` and `springVelocity` to allow iOS spring animations
- [TIMOB-26523](#) - iOS 11+: Support large title style (dynamic fonts)
 - Added support for dynamic fonts in `UIFontTextStyleLargeTitle`

UIFontTextStyleLargeTitle

> Expand

```
var win = Ti.UI.createWindow({
  backgroundColor: '#fff',
  extendSafeArea: false
});

var scrollView = Ti.UI.createScrollView({ layout: 'vertical', top: 50 });

var textStyles =
  Ti.UI.TEXT_STYLE_HEADLINE,
  Ti.UI.TEXT_STYLE_SUBHEADLINE,
  Ti.UI.TEXT_STYLE_BODY,
  Ti.UI.TEXT_STYLE_FOOTNOTE,
  Ti.UI.TEXT_STYLE_CAPTION1,
  Ti.UI.TEXT_STYLE_CAPTION2,
  Ti.UI.TEXT_STYLE_CALLOUT,
  Ti.UI.TEXT_STYLE_TITLE1,
  Ti.UI.TEXT_STYLE_TITLE2,
  Ti.UI.TEXT_STYLE_TITLE3,
  Ti.UI.TEXT_STYLE_LARGE_TITLE,
;

for (var i = 0; i < textStyles.length; i++) {
  var textStyle = textStyles[i];
  scrollView.add(Ti.UI.createLabel({
    top: 20,
    text: textStyle,
    font: {
      textStyle: textStyle
    }
  }));
}

win.add(scrollView);
win.open();
```

source

- [TIMOB-26599](#) - iOS: Add "allowTranscoding" option to `Ti.Media.openPhotoGallery()`
 - Added option to disable video compression when importing video
- [TIMOB-26973](#) - iOS: Add `accessibilityIdentifier`
 - Added `accessibilityIdentifier` which can compose views on iOS the same way `content description` is composed on Android
- [TIMOB-27124](#) - iOS 13: Support running on Xcode 11 Simulators
 - Added support for Xcode 11

Windows platform

- [TIMOB-25778](#) - Windows: Implement `Ti.UI.Label.ellipsize`
 - Implemented `Ti.UI.Label.ellipsize`
 - `TEXT_ELLIPSIZE_TRUNCATE_START`, `TEXT_ELLIPSIZE_TRUNCATE_MIDDLE`, and `TEXT_ELLIPSIZE_TRUNCATE_MARQUEE` are not supported on Windows because platform component doesn't support them

Multiple platforms

- [TIMOB-26571](#) - TiAPI: Global process object compatible with Node.js
 - Expanded the global process object to be more compatible with Node.js
- [TIMOB-26669](#) - TiAPI: Create Node-compatible `assert` module API
 - Added Node.js-compatible `asset` module API

- [TIMOB-26670](#) - TiAPI: Create Node-compatible util module API
 - Added Node.js-compatible util module API
 - This API does not include every method of `util.types`, nor does it include `util.extends()`, `util.isDeepStrictEqual()`, or `util.inspect`
 - Currently supports `depth`, `showHidden`, and to some degree `breakLength`
- [TIMOB-27150](#) - Add ability to alias required modules in Titanium
 - Added ability to alias required modules

