





# Titanium Tutorial



# Your instructors



**Phil Smithson**

[Titanium Certified Expert](#)

Co-founder:

UXMNL/Foolproof Labs

[@foolprooflabs](#)

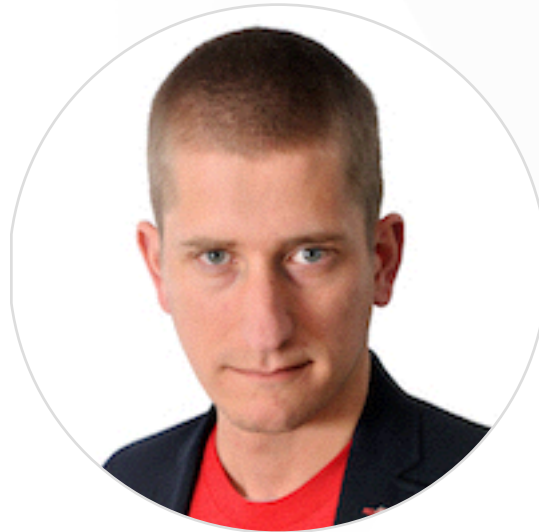


**Rocky Coronado**

Foolproof Labs

Mobile App Specialist

# With thanks to...



**Fokke**

Titan

Certified Instructor

Self-employed

[@FokkeZB](#)

# Program

- **What is Titanium?**
  - Finding resources
  - Setting up
- **What is Alloy?**
- **Build your first app**
  - Using TiShadow
  - Step by step assignments
- **What's next?**

# Yeah, yeah but when do we eat?

- **Coffee** - available in the kitchen, help yourself
- **Lunch** - will be served
- **CR** - through the kitchen or upstairs
- **Smoking** - on the balcony

# Twitter

- **Tweet using hashtag:**
  - #titanium
- **If you can, please include:**
  - @appcelerator
  - @fokkeZB
  - @foolprooflabs



**Titanium**

... and what is Appcelerator?



Founded  
In **2007**

**Mountain View**

London  
Tokyo  
Beijing

**150**  
Employees

**\$50M**

Mayfield,  
Translink, Storm,  
Sierra, eBay, Red  
Hat



# About Appcelerator

**Gartner.**

**Gartner 2013  
Magic Quadrant**

Highest placement in  
both vision and  
execution thereof.

# You've probably already used Titanium



**600K**

Developers



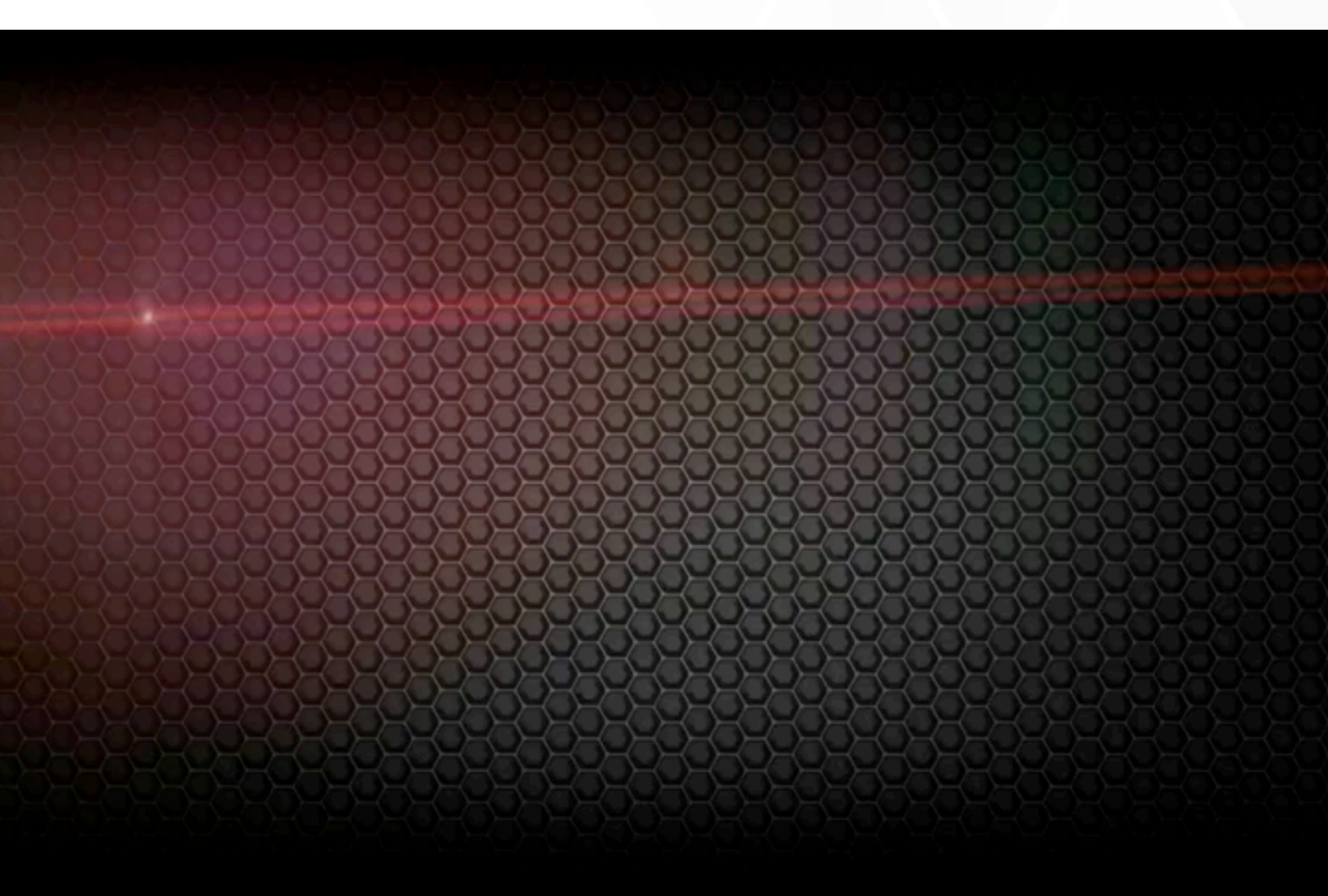
**70K**

Apps



**230M**

Devices



# Native UI





# Resources

Where to get your information?

