

SAVING THE WORLD ONE CODEBASE AT  
A TIME

---

DEVELOPING NATIVE MOBILE  
APPS WITH ANGULAR

---

# WHAT ARE WE DOING HERE!?

- ▶ Who builds web apps with Angular?
- ▶ Who builds native apps?
- ▶ Who wants to build both with one codebase?

# WHO AM I?

- ▶ Code Slinger
- ▶ Steak Griller
- ▶ Wrangler of rug rats
- ▶ Not a Coffee Drinker

Twitter- [@halpin\\_io](https://twitter.com/@halpin_io)

Web- [halpincodes.com](http://halpincodes.com)



---

# SHORT AND SWEET

- ▶ What is NativeScript?
- ▶ How does NativeScript work?
- ▶ How do you use it with Angular?
- ▶ Demo / Live Coding

---

# YOUR BOSS



**Uh, yeah.**

---

# YOUR PROBLEM



@Halpin\_io

---

# YOUR TEAM



@Halpin\_io

---

# YOUR RELIEF



@Halpin\_io

# WHAT IS NATIVESCRIPT?



---

# HISTORY LESSON

- ▶ Open source framework for creating iOS and Android apps
- ▶ Maintained by Telerik
- ▶ Leverages Plain JS, Typescript, Vue or Angular
- ▶ Native Features:
  - ▶ Offline support
  - ▶ Device APIs
  - ▶ App Monetization
  - ▶ Home Screen Availability

---

# WHATS COOL ABOUT NATIVESCRIPT?

- ▶ You don't have to learn Java, Swift or Objective-C
- ▶ A single code base across multiple applications
- ▶ Use your existing skills!



---

# NATIVE LOOK AND FEEL

- ▶ No Wrappers
- ▶ No DOM
- ▶ No Cross Compilation - Just JavaScript
- ▶ Cross platform abstraction via their API
- ▶ Directly access all native API - can re-use code