Community credits

- Prafulla Bansode - [TIMOB-24415](#)
- Sandro Lain - [TIMOB-24418](#), [TIMOB-24423](#), [TIMOB-24443](#)
- Hans Knöchel - [TIMOB-26101](#), [TIMOB-26854](#), [TIMOB-26862](#), [TIMOB-26885](#), [TIMOB-26946](#), [TIMOB-27049](#), [TIMOB-27082](#), [TIMOB-27089](#), [TIMOB-27090](#), [TIMOB-27100](#), [TIMOB-27103](#), [TIMOB-26523](#), [TIMOB-26566](#), [TIMOB-26509](#), [TIMOB-26686](#), [TIMOB-26770](#), [TIMOB-27155](#), [TIMOB-27210](#)
- kedomuz - [TIMOB-26326](#)
- Michael Gangolf - [TIMOB-26456](#), [TIMOB-26993](#), [TIMOB-25556](#), [TIMOB-26793](#)
- Sergey Volkov - [TIMOB-26973](#)
- shumne - [TIMOB-26795](#)
- alessandro la rocca - [TIMOB-26829](#)
- Andreas Pingas - [TIMOB-26878](#), [TIMOB-27188](#), [TIMOB-27232](#)
- Arbind - [TIMOB-26898](#)
- grebulon - [TIMOB-26954](#)
- Muhammad Ahmed Fahad - [TIMOB-26986](#)
- Adam Armstrong - [TIMOB-27087](#)
- Tim Poulsen - [TIMOB-14460](#)
- Richard Lustenberg - [TIMOB-26599](#)
- Mark Henderson - [TIMOB-27098](#)

Fixed issues

Android

- [TIMOB-24415](#) - Android: Listview not able to add accessibility label in the ListItem components
- [TIMOB-24881](#) - Android: Label ellipsize modes START and MIDDLE don't work if "autoLink" is true
- [TIMOB-26456](#) - Android: Optimize TiBlob.imageAsResized
- [TIMOB-26650](#) - Android: Using toImage on views with border properties not added to UI results in empty blob
- [TIMOB-26841](#) - Android: Reading TextField "backgroundDisabledColor" will crash if background/border color is assigned
- [TIMOB-26862](#) - Android: TextField/TextArea within a TableView can have performance issues with some keyboards
- [TIMOB-26869](#) - CLI: Android build fails with cannot find module error
- [TIMOB-26878](#) - Android: Ti.Database.install() fails if destination directory does not exist
- [TIMOB-26954](#) - Android: Reading a JSON file fails on Android Q
- [TIMOB-26993](#) - Android: WebView sometimes crashes on window close after zooming content
- [TIMOB-27087](#) - Android: ShortcutItem not being updated based on "id"
- [TIMOB-27094](#) - Android: ListView's SearchBar/SearchView wrongly overlaps rows as of 8.0.1
- [TIMOB-27100](#) - Android: TabGroup labels not displayed anymore (SDK 8.1.0+)
- [TIMOB-27153](#) - Android: Prevent multiple exception logs
- [TIMOB-27188](#) - Android: TabGroup crashes if tab "title" property is not set as of 8.0.2
- [TIMOB-27232](#) - Android: Setting Window "barColor" while using a theme without an ActionBar/TitleBar will cause a crash as of 8.1.0
- [TIMOB-27269](#) - Android: Do not retry sending events if no connectivity
- [TIMOB-27300](#) - Android: AssetCryptImpl buffer overflow for large JS assets
- [TIMOB-27314](#) - Android: Unable to build to device on Windows

iOS

- [TIMOB-26795](#) - iOS: Event listener on overlay is not working for camera when camera is kept idle for a few minutes
- [TIMOB-26829](#) - iOS: Titanium.Media.openMusicLibrary returns empty items on iOS 9.3.5
- [TIMOB-26833](#) - iOS: Two startup log messages are output
- [TIMOB-26838](#) - iOS: the openWindow() method of NavigationWindow does not work when called from base window open event
- [TIMOB-26854](#) - iOS: Crash when handling user-activities
- [TIMOB-26885](#) - iOS: Crash when trying to push a window twice (e.g. on slow devices)
- [TIMOB-26946](#) - iOS: Launch screen shows last screen when quickly relaunching the app
- [TIMOB-27049](#) - iOS: Simulator can crash because of the "trackUserInteraction" feature
- [TIMOB-27074](#) - iOS: Rebuilding the app with simulator fails with due to log port being occupied
- [TIMOB-27082](#) - iOS: Modules using TiBlob and built before SDK 8.1 causing app crash
- [TIMOB-27089](#) - iOS: App crashes when asking for location permissions (SDK 8.1.0+)
- [TIMOB-27090](#) - iOS: Using a non-string value in alert() causes app to crash in SDK 8.0.0+