# Resources



- [tidev.io/resources](http://tidev.io/resources)
- [appcelerator.com/developers](http://appcelerator.com/developers)
- [docs.appcelerator.com](http://docs.appcelerator.com)
- [Search “Manila Appcelerator Titanium Group” on Facebook](#)
- <https://plus.google.com/u/0/communities/103721188616163608038>
- [Manila Titanium Meetup Group](#)



# Set-up

Getting Ready!

# What makes Titanium?

- NodeJS CLI
- SDK
- Titanium Studio (optional)
- Target platform SDKs





# Assignment

- Sign up and [download Studio](#)
- Install Studio and its dependencies
  - JAVA, NodeJS, CLI, SDK
- Use the Platform Configuration Tool
- Create the **Classic / Default Project**
- Run in a simulator or device
- Investigate **Resources/app.js**



# Alloy

Conventions vs Chaos

# This might be confusing



# Classic vs Alloy

## CLASSIC: APP.JS

```
var win = Ti.UI.createWindow({backgroundColor:"white"});  
win.open();
```



## ALLOY: INDEX.XML

```
<Alloy><Window id="win" /></Alloy>
```

## ALLOY: INDEX.TSS

```
"#win": { backgroundColor: 'white' }
```



## ALLOY: INDEX.JS

```
$.win.open();
```

# Few more things

Widgets

Project Configuration File

Built-ins

Controller-less views

Sync adapters

Build Configuration File

Backbone.js

Themes

Underscore.js

Moment.js

Source Maps

Require

Dynamic Styling

Data-binding

# Open Source Widgets\*

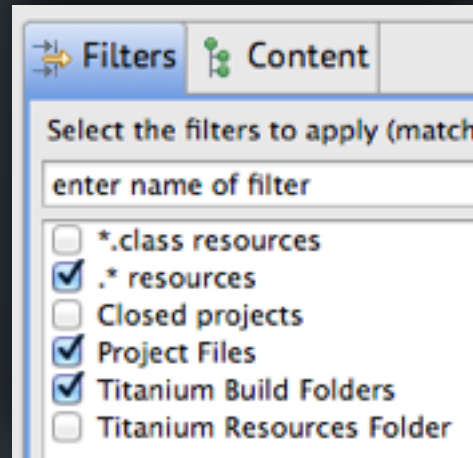
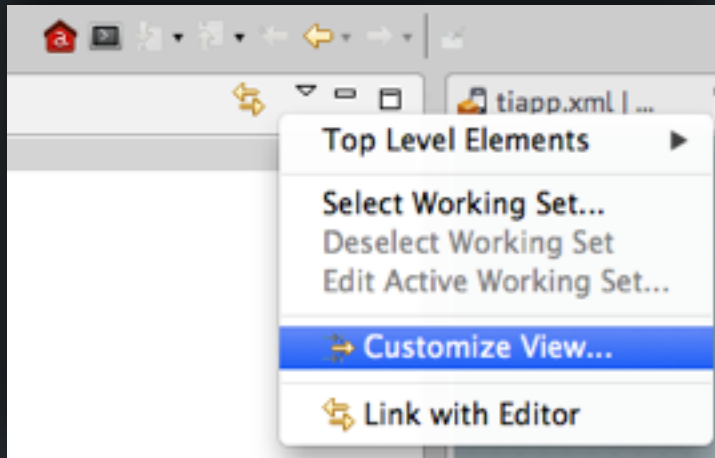
\* And Modules



<http://gitt.io>

# Assignment

- Create the **Two-tabbed Alloy Application**
- Run in a simulator or device
- Compare **index.xml/tss/js** with classic
- Unhide the Resources folder



- Compare **app.js** & **iphone/alloy/controllers/index.js**

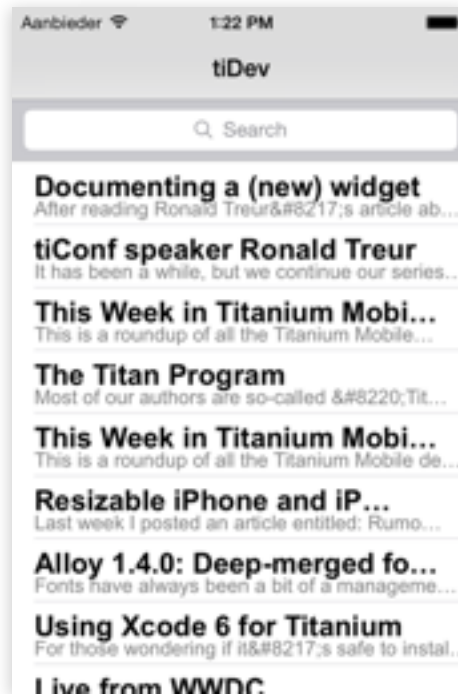


# Zero to App

What you will build today



# tiDev RSS reader

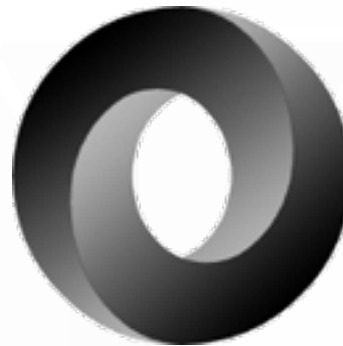


# Ingredients

- JSON feed: [blog.foolprooflabs.com/feed/json](http://blog.foolprooflabs.com/feed/json)
- Repo: [github.com/foolprooflabs/Tutorial](https://github.com/foolprooflabs/Tutorial)