---

# NATIVESCRIPT IS NOT...

- ▶ A web view
- ▶ A programming language
- ▶ An add-on

---

# OTHER CHOICES

- ▶ React Native
- ▶ Xamarin
- ▶ Titanium
- ▶ Cordova
- ▶ PWA's
- ▶ Ruby Motion

# APPLICATION ARCHITECTURE

NATIVE

# Native APIs



NATIVESCRIPT RUNTIME

JAVASCRIPT  
VIRTUAL MACHINE

calls JavaScript functions

accesses native objects

Call Dispatcher

Type conversion

Metadata

calls JavaScript functions

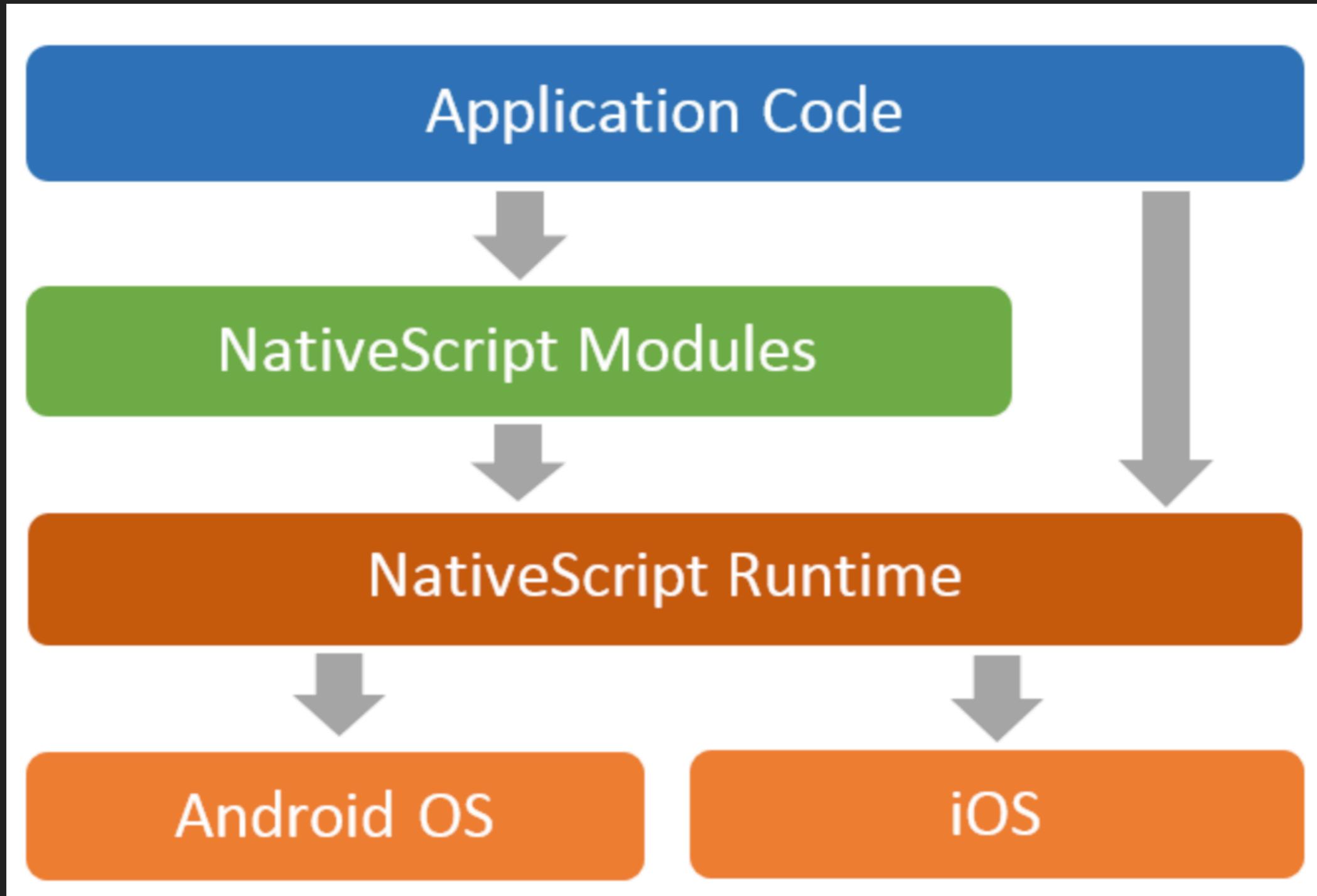
calls Native API using JavaScript

App code (JavaScript)

JavaScript Libraries



# HOW DOES IT WORK?



# USING NATIVESCRIPT WITH ANGULAR

---

# GETTING STARTED

- ▶ Docs are great
- ▶ Windows Setup is harder than Mac
- ▶ Different Options:
  - ▶ Scripted vs. Manual
  - ▶ Sidekick

<https://docs.nativescript.org/start/quick-setup>

---

# START FROM SCRATCH

- ▶ Must have the following installed:
  - ▶ Node.JS
  - ▶ Xcode / Android Studio
  - ▶ NativeScript CLI
  - ▶ Angular CLI
  - ▶ NativeScript Schematic

```
ng new --collection=@nativescript/schematics project-name –  
shared
```

---

# CODE SHARE WITH WEB APP

- ▶ Prerequisite: NativeScript CLI and Angular CLI Installation
- ▶ Update your Angular CLI:  
`ng update @angular/cli @angular/core`
- ▶ Install the NativeScript Schematic:  
`npm i -g @nativescript/schematics`
- ▶ Add the NativeScript Schematic:  
`ng add @nativescript/schematics`

---

# WHAT IS GOING ON?

- ▶ NativeScript-specific copy gets added for:
  - ▶ Main file - `main.tns.ts`, which bootstraps the NativeScript Entry Module
  - ▶ Entry Module file - by default this is `app.module.tns.ts`
  - ▶ Entry Component file - by default this is `app.component.ts`
  - ▶ App Routing file - `app-routing.module.ts`, which brings in NativeScriptRouterModule

# MIGRATE



---

► Routing

► Components

► Modules

# ROUTE MIGRATION

- ▶ Routes are now split, but can be combined
- ▶ Combine navigation paths as you migrate page components to NativeScript
- ▶ Keep routes separate for different app pages

```
import { NgModule } from '@angular/core';
import { RouterModule } from '@angular/router';
import { ROUTES } from './app.routes';

@NgModule({
  imports: [RouterModule.forRoot(ROUTES)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

```
import { NgModule } from '@angular/core';
import { NativeScriptRouterModule } from 'nativescript-angular/router';
import { ROUTES } from './app.routes';

@NgModule({
  imports: [NativeScriptRouterModule.forRoot(ROUTES)],
  exports: [NativeScriptRouterModule]
})
export class AppRoutingModule { }
```

# COMPONENT MIGRATION

- ▶ Use the helper:

```
ng g migrate-component --name=component-name
```

- ▶ Generates NativeScript component code
- ▶ Adds component to NativeScript (app.module.tns)

