

Setup

- 1) Connect to the wifi (on your laptop and devices)
 - Network name:
 - Password:
- 2) Go to platform.telerik.com and login
- 3) Make sure you have the NativeScript app installed on your Android and/or iOS device(s)





NativeScript Deep Dive

- TJ VanToll | @tjvantoll
- Clark Sell | @csell5
- Sebastian Witalec | @sebawita

Agenda

- NativeScript intro (TJ)
 - Lab #1
- Data, styling, layouts, and more (Clark)
 - Lab #2
- MVVM, modules, extensibility (Sebastian)
 - Lab #3



Follow NativeScript

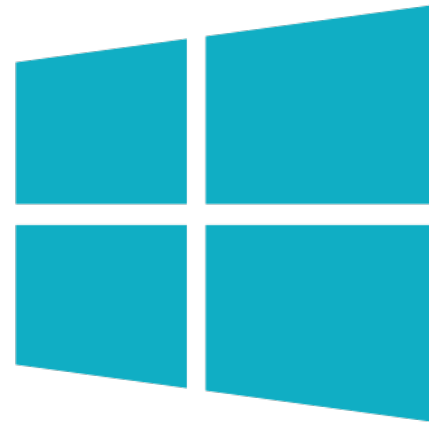


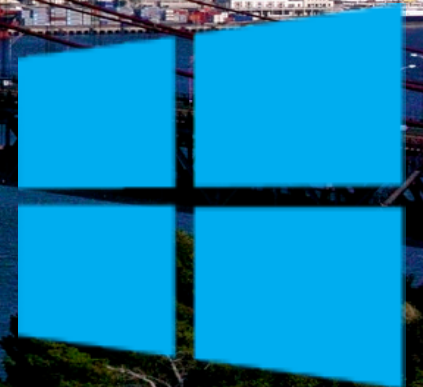
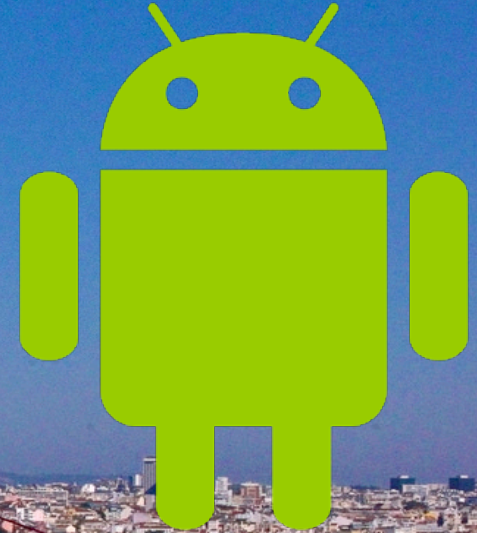
- nativescript.org
- [@nativescript](https://twitter.com/nativescript)
- nativescript.org/blog



What is NativeScript?

- A runtime for building and running *native* iOS, Android, and Windows Phone apps with a single, JavaScript code base





Why NativeScript?

- Skills reuse
 - Standards-based JavaScript, CSS, optionally TypeScript
- Code reuse
 - npm modules, 3rd-party iOS and Android libraries
- Easily use native APIs
 - No wrappers to access native APIs
 - Use native UI elements
- Open source!



Contribute!

(nativescript.org/contribute)

Contributing to NativeScript

Thank you for your interest in contributing to the NativeScript project!

Anyone wishing to contribute to the NativeScript project MUST read & sign the NativeScript [Contribution License Agreement](#). The NativeScript team cannot accept pull requests from users who have not signed the CLA first.

NativeScript is a complex framework, involving cross-platform modules, a Command-Line Interface and platform-specific runtimes. Each of these follows a specific technology, therefore the contribution instructions are different for each.