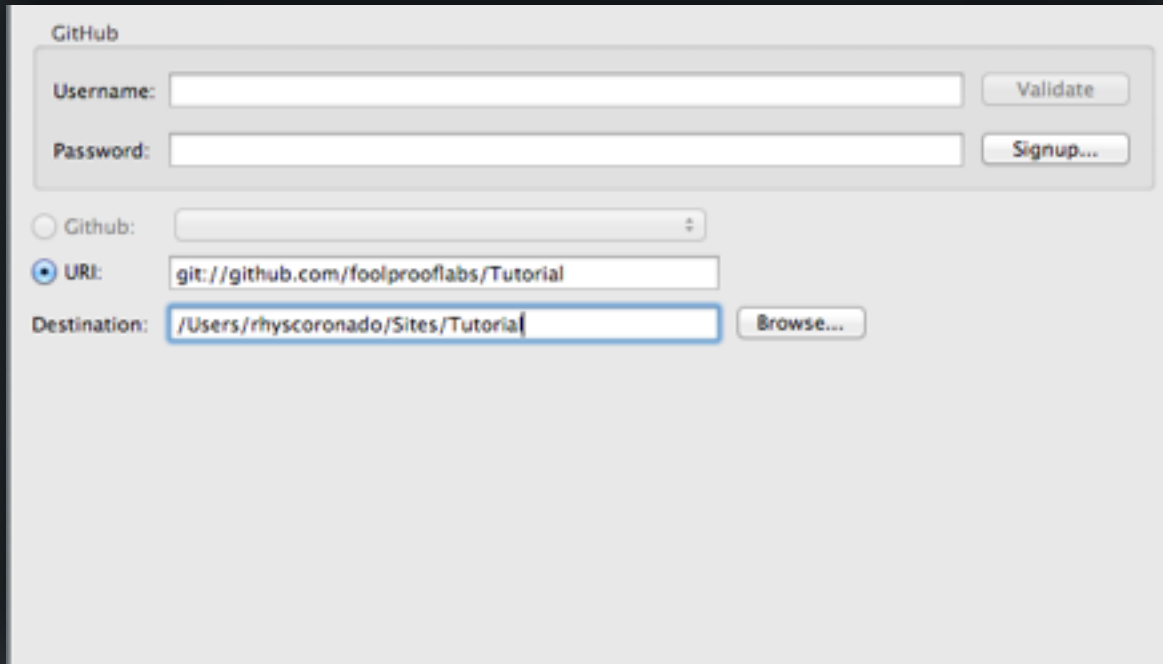


VS



# Assignment

- Import a clone of the [repository](#)
  - File > Import > Git > .. as New Project



The screenshot shows a 'GitHub' dialog box with the following fields and buttons:

- Username:** A text input field with a 'Validate' button to its right.
- Password:** A text input field with a 'Signup...' button to its right.
- Repository Selection:** Two radio buttons are present. The first is labeled 'Github:' and is unselected. The second is labeled 'URI:' and is selected (indicated by a blue dot).
- URI:** A text input field containing the value 'git://github.com/foolprooflabs/Tutorial'.
- Destination:** A text input field containing the value '/Users/rhyscoronado/Sites/Tutorial'. To its right is a 'Browse...' button.

- Run the app
- Copy @Slides/slides.pdf to your Desktop



# TiShadow

Rapid Application Development



**“If you're not using TiShadow  
in your Titanium projects, you  
might be wasting hours per  
week on builds. Love it.”**

**@kevinwhinnery**

# Compile process

- Compiling for Alloy
- **Compiling for Titanium**
- **Compiling for Xcode/Android**
- **Deploying to simulator/device**
- **Starting the app**
- Running your code

# Assignment

- Install [TiShadow](#)

```
~ $ [sudo] npm install -g tishadow
```

- Run an app using CLI and TiShadow Express

```
~/Tutorial $ ti build -p ios --shadow
```

- Change **backgroundColor** to **red** and save





# Window

Creating a Window



# Views

- UI components are name-spaced **Ti.UI.\***
- Most UI components extend [Ti.UI.View](#)
- Create components: **Ti.UI.create<View>()**
- In [Alloy](#) **<View>** ➤ **Ti.UI.create<View>()**
- XML attributes and TSS properties are merged and passed to the create-method.
- TSS can select by **tag**, **id** **or** **class**.
- JS can refer to **\$.<id>**

# Classic vs Alloy

## CLASSIC: APP.JS

```
var win = Ti.UI.createWindow({backgroundColor:"white"});  
win.open();
```



## ALLOY: INDEX.XML

```
<Alloy><Window id="win" /></Alloy>
```

## ALLOY: INDEX.TSS

```
"#win": { backgroundColor: 'white' }
```



## ALLOY: INDEX.JS

```
$.win.open();
```

# Assignment

- Reset any changes:

```
~/Tutorial $ git reset --hard
```

- Switch to **window** branch.

```
~/Tutorial $ git checkout window
```

- Run the app.
- Add a Label saying **Hello world**.
- Give the label a class.
- Use the class to make the color red.