```
src
└── app
    ├── home
    │   ├── home.component.html
    │   ├── home.component.tns.html  <= create
    │   └── home.component.ts
    ├── app.module.tns.ts          <= update
    └── app.module.ts
```

---

# COMPONENT MODULE MIGRATION

- ▶ Components with different module structure:  
`ng g migrate-component --name=dog --module=animals`
- ▶ For Nonstandard module names, use –module-path flag  
`ng g migrate-component --name=dog –module-path=animals/animals-md.ts`

# COMPONENT DIFFERENCES

- ▶ You must create the NativeScript UI
- ▶ Platform specific code must be split for Web vs Native functionality (Services, Modules, etc)

```
import { DrawerHelper } from "./drawer-helper";
@Component({...})
export class NameComponent {
    // a lot of shared code here

    public showMenu() {
        DrawerHelper.show();
    }

    public hideMenu() {
        DrawerHelper.hide();
    }
}
```

# SERVICE MIGRATION

- ▶ Split web and mobile functionality into two separate files
- ▶ Inject the service into the component providers

```
import { DrawerService } from "./drawer-service";
@Component({
  providers: [DrawerService]
})
export class NameComponent {
  // a lot of shared code here

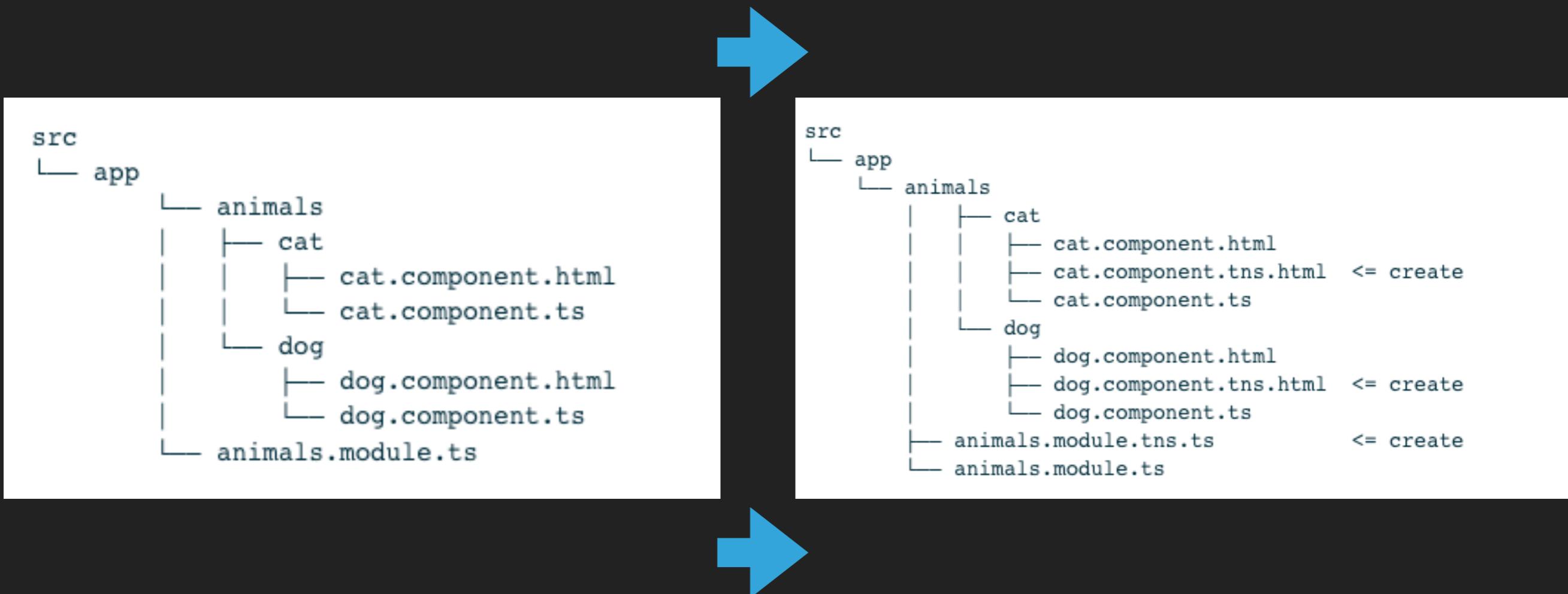
  constructor(drawerService: DrawerService) {}