Please, visit these repositories for detailed contribution guidelines:

[Cross-Platform modules](#)

[Command-Line Interface](#)

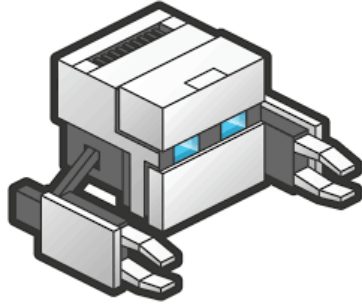
[Android-Runtime](#)

[iOS-Runtime](#)





!=



- No DOM



!=



- No cross compilation



NativeScript Android example

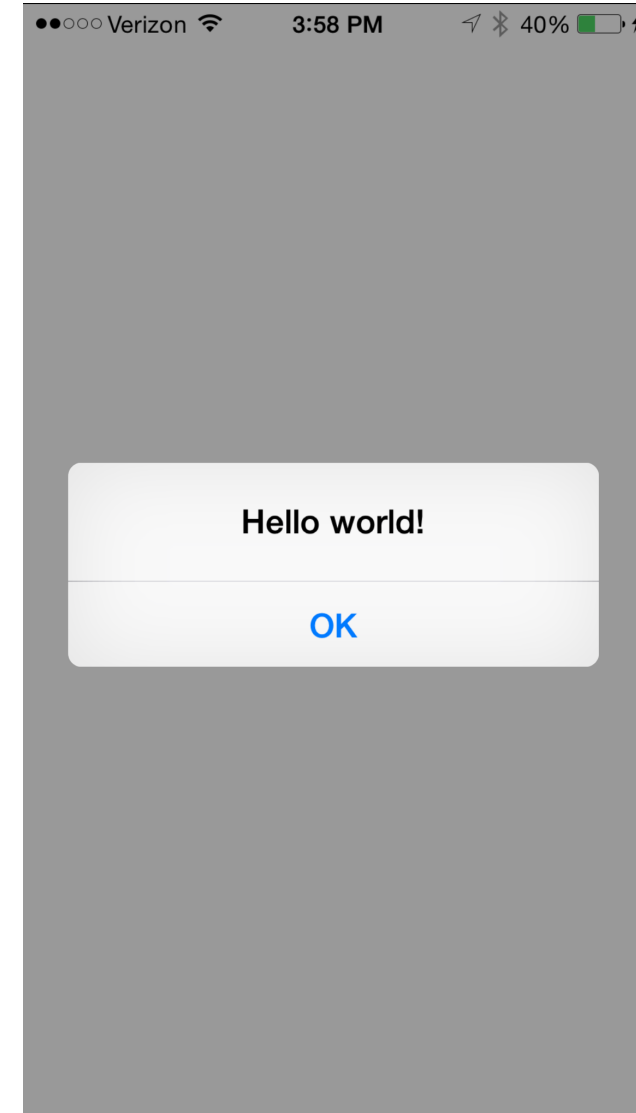
```
var time = new android.text.format.Time();  
time.set( 1, 0, 2015 );  
console.log( time.format( "%D" ) );
```

Output: "01/01/15"



NativeScript iOS example

```
var alert = new UIAlertView();  
alert.message = "Hello world!";  
alert.addWithTitle( "OK" );  
alert.show();
```



UIAlertView Class Reference

Apple Inc. [US] https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UIAlertView_C...

IOS Developer Library — Pre-Release

UIKit Framework Reference > UIAlertView Class Reference

Search iOS Developer Library

Language: [Swift](#) [Objective-C](#) [Both](#) On This Page Options

UIAlertView

Setting Properties

[delegate](#) *Property*

[alertViewStyle](#) *Property*

[title](#) *Property*

[message](#) *Property*

[visible](#) *Property*

Configuring Buttons

[- addButtonWithTitle:](#)

[numberOfButtons](#) *Property*

[- buttonTitleAtIndex:](#)

[- textFieldAtIndex:](#)

[cancelButtonIndex](#) *Property*

[firstOtherButtonIndex](#) *Property*

Displaying

[- show](#)

Tasks

- Creating Alert Views
- Setting Properties
- Configuring Buttons
- Displaying
- Dismissing

Constants

- UIAlertViewStyle

Related Documentation

- Alert Views in UIKit User Interface Catalog

Related Sample Code

- AdvancedURLConnections
- GKTapper
- MVCNetworking
- SquareCam
- URLCache

Feedback

```
var alert = new UIAlertView();
alert.message = "Hello world!";
alert.addButtonWithTitle( "OK" );
alert.show();
```



How does this work?