# ListView

Not the most easy component

# ListViews

- Docs > Search: [guide: ListViews](#)
- Docs > Search: [UI.ListView](#)
- “Crossing the bridge” has a toll
- **ListView** vs [TableView](#)
- Alloy does most of the hard work

# Assignment

- Study the [ListViews](#) guide.
- Compare the [API doc](#) classic & Alloy examples.
- Reset & Switch to the **listview** branch.
- Run the app.
- Try the first API doc Alloy example JS & TSS.
- Compare **Resources/iphone/alloy/controllers/index.js** with the classic examples.
- Remove the blank icons, align the text left, reduce the sections to one and remove the [headerTitle](#) of that section.

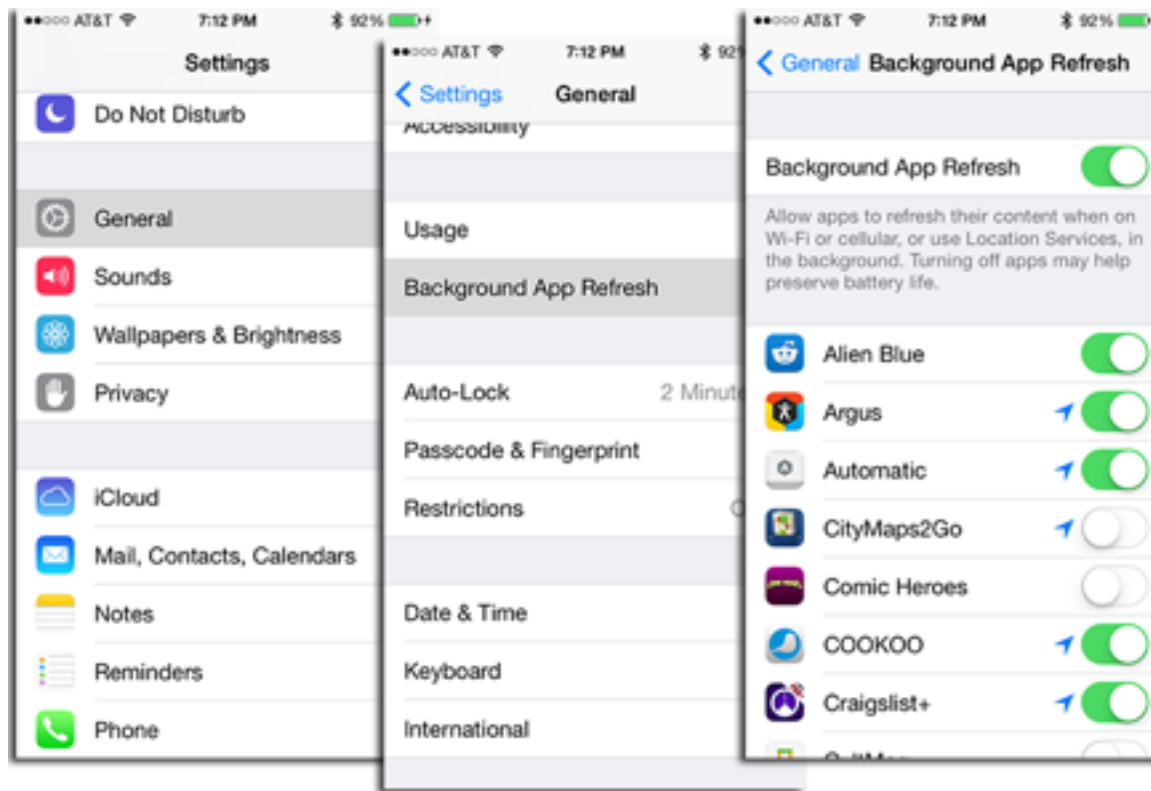


# NavigationWindow

Stacking Windows

# NavigationWindow

- Docs > Search: [NavigationWindow](#)
- iOS only





# Event Handling

- Docs > Search: [Alloy XML Markup](#)
- HTML-like **on[Event]** attribute

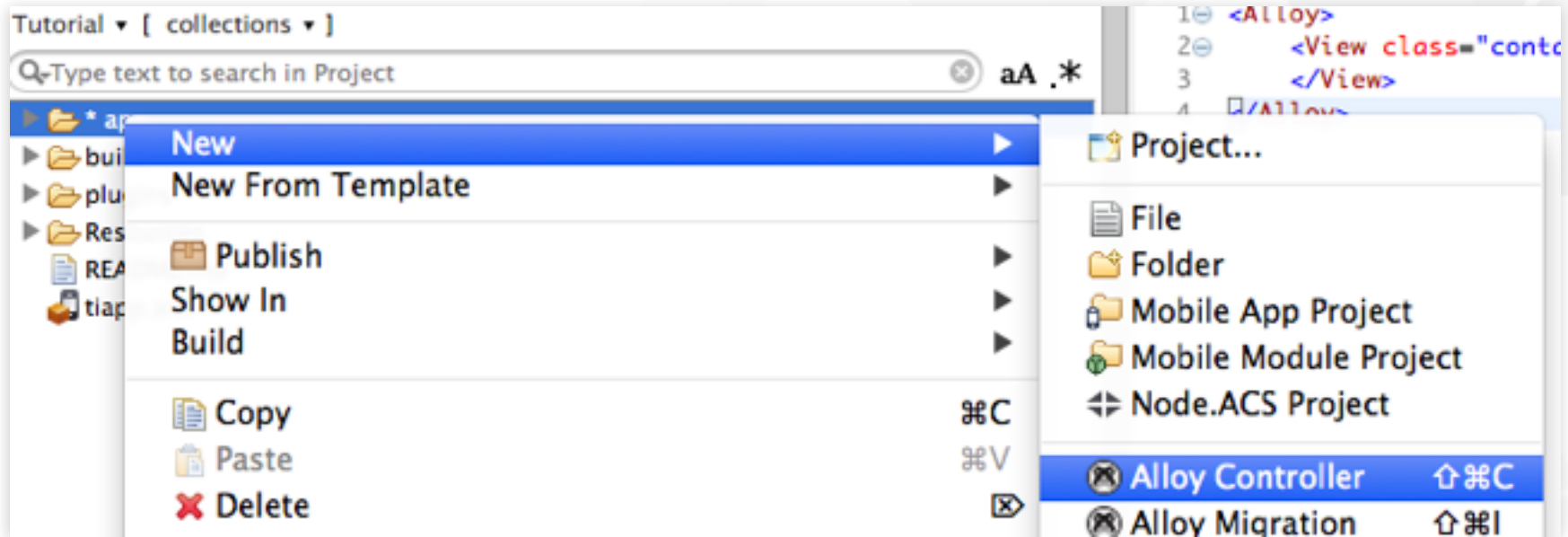
## INDEX.XML

```
<Button onClick="talkToMe" />
```