  public showMenu() {
    drawerService.show();
  }

  public hideMenu() {
    drawerService.hide();
  }
}
```

# MODULE MIGRATION

- ▶ Migrating the module:  
`ng g migrate-module --name=animals`



---

## ADDITIONAL INFORMATION - PAGE NAVIGATION

- ▶ //NativeScript Router Link  
`nsRouterLink="/detail/{{hero.id}}"`
- ▶ //Angular Router Link  
`routerLink="/detail/{{hero.id}}"`
- ▶ Add Routing Module  
`import { NativeScriptRouterModule } from "nativescript-angular/router";`

---

# AOT COMPILATION

- ▶ Configure the `tsconfig` files to exclude platform specific modules and components.
- ▶ If it is a native only component, exclude them in `tsconfig.app.json`
- ▶ If it is a web only component, exclude them in `tsconfig.tns.json`

# DEMO

# ECOSYSTEM

---

# WEBPACK CONFIGURATION

- ▶ Create a fast starting app with NativeScript Webpack  
`npm install --save-dev nativescript-dev-webpack`  
`npm install #for dependencies`
- ▶ Run added scripts to build with Webpack optimizations
- ▶ Reduces file I/O
- ▶ Build with webpack, then run  
`tns build ios --bundle`  
`tns run iOS --bundle`

---

# PUBLISHING

- ▶ iOS - <https://bit.ly/2JAP4H1>
- ▶ Android - <https://bit.ly/32twGY9>

**BUT I WANT TO  
DO IT MANUALLY...**

**SHUTUP AND  
USE SIDEKICK**





@Halpin\_io

<http://bit.ly/2JdoNOY>

N

Send Feedback Getting Started Fooa

# Create App

Template Category

General Industry

Project Type

Angular & TypeScript

Template

Blank Drawer Navigation Tab Navigation Master-Detail (with Kinvey) Master-Detail (with Firebase) Enterprise Auth

App Name \* MyNewApp !

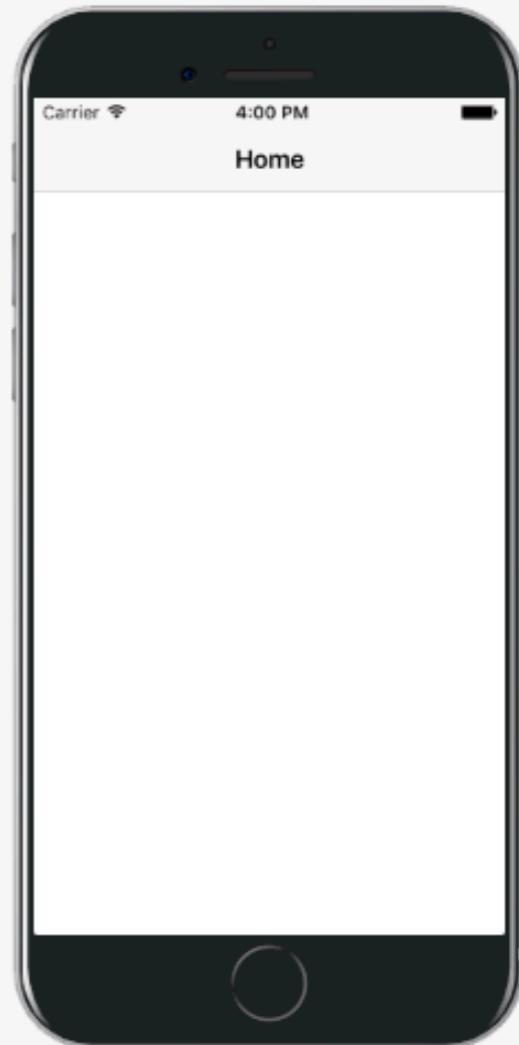
Project Folder \* User/Projects Browse...

Automatically set App ID org.nativescript.mynewapp

This value should match the App ID of your iOS provisioning profile Need help?

Create Cancel

TEMPLATE DESCRIPTION:  
Blank template for NativeScript apps using Angular



The screenshot shows the NativeScript application creation interface. On the left, there's a sidebar with 'Template Category' (General, Industry), 'Project Type' (Angular & TypeScript), and a list of templates: Blank, Drawer Navigation, Tab Navigation, Master-Detail (with Kinvey), Master-Detail (with Firebase), and Enterprise Auth. The 'Blank' template is selected. The main form has fields for 'App Name' (MyNewApp) and 'Project Folder' (User/Projects). A checkbox for 'Automatically set App ID' is checked, with the value 'org.nativescript.mynewapp' displayed below it. A warning message says 'This value should match the App ID of your iOS provisioning profile' with a 'Need help?' link. At the bottom are 'Create' and 'Cancel' buttons. To the right, there's a preview of a NativeScript mobile application on an iPhone, showing a blank white screen with the status bar indicating 'Carrier' and '4:00 PM'. Below the preview is a 'TEMPLATE DESCRIPTION' section with the text 'Blank template for NativeScript apps using Angular'.



<http://bit.ly/2NIqPLs>

The screenshot shows the NativeScript Playground Editor interface. The top navigation bar includes 'Playground' (selected), 'Editor' (active), 'Get started', and a notification bell. Below the bar, the title 'My Playground' is displayed with a save icon. The main workspace has tabs for 'New', 'QR code', 'Save', 'Fork', and 'Download'. On the left, the 'Explorer' sidebar lists project files: 'app' (app-routing.module.ts, app.component.html, app.component.ts, app.css, app.module.ts) and 'home' (home-routing.module.ts, home.component.css, home.component.html, home.component.ts, home.module.ts, main.ts). The 'Components' sidebar under 'ESSENTIALS' shows 'Button' and 'Label'. The central editor area displays the 'home.component.html' template:

```
1 <ActionBar title="Home" class="action-bar">
2 </ActionBar>
3
4 <GridLayout>
5   <ScrollView class="page">
6     <StackLayout class="home-panel">
7       <!--Add your page content here-->
8       <Label textWrap="true" text="Play with NativeScript!" class="h2 description-label"></Label>
9       <Label textWrap="true" text="Write code in the editor or drag and drop components from the Components sidebar." class="h2 description-label"></Label>
10      <Label textWrap="true" text="Scan the QR code with your mobile device and see the result in your phone." class="h2 description-label"></Label>
11    </StackLayout>
12  </ScrollView>
13</GridLayout>
```

To the right of the editor, a sidebar provides introductory text and links:

We have prepared a few interactive tutorials for you!

To help you get started, we would like to guide you through the process of creating your first NativeScript application.

**TUTORIAL 1**  
Getting Started

**TUTORIAL 2**  
Build a Full App

No, thanks

ACE OF SPADES FARM

LOCAL

grown in spokane, wa

HOMESTEAD SALAD  
6 OZ 5 LB 10  
RICH MIX 5  
ARUGULA 5  
SPINACH 5  
MICROGREENS 5  
RASPBERRIES 4  
BEETS 3  
TURNIPS 3  
TOMATOES 5  
POTATOES 4



<http://bit.ly/2FTelom>

**N NativeScript | Marketplace**

Authors | Privacy | [NativeScript.org](https://NativeScript.org)

Plugins Templates Samples [SUBMIT PLUGIN](#)

 Search for plugins

---

## Recently Added

**Jailbreak Detector**  
by shivang-sai | Version 1.0.0  
This plugin checks for root in android devices as well as for jailbreak in IOS devices  


## Recently Updated

**Fancy Alert**  
by walkerrunpdx | Version 3.0.8  
Fancy alerts for NativeScript.  


<http://bit.ly/2RY5An5>

The screenshot shows the Genymotion website homepage. At the top, there's a navigation bar with links for Solutions, Use cases, Pricing, Help, and Blog. To the right are buttons for Contact Us, Trial, and Sign In. Below the navigation is a banner featuring a stylized landscape with clouds and a path. The main headline reads: "Bring your app to perfection with a full set of features and sensors". On the left, a pink call-to-action button says "Simulate all Android device specificities". Below it, two sections explain the software's capabilities: "Genymotion Desktop can simulate any **Android hardware specificities**, allowing you to test your app on a wide range of Android devices in a snap!" and "Simulate **screen size**, different Android version, and much more: **memory capacities, camera, multitouch, etc.**". To the right, there's a screenshot of the Genymotion desktop application. It shows a virtual Android device running an OS with a beach wallpaper. A floating window titled "Create a custom device" is open, showing settings for a device named "Clone - 411". The "Display" section has "Predefined" selected, with "240x320" and "MDPI" highlighted. Other options include "Custom", "Start in full-screen mode", and a dropdown for "System" (Android version 4.1). The "Network mode" is set to "NAT (default)". At the bottom right of the dialog are "CANCEL" and "CONFIRM" buttons.

# CONTACT

@Halpin\_io

---

## GET IN TOUCH!

- ▶ Twitter: [@Halpin\\_io](https://twitter.com/Halpin_io)
- ▶ Website: [halpin.codes.com](http://halpin.codes)

# THANK YOU!