- TIMOB-27098 - iOS: debugger does not show all app files loaded
- TIMOB-27103 - iOS: Native extensions without sources are crashing build
- TIMOB-27155 - iOS: Incremental builds broken
- TIMOB-27175 - iOS: `Ti.Platform.id` changes when updating to SDK 8.0.0
- TIMOB-27202 - iOS: Rebuilding app on device throws Couldn't find module error
- TIMOB-27203 - iOS: no apiversion validation performed on application build
- TIMOB-27210 - iOS: Cannot find iOS 13 simulators in Xcode 11 Beta 3
- TIMOB-27230 - iOS: CLI prefers Xcode 11/iOS 13 even though Xcode 10.2.1 is selected
- TIMOB-27270 - iOS: Allow capture of signal exceptions
- TIMOB-27277 - iOS: Invalid analytics platform
- TIMOB-27278 - iOS: App crashes upon giving permission to Calendar
- TIMOB-27313 - iOS: Regression-Console.log does not concatenate and log if it has multiple arguments with space

Windows

- TIMOB-24418 - Windows Phone: `OptionDialog` undefined error
- TIMOB-24423 - Windows Phone: `TableViewCell` and `Label` layout not respected
- TIMOB-24443 - Windows Phone: `Progressbar` not respect width and not resize
- TIMOB-25134 - Windows: Mocha test `Titanium.UI.TextField` padding test failure
- TIMOB-26550 - Windows: Add support for file property to `HttpClient`
- TIMOB-26664 - Windows: `Ti.Geolocation.lastGeolocation` should return string
- TIMOB-26689 - Windows: Wrong height for `TableViewCell` with `Label`
- TIMOB-26690 - Windows: `TabGroup` does not fire open/close events
- TIMOB-26791 - Windows: `Ti.UI.View` reports top/left/bottom/right values as `Strings` always
- TIMOB-26797 - Windows: `removeEventListener` removes wrong callback
- TIMOB-26969 - Windows: `Codec.encodeNumber/decodeNumber` should throw `Error` on invalid setup
- TIMOB-27257 - [Windows] `Error` is shown when packaging a windows module

Multiple platforms

- TIMOB-27204 - CLI: apiversion validation always fails for native modules installed via npm

Improvements

Android

- TIMOB-26686 - Android: Allow `Ti.Network.HttpClient` "file" parameter to be set to `Ti.Filesystem.File` object like iOS
 - Parity for `File` object support
- TIMOB-26817 - Android: Add parity for `Window`'s `barColor` property.
 - Parity for `barColor` property of `Ti.UI.Window`
- TIMOB-26848 - Android: Re-add "baseUrl" support to `WebView.setHtml()` that was removed in 6.0.3
 - Re-implemented `baseUrl` to support `WebView.setHtml`
- TIMOB-26877 - Android: `Ti.Database.install()` should throw exception like iOS if source db file not found
 - Modified `Ti.Database.install()` to throw an exception if given source database file was not found
- TIMOB-26909 - Android: Improve V8 cold start performance
 - Updated V8 to 7.3.492.26 which provides performance updates
- TIMOB-26957 - Android: Update V8 runtime to 7.3.492.26
 - Updated V8 to 7.3.492.26 which provides performance updates
- TIMOB-27042 - Android: Update `ti.playservices` to 16.1.3
 - Updated `ti.playservices` to version 16.1.3
- TIMOB-27111 - Android: Implement async `Ti.Database.DB` methods
 - Implement asynchronous `Ti.Database.DB` methods for executing SQL queries
 - `Ti.Database.DB.executeAsync(query, parameters[], callback)`
 - `Ti.Database.DB.executeAllAsync(queries[], callback)`
 - Implement method for executing multiple queries in one request

iOS

- TIMOB-26509 - iOS: Allow `Ti.UI.SearchBar` to be displayed in navigation-bar
 - Added support to show search bar navigation for `Ti.UITableView` and `Ti.UITableViewController`
- TIMOB-26770 - iOS: Use Swift 5 by default
 - Added support for Swift 5
- TIMOB-27043 - Cache JS processing between builds
 - Improved incremental app build times
- TIMOB-27047 - iOS: Version of SDKs default modules are not up to date
 - Updated the following modules
 - `urlSession` from 2.1.0 to 2.2.0