## INDEX.JS

```
function talkToMe(e) {  
    alert("hello world");  
}
```

# Creating Controllers



```
~/Tutorial $ alloy generate controller detail
```

# Using Controllers

- Use [<Require>](#) to create + add in XML
- Use [Alloy.createController](#) in JS

## INDEX.XML

```
<Window id="win">  
  <Require src="detail" />  
</Window>
```

## INDEX.JS

```
var detailController = Alloy.createController('detail');  
var rootView = detailController.getView();  
$.win.add(rootView);
```

# Assignment

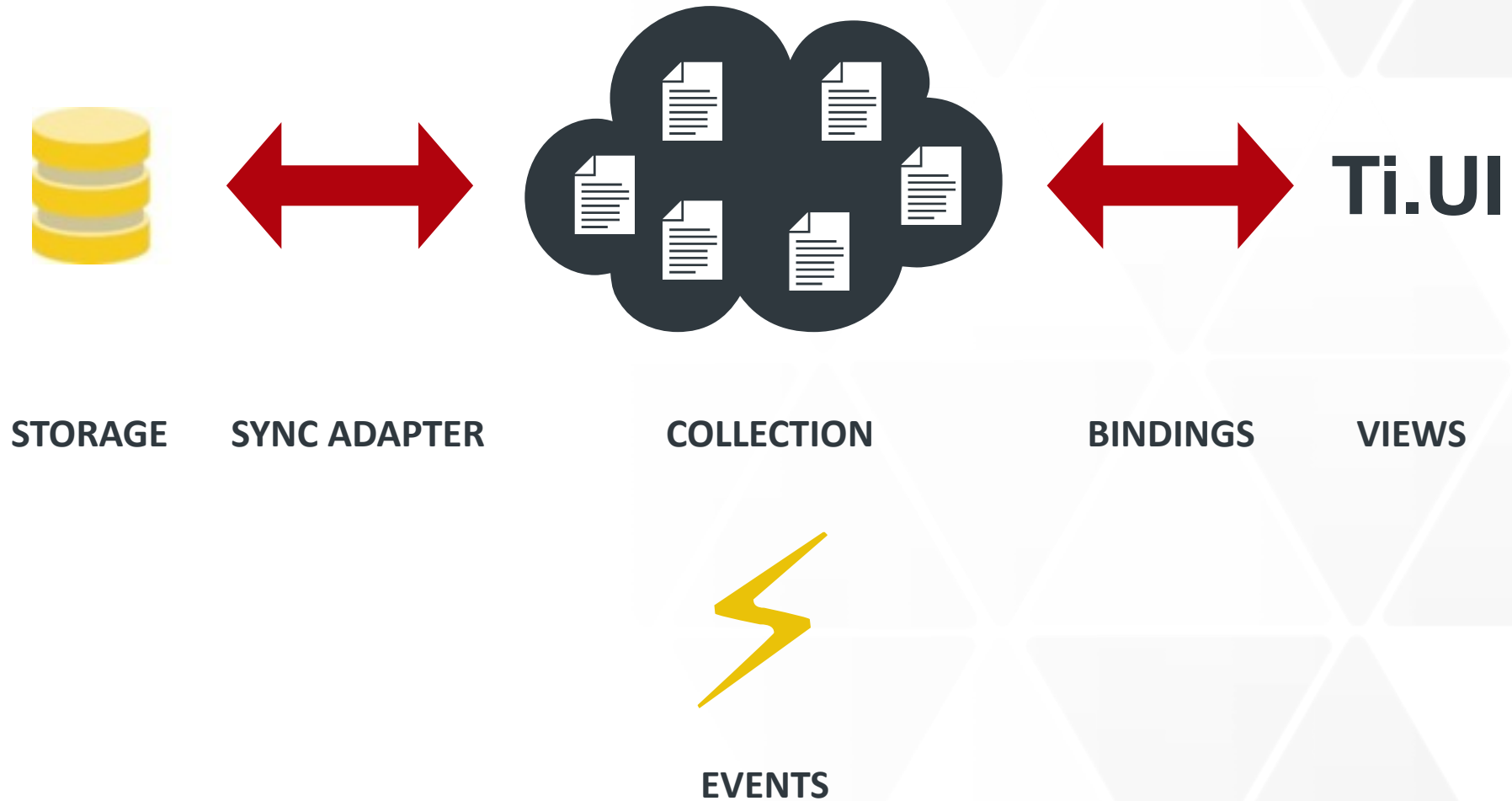
- Study the [NavigationWindow](#) API docs.
- Reset & Switch to the **navwin** branch and run it.
- Wrap the **Window** in a **NavigationWindow**  
For Android add **module="xp.ui"** attribute.
- Add an [itemclick](#) listener to the ListView.
- [Create](#) a new **detail** controller and make it a Window. Give both windows a title.
- On click, [create](#) a detail controller instance and [open](#) the window via NavigationWindow.
- Study **lib/xp.ui.js**. What happens on Android?