NativeScript and JS VMs

- NativeScript runs JavaScript on a JavaScript VM
 - JavaScriptCore on iOS
 - V8 on Android
 - JavaScriptCore on Windows



```
var time = new android.text.format.Time();  
time.set( 1, 0, 2015 );  
console.log( time.format( "%D" ) );
```

- Runs on V8

```
var alert = new UIAlertView();  
alert.message = "Hello world!";  
alert.addButtonWithTitle( "OK" );  
alert.show();
```

- Runs on JavaScriptCore



**So do you only write native
code?**

No



NativeScript modules

- NativeScript-provided modules that provide cross-platform functionality.
- There are dozens of them and they're easy to write yourself.
- NativeScript modules follow Node module's conventions (CommonJS).



NativeScript file module

```
var fileSystemModule = require( "file-system" );  
new fileSystemModule.File( path );
```



`new java.io.File(path);`



`NSFileManager defaultManager();`
`fileManager.createFileAtPathContentsAttributes(path);`

HTTP module example

```
var http = require( "http" );  
http.getJSON( "https://api.myservice.com" )  
  .then(function( result ) {  
    // result is JSON Object  
  });
```



Custom NativeScript modules

```
// device.ios.js
module.exports = {
    version: UIDevice.currentDevice().systemVersion
}
```

```
// device.android.js
module.exports = {
    version: android.os.Build.VERSION.RELEASE
}
```



Using the custom device module

```
var device = require( "./device" );  
console.log( device.version );
```



But how do I turn this into an app?



Two ways to use NativeScript

1)  **Telerik**PlatformSM

2) `npm install -g nativescript`





Telerik PlatformSM

<http://telerik.com/platform>

- Backend-as-a-service
 - Push notifications, cloud data, file storage, and more
- Analytics
- AppBuilder
 - Cloud builds (build iOS apps on Windows, Windows Phone apps on a Mac)
 - NativeScript debugging and tooling
- Automated app testing
- And more!





<https://www.telerik.com/purchase/platform>

Telerik Platform 30 Day Trial

FREE

Start now

Try everything Telerik Platform
has to offer, FREE, for 30 days

 **All Platform Services**

 **Web, Hybrid & Native UI**

Unlimited trial support


Telerik Platform Developer

\$39 /month/user
requires annual upfront payment

Subscribe

Ideal for tinkerers and hobbyists just
getting started with mobile app
development

 **Core Platform**

 **Hybrid UI**

Limited web support

Telerik Platform Professional

\$79 /month/user
requires annual upfront payment

Subscribe

For professional developers and small
teams building full-featured employee and
consumer apps

 **Core Platform**

+ Advanced Cloud Services
+ Direct App Store Deployment

 **Hybrid & Native UI**

Limited web support

**MOST
POPULAR**

Telerik Platform Business


\$149 /month/user
requires annual upfront payment

Subscribe

For developers and large teams
building advanced apps
connected to business data

 **Pro Platform**

+ Active Directory Integration
+ Enterprise Data Connectors
+ Private App Distribution

 **Web, Hybrid & Native UI**

Unlimited web support

Telerik AppBuilder IDE Options

- In-Browser Client
- Visual Studio Extension
- Sublime Text Package
- Command-Line Interface



