



NativeScript

Community Chat 8/05/2020

NativeScript 7.0 - What to expect

- * es5 > es2017 and higher
- * @NativeClass decorator
- * @nativescript/core - no more deep path confusion
- * tree shaking
- * global.isIOS and global.isAndroid webpack nicety

NativeScript 7.0 - What to expect

This eye sore will be no longer:

```
import * as utils from 'tns-core-modules/utils/utils';
import * as utilsModule from 'tns-core-modules/utils';
import * as app from 'tns-core-modules/application/application';
import * as application from 'tns-core-modules/application';
import { StackLayout } from 'tns-core-modules/ui/layouts/stack-layout/stack-
layout';
import * as stackLayout from 'tns-core-modules/ui/layouts/stack-layout';
import { Color } from 'tns-core-modules/color/color';
import { getDisplayDensity } from '@nativescript/core/utils/layout-helper';
import { KeyframeAnimation } from '@nativescript/core/ui/animation/keyframe-
animation';
```

NativeScript 7.0 - What to expect

This clarity and simplification throughout:

```
import {  
  Application,  
  Color,  
  KeyframeAnimation,  
  StackLayout,  
  Utils  
} from '@nativescript/core';
```

Enables Advanced Tree Shaking: <https://webpack.js.org/guides/tree-shaking/>

The webpack 2 release came with built-in support for ES2015 modules (alias harmony modules) as well as unused module export detection. The new webpack 4 release expands on this capability with a way to provide hints to the compiler via the "sideEffects" package.json property to denote which files in your project are "pure" and therefore safe to prune if unused.

Q&A