# Collections

Models with brains

# BACKBONE.JS





- Docs > Search: [Alloy Models](#)
  - **Note:** Older version (0.9.2)
- In Titanium JS has no DOM
- Instantiate a global collection:  
**Collection** tag
- [Bind](#) a collection to a loop-able view:  
**dataCollection** attribute
- Bind a model attribute to a child view property:  
**{<attribute>}** value

# Assignment

- Study the [Alloy Data Binding guide](#).
- Reset & Switch to the **collections** branch and run it.
- Bind the **feed** collection to the **ListSection** so that the list is populated showing the **title** and **excerpt** of the items.
- Study **models/feed.js** and **lib/alloy/sync/json.js**. How do they work? And why that second path?





# WebView

Titanium > Phonegap

# WebView

- Docs > Search: [guide: Communication](#)
- Docs > Search: [UI.WebView](#)
- Display local, remote and “inline” HTML
- Communicate via events and [evalJS](#)
- Displaying formatted text native is [complex](#)
- Displaying HTML in WebView is easy :)

# Assignment

- Study the [WebView](#) docs.
- Reset & Switch to **webview** branch and run it.
- Study **index.xml** to see how we expose the id of the model via [itemId](#).
- Pass the model to the detail controller.
- Add a WebView to the detail Window and use the model to populate it.
- Use the model to set the **title** of the detail Window.

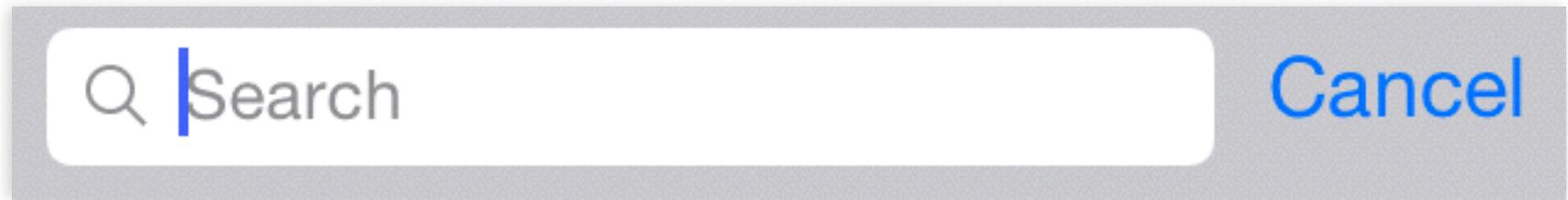


# SeachBar

Filtering our feed

# SearchBar

- Docs > Search: ListViews > [Search Text and View](#)
- Docs > Search: [UI.ListView](#) (last examples)
- Docs > Search: [UI.SearchBar](#)



- **ListView.searchView** works together with **ListView.searchText** & **ListItem.searchableText**

# Assignment

- Study the last two [ListView](#) examples.
- Reset & switch to **searchbar** and run it.
- Add a [SearchBar](#) (also for Android) to the list to search on the **excerpt**.
- If you have time left, try to search on the **content**, after using the [dataTransform](#) attribute to remove HTML tags from it.