NativeScript CLI

- Free and open source
- <https://github.com/nativescript/nativescript-cli>



NativeScript CLI requirements

- <https://github.com/nativescript/nativescript-cli#system-requirements>



- JDK, Apache Ant, Android SDK



- Xcode, Xcode CLI tools, iOS SDK



Starting a new project

```
$ npm install -g nativescript
```

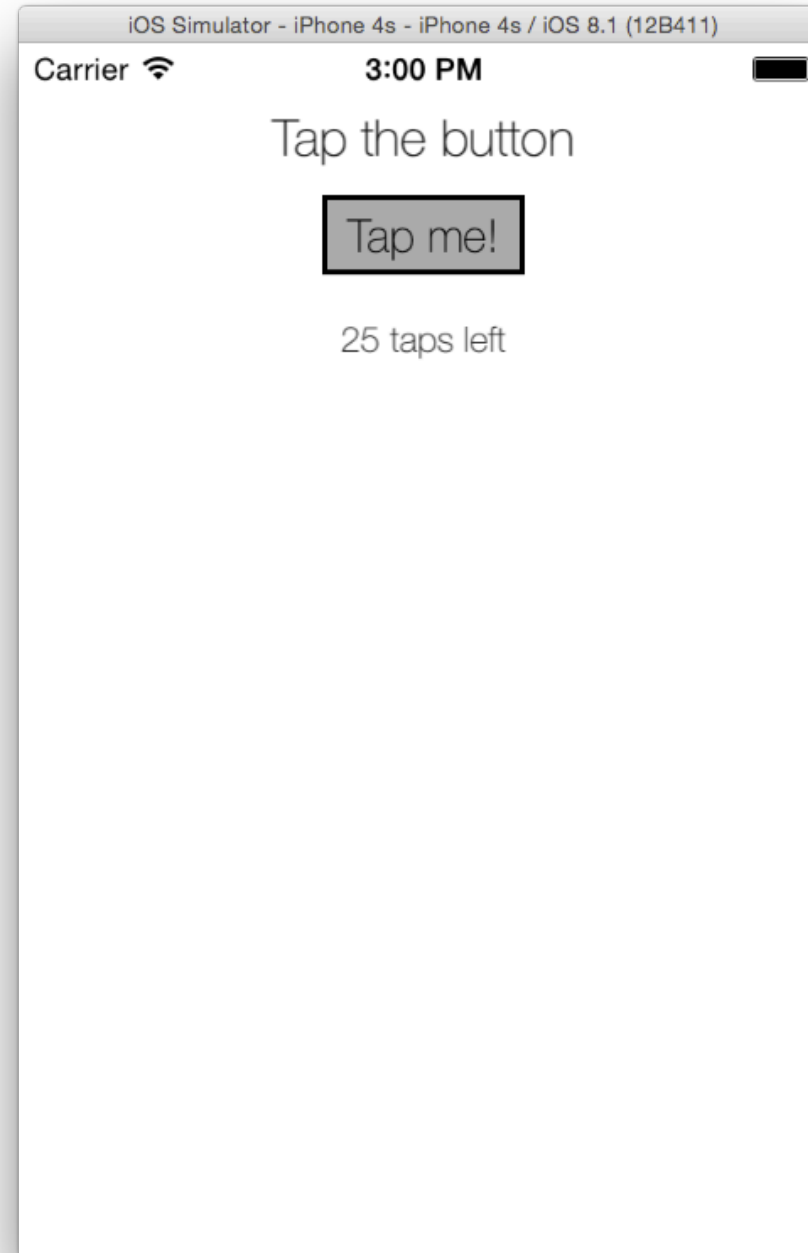
```
$ tns create hello-world
```

```
$ cd hello-world
```



