

Training: NativeScript

Course Duration	5 days
Prerequisites	All attendees must have substantial prior experience developing with JavaScript, TypeScript and Angular 2+.
Description	NativeScripting's Comprehensive NativeScript with Angular bootcamp teaches students the basic and advanced skills they need to design, build, and test sophisticated native mobile applications that function well on iOS and Android operating systems.
Objectives	<ul style="list-style-type: none">• Use industry best practices to architect an Angular mobile application with NativeScript• Manage the state of a modern mobile application using the latest methodologies such as the flux pattern• Understand how Angular is different than traditional web development frameworks• Code using new ES6 and TypeScript language features• Develop an application from scratch using NativeScript with Angular• Explore Angular coding and architecture best practices• Understand and use Angular Forms, Observables, Dependency Injection, and Routing• Retrieve, update, and delete data using Angular's Http service• Unit test all the parts of an Angular application including Modules, Components, Services, and Pipes• Create, build, and deploy a NativeScript Angular application using the CLI approach

What you will learn

Getting started

- Tools: CLI
- Tools: Playground intro
- Tools: Sidekick intro
- Tools: NativeScript VS Code Plugin
- project dependencies
- high level app folder structure
- API Server
- Consuming the API

Tracing and Debugging

- Setting up tracing
- Using tracing
- Debugging options
- Debugging commands
- Debugging with Visual Studio Code
- Debugging with Chrome
- Http Debugging

Application Management

- Bootstrapping options
- Application configuration
- Auto injection
- Application internationalization
- Console visualization
- Application Logging
- Error handling
- Application Life Cycle
- iOS UIApplicationDelegate
- Android Activity Events
- Project Status Bar utils

Application State

- Methodology
- Reactive Store and State
- Smart and Presentation Components
- Persisting State
- NgRx intro

Routing and Navigation

- Declarative routing Approach
- Programmatic Routing Approach
- RouterExtensions
- Navigation Service
- NavigationOptions
- Route Guards
- Combining Routing outlet options
- Routing parameters
- Lazy loading
- Navigation Directive

Deeper UI

- What is a frame?
- Using the Page
- View life cycle
- Dialogs
- Modals
- Multiple ListView templates
- FormattedString
- ActionBar
- Local notifications
- Controlling the keyboard
- Using icon fonts
- Lower cased and dashed components
- Programmatic components
- Custom components
- Platform specific declarations
- Screen size qualifiers
- Platform qualifiers
- Orientation qualifiers

Hardware Access

- Using Location
- Accessing the camera

Native Code and Plugins

- tns-platform-declarations
- Utilizing native platform APIs
- Extending the framework with plugins

Testing

- Configuring your project
- Creating tests
- Running tests
- Configuring the Karma server
- Continuous Integration

Preparing for deployment

- AOT
- Configuring WebPack
- Bundling with WebPack
- Bundling Extra Assets
- Advanced WebPack
- iOS Launch screen
- iOS App Icons
- iOS publishing
- Android Launch screen
- Android App Icons
- Android publishing

More UI Widgets

- Placeholder
- Switch
- ListPicker
- DatePicker
- TimePicker
- ImagePicker
- Slider
- Repeater
- WebView
- HtmlView
- Progress
- SearchBar

Contact us at education@progress.com or visit our website progress.com/services/education

Progress is registered trademark of Progress Software Corporation and/or one of its subsidiaries or affiliates in the U.S. and/or other countries. Any other trademarks contained herein are the property of their respective owners.

© 2019 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.