**HINT:** [Google](#) is your friend.



# Artwork

Density Hell

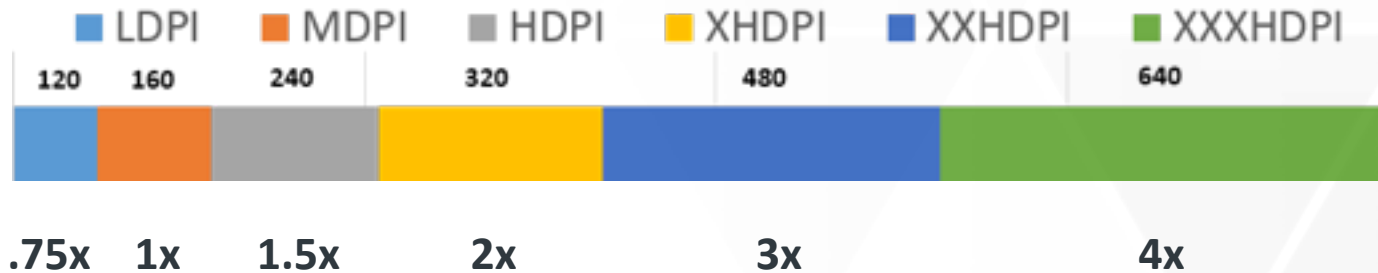
## NON-RETINA



## RETINA (2X)



### Android DPI Categories





# Icons & Splashes



IOS Device	Purpose	Dimensions	DPI	File name	Titanium folder location
iPhone/iPod retina (iOS 7 and later)	App icon	120 x 120	72	appicon-60@2x.png	Resources or Resources/iphone
iPad non-retina (iOS 7 and later)	App icon	76 x 76	72	appicon-76.png	Resources or Resources/iphone
iPad retina (iOS 7 and later)	App icon	152 x 152	72	appicon-76@2x.png	Resources or Resources/iphone
iPhone/iPod non-retina	App icon	57 x 57	72	appicon.png	Resources or Resources/iphone
iPhone/iPod retina (iOS 6 and prior)	App icon	114 x 114	72	appicon@2x.png	Resources or Resources/iphone
iPad 1 & 2 (iOS 6 and prior)	App icon	72 x 72	72	appicon-72.png	Resources or Resources/iphone
iPad 3 (iOS 6 and prior)	App icon	144 x 144	72	appicon-72@2x.png	Resources or Resources/iphone
Universal non-retina (iOS 7 and later)	Spotlight & settings	40 x 40	72	appicon-Small-40.png	Resources or Resources/iphone
Universal retina (iOS 7 and later)	Spotlight & settings	80 x 80	72	appicon-Small-40@2x.png	Resources or Resources/iphone
iPhone/iPod non-retina	Spotlight & settings	29 x 29	72	appicon-Small.png	Resources or Resources/iphone
iPhone/iPod retina (iOS 6 and prior)	Spotlight & settings	58 x 58	72	appicon-Small@2x.png	Resources or Resources/iphone
iPad 1 & 2 (iOS 6 and prior)	Spotlight & settings	50 x 50	72	appicon-Small-50.png	Resources or Resources/iphone
iPad 3 (iOS 6 and prior)	Spotlight	100 x 100	72	appicon-Small-50@2x.png	Resources or Resources/iphone
iPad 3 (iOS 6 and prior)	Settings	58 x 58	72	appicon-Small@2x.png	Resources or Resources/iphone
iPhone non-retina	Splash screen	320 x 480	72	Default.png	Resources or Resources/iphone
iPhone 4/4S iPod touch 4th generation	Splash screen	640 x 960	72	Default@2x.png	Resources or Resources/iphone
iPhone 5 iPod touch 5th generation	Splash screen	640 x 1136	72	Default-568h@2x.png	Resources or Resources/iphone
iPad 1 & 2 landscape	Splash screen	1024 x 748	72	Default-Landscape.png	Resources or Resources/iphone
iPad 1 & 2 portrait	Splash screen	768 x 1004	72	Default-Portrait.png	Resources or Resources/iphone
iPad 3 landscape	Splash screen	2048 x 1496	72	Default-Landscape@2x.png	Resources or Resources/iphone
iPad 3 portrait	Splash screen	1536 x 2008	72	Default-Portrait@2x.png	Resources or Resources/iphone
Any	iTunes Artwork	512 x 512	72	iTunesArtwork (no ext)	Resources or Resources/iphone

# Platform & Density specific folders

```
assets/some_dir/some.file  
assets/android/some_dir/some.file  
assets/iphone/some_dir/some.file
```

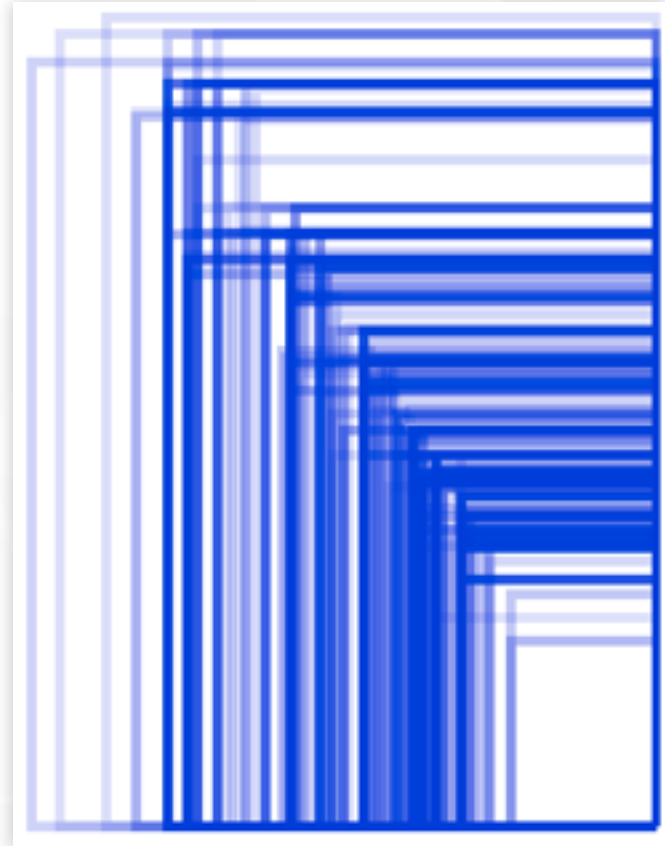
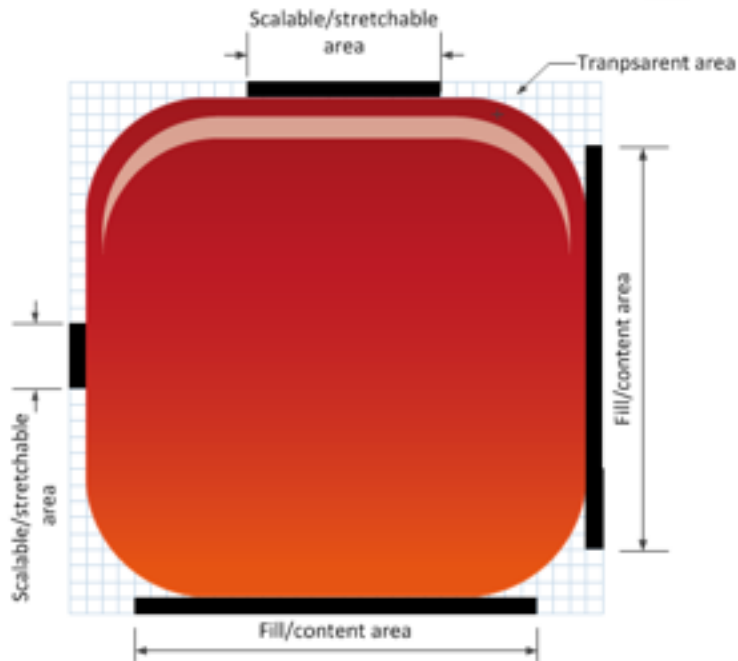
```
assets/android/images/res-xhdpi/some.file  
  
platform/android/res/drawable-xhdpi/some.file  
platform/android/res/drawable-nl-port-xhdpi/some.file
```

```
assets/iphone/images/image@2x.png  
assets/images/image.png  
assets/android/images/res-mdpi/image.png  
assets/android/images/res-xhdpi/image.png
```