- `ti.map` from 3.1.0 to 3.1.2
- `ti.safaridialog` from 1.1.1 to 3.1.0
- `ti.identity` from 1.0.5 to 1.0.6
- **TIMOB-27164** - iOS: Implement async `Ti.Database.DB` methods
 - Implement asynchronous `Ti.Database.DB` methods for executing SQL queries
 - `Ti.Database.DB.executeAsync(query, parameters[], callback)`
 - `Ti.Database.DB.executeAllAsync(queries[], callback)`
 - Implement method for executing multiple queries in one request

Windows

- **TIMOB-19965** - Windows: Implement `WebView.data`
 - Implemented `WebView.data`

WebView.data
> Expand

```

var win = Ti.UI.createWindow({
    backgroundColor: 'green'
});
var blob = Ti.Filesystem.getFile('app.js').read();
var webview = Ti.UI.createWebView({
    data: blob,
    height: Ti.UI.FILL,
    width: Ti.UI.FILL
});
webview.addEventListener('load', function () {
    Ti.API.info(typeof webview.data);
});
win.add(webview);
win.open();

```

[source](#)

- **TIMOB-26575** - Windows: Support async variants of `Ti.Network.TCP` `#read` and `#write`
 - Parity for async variants of `Ti.Network.TCP`
- **TIMOB-26855** - Windows: Add "baseUrl" support to `WebView.setHtml()`
 - Re-implemented `baseUrl` to support `WebView.setHtml`
- **TIMOB-26961** - Windows: Save `HttpClient` instance until callback is fired
 - The `httpClient` instance is saved until `onerror` or `onloadcallback` is called

> [Expand](#)

```
win.addEventListener('open', function () {
  var _tmpURL = "https://docs.appcelerator.com/platform/latest/#!/api/"
  var _tmpClient = Ti.Network.createHttpClient({

    onload: function (e) {
      alert('Loaded');
    },
    onerror: function (e) {
      alert("Error:" + e.error);
    },
    timeout: 30000
  });

  _tmpClient.open("GET", _tmpURL);
  _tmpClient.send();
  // In this case _tmpClient may be garbage-collected as soon as this
  function is finished.
  // We could think this as "logical" failure but I think Titanium
  developers tend to use HttpClient like this.
});

win.open();
```

Multiple platforms

- [TIMOB-26574](#) - TiAPI: Implement set/clearImmediate
 - Implemented `process.nextTick()` and the global `setImmediate/clearImmediate`
- [TIMOB-26729](#) - TiAPI: Ship baseline polyfills in built SDK
 - This update injects the `core-js/regenerator` polyfills into the bundle `common` JS SDK code we ship when we build
- [TIMOB-27045](#) - Pass along ENV variables to `process.env` on development builds
 - Write `env.json` file with ENV variables
- [TIMOB-27182](#) - Update Hyperloop to 4.0.3
 - Android: Debugging in Hyperloop enabled projects
 - iOS: Debugging in Hyperloop enabled projects
 - iOS: Correctly require minimum SDK 8.0.0

API changes

New APIs

The following APIs are new or have expanded platform support in release 8.1.0.

API	Type	Notes
<code>Titanium.Database.DB.executeAll</code>	method	Synchronously executes an array of SQL statements against the database and returns an array of <code>ResultSet</code> . On failure, this will throw an <code>Error</code> that reports the failed index and partial results (New API, supported on Android.)
<code>Titanium.Database.DB.executeAllAsync</code>	method	Asynchronously executes an array of SQL statements against the database and fires a callback with a possible <code>Error</code> , and an array of <code>ResultSet</code> . On failure, this will call the callback with an <code>Error</code> that reports the failed index, and a second argument with the partial results (New API, supported on Android.)
<code>Titanium.Database.DB.executeAsync</code>	method	Asynchronously executes an SQL statement against the database and fires a callback with a possible <code>Error</code> argument, and a second argument holding a possible <code>ResultSet</code> . (New API, supported on Android.)