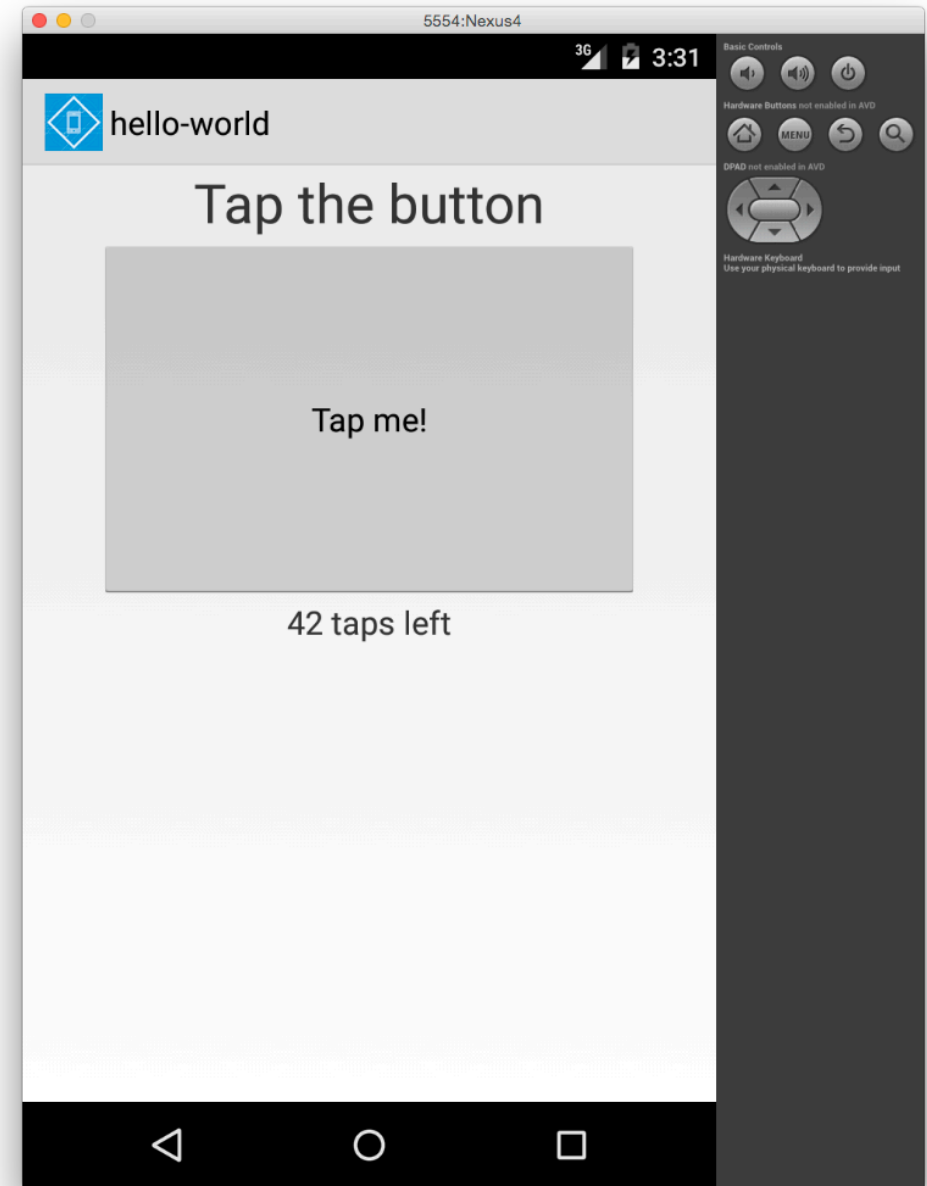
Running on iOS

```
$ tns platform add ios  
$ tns run ios --emulator
```



Running on Android

```
$ tns platform add android  
$ tns run android --emulator
```



Creating Platform apps

- Demo



```
.
└─ app
    ├── App_Resources <-- icons, splash screens, config files
    │   ├── Android
    │   └── iOS
    ├── app.css
    ├── app.js
    ├── main-page.css
    ├── main-page.js
    ├── main-page.xml
    ├── node_modules <-- npm modules
    │   └── ...
    ├── package.json
    ├── tns_modules <-- NativeScript modules
    │   └── ...
    └─ platforms
        ├── android
        └── ios
```



app.js

```
var application = require( "application" );  
application.mainModule = "main-page";  
application.start();
```



Pages

- XML markup structure
- Elements (e.g. `<Page>`, `<Label>`) are NativeScript modules

```
<Page>  
    <Label text="hello world" />  
</Page>
```



Custom XML Components

Example: Code-Only Custom Component

This sample `main-page.xml` uses two custom components defined in separate declarations in the `xml-declaration` directory. The custom controls are wrapped horizontally.

XML

```
<Page
  xmlns:customControls="app/xml-declaration/mymodule"
  xmlns:customOtherControls="app/xml-declaration/mymodulewithxml">
  <WrapLayout>
    <customControls:MyControl />
    <customOtherControls:MyControl />
  </WrapLayout>
</Page>
```

<http://docs.nativescript.org/ui-with-xml#custom-components>



Questions?



Getting started

- Clone ...?