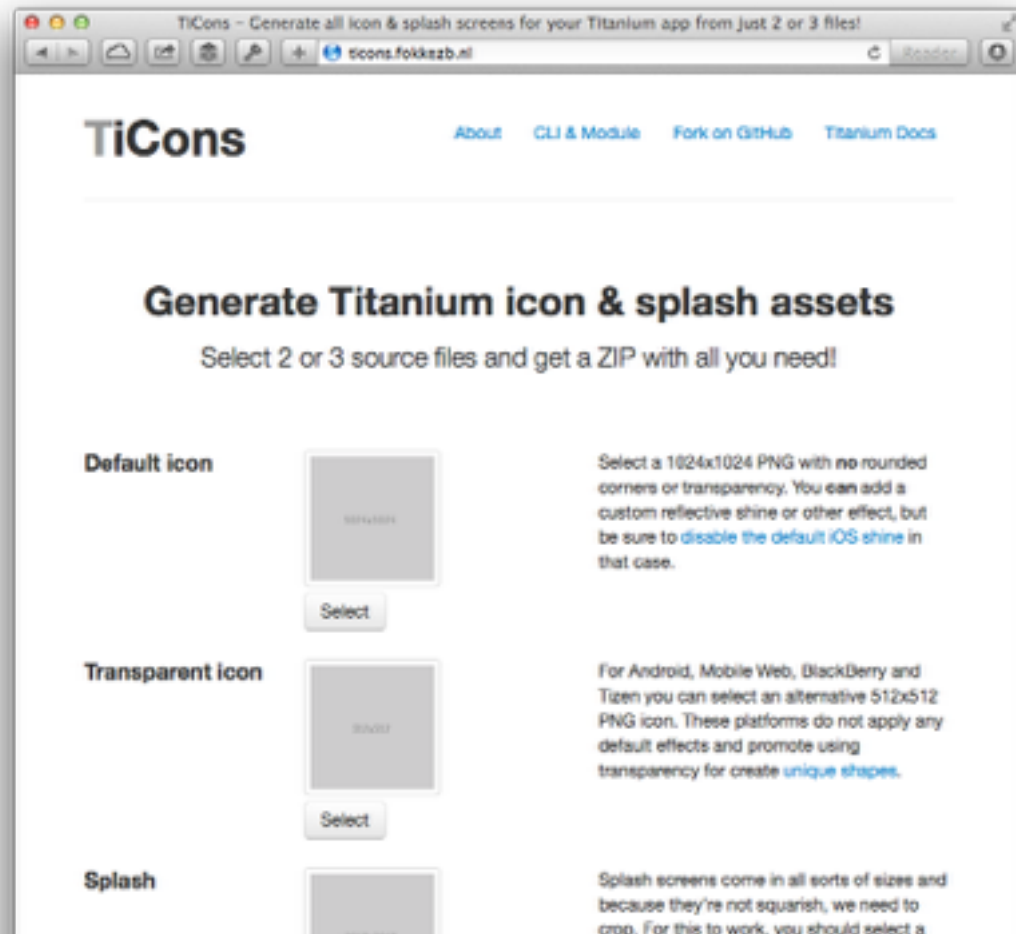
BEST  
PRACTICE

# Android 9-Patch



# No more handwork

```
~ $ [sudo] npm install -g ticons
```



# Assignment

- Study the [Icons and Splash Screens guide](#)
- Install the ticons [CLI](#) or visit the [site](#).

```
~ $ [sudo] npm install -g ticons
```

- Reset & Switch to **artwork** branch and run it.
- Generate icons and splashes using the CLI or site on the **logo.png** image.
- For splashes, find out how to prevent the logo from being cropped.



**You are a TiDev now!**



# What's next?

Join the community

# Join the community

- Follow [@appcelerator](#) & [@TiDevIO](#).
- Follow [@FokkeZB](#) & [@JasonKneen](#).
- Find a local Titanium Meetup on [DevMap](#).
- Go to a [tiConf](#) on your continent.
- Find help on the [Q&A](#).
- Find widgets & modules on [gitTio](#).
- Share your libraries & widgets on GitHub.



# Get certified



[training.appcelerator.com/get-certified](https://training.appcelerator.com/get-certified)



**Thank you!**



# Bonus Assignment

- Reset & Switch to **master** branch and run it.
- Fine-tune the app! For example:
  - Replace the list title by the tiDev logo.
  - Add the date and category to the list view.
  - Add a button to open a blog in the browser.
  - Add a button for the user to share a blog.
  - Be creative!