Titanium.Platform.canOpenURL	method	Returns whether the system is configured with a default application to handle the URL's protocol/scheme.(Added support for Android.)
Titanium.UI.Animation.dampingRatio	property	The damping ratio for the spring animation as it approaches its quiescent state.(New API, supported on iPhone and iPad.)
Titanium.UI.Animation.getDampingRatio	method	Access Titanium.UI.Animation.dampingRatio instead.
Titanium.UI.Animation.getSpringVelocity	method	Access Titanium.UI.Animation.springVelocity instead.
Titanium.UI.Animation.setDampingRatio	method	Set the value using Titanium.UI.Animation.dampingRatio instead.
Titanium.UI.Animation.setSpringVelocity	method	Set the value using Titanium.UI.Animation.springVelocity instead.
Titanium.UI.Animation.springVelocity	property	The initial spring velocity. (New API, supported on iPhone and iPad.)
Titanium.UI.ListView.getSeparatorStyle	method	Access Titanium.UI.ListView.separatorStyle instead.
Titanium.UI.ListView.getShowSearchBarInNavBar	method	Access Titanium.UI.ListView.showSearchBarInNavBar instead.
Titanium.UI.ListView.separatorStyle	property	Separator style constant. (Added support for Android.)
Titanium.UI.ListView.setSeparatorStyle	method	Set the value using Titanium.UI.ListView.separatorStyle instead.
Titanium.UI.ListView.setShowSearchBarInNavBar	method	Set the value using Titanium.UI.ListView.showSearchBarInNavBar instead.
Titanium.UI.ListView.showSearchBarInNavBar	property	A Boolean indicating whether search bar will be in navigation bar. (New API, supported on iPhone and iPad.)
Titanium.UI.TEXT_STYLE_LARGE_TITLE	property	Specifies the text style for the Font Object. (New API, supported on iPhone and iPad.)
Titanium.UI.TableView.getShowSearchBarInNavBar	method	Access Titanium.UI.TableView.showSearchBarInNavBar instead.
Titanium.UI.TableView.setShowSearchBarInNavBar	method	Set the value using Titanium.UI.TableView.showSearchBarInNavBar instead.
Titanium.UI.TableView.showSearchBarInNavBar	property	A Boolean indicating whether search bar will be in navigation bar. (New API, supported on iPhone and iPad.)
Titanium.UI.WebView.setHtml	method	Sets the value of html property. (Added support for Android.)
Titanium.UI.Window.getHidesSearchBarWhenScrolling	method	Access Titanium.UI.Window.hidesSearchBarWhenScrolling instead.
Titanium.UI.Window.hidesSearchBarWhenScrolling	property	A Boolean value indicating whether the integrated search bar is hidden when scrolling any underlying content. (New API, supported on iPhone and iPad.)
Titanium.UI.Window.setHidesSearchBarWhenScrolling	method	Set the value using Titanium.UI.Window.hidesSearchBarWhenScrolling instead.

SDK modules versions

Android and iOS

Module	Android version	iOS version
urlSession	n/a	2.2.0
facebook	8.0.0	6.0.0
ti_coremotion	n/a	2.0.1
ti_map	4.3.1	3.1.2
ti_safaridialog	n/a	1.1.1
ti_webdialog	1.1.0	1.1.0
ti_touchid	3.0.1	2.1.4

ti_identity	2.1.0	1.0.6
ti_cloudpush	6.0.1	n/a
ti_playservices	16.1.3	n/a

CommonJS

Module	Version
ti.cloud	3.2.11

Hyperloop

Module	Version
Hyperloop	4.0.3