


# APPCELERATOR TITANIUM

## AN INTRODUCTION

Bryan Dent | dent.ly | @dently | bryan@dent.ly

Sponsored by

# ABOUT ME

- 15+ years of Development experience
- Primarily Microsoft platform up until 2012
- Employed or contracted for the following companies:
  - SYKES
  - PriceWaterhouseCoopers
  - L'Oréal Paris
  - Charles River Laboratories International, Inc
- First Mobile App in 2010
- Co-Founded Dently in April 2012
- Fan of rock & heavy metal

# APPCELERATOR PLATFORM

- **Analytics -** Development Lifecycle, Cloud & API Consumption, Adoption & Usage
- **Enterprise Connectors -** Oracle, SAP, Salesforce & more
- **Public Connectors -** Google+, Facebook, Twitter, Instagram & more
- Security, Caching & Synchronization
- Data Transformation, Payload Optimization & Data Orchestration
- Extensible, Open Platform & built on NodeJS

**Free & Paid Versions**

# TITANIUM

- Titanium is a **cross-platform** development environment where you can build **iOS**, **Android**, **BlackBerry** and **Hybrid/HTML5** apps
- Titanium apps are written in **JavaScript**
- Your JavaScript interfaces with **native controls** through an abstraction layer (you're not building a webpage)
- Titanium features an Eclipse-based IDE called **Titanium Studio**
- Titanium has an MVC framework called **Alloy**, and Appcelerator offers **Cloud Services** to help bootstrap your app
- Titanium is **free** and **open-source**

# IOS AND ANDROID DEVELOPMENT

- With **Android**, you write native apps in **Java**
- With **iOS**, you write native apps in **Objective-C**
- With **Titanium**, you write cross-platform apps in **JavaScript**, that run on Android, iOS and other platforms

# HOW IT WORKS

- You write code in JavaScript
- Your JavaScript code is minified and optimized during the build process, but still **evaluated as JavaScript at runtime** on the platform
- At runtime, your application has 3 major components:
  - JavaScript source code
  - Titanium API implementation in the native OS
  - JavaScript interpreter: V8 (Android) or JavaScriptCore (iOS)

# HOW IT WORKS

- The JavaScript interpreter runs your JavaScript code in an environment with proxies for the native objects (windows, controls, etc)
- `Ti.UI.createTextField()` creates a `UITextView` on iOS and a `TextView` on Android
- You are **not** creating a webpage with HTML and CSS (which is how PhoneGap works), you build UI through code or the Alloy MVC framework

# GETTING STARTED

# CREATE A FREE DEVELOPER ACCOUNT

https://my.appcelerator.com/auth/signup

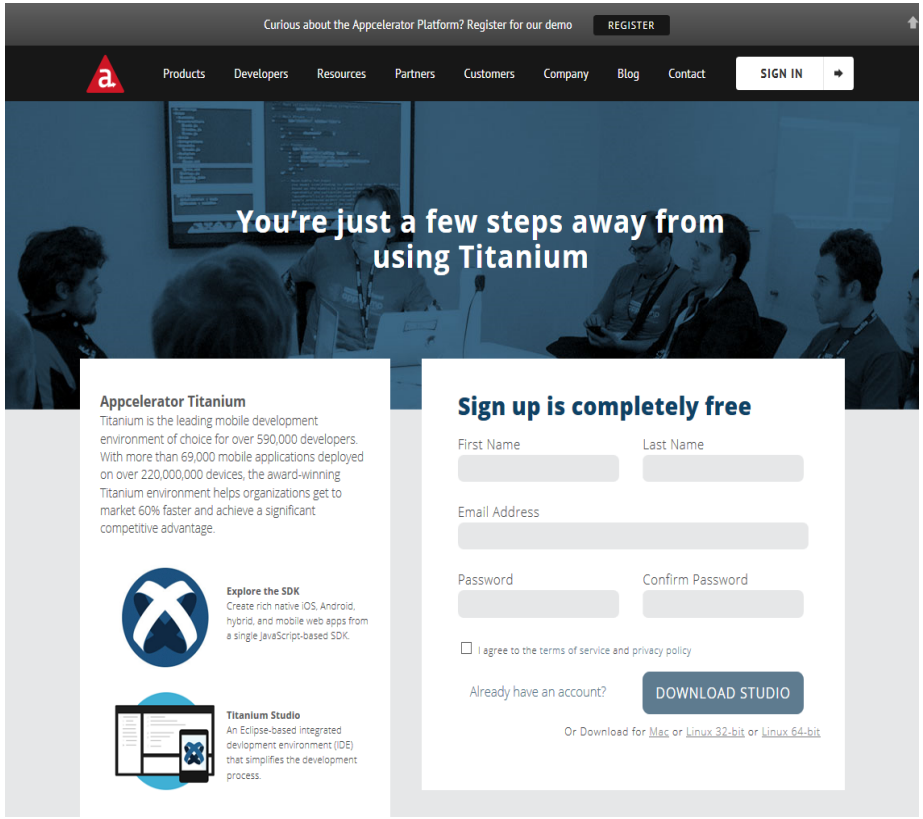
The Developer account is free. You can build, test and deploy to the app stores with the free account.

There is an Enterprise plan that provides a SLA, analytics, more cloud, support, training and more.

# TITANIUM STUDIO

## Download Titanium Studio for free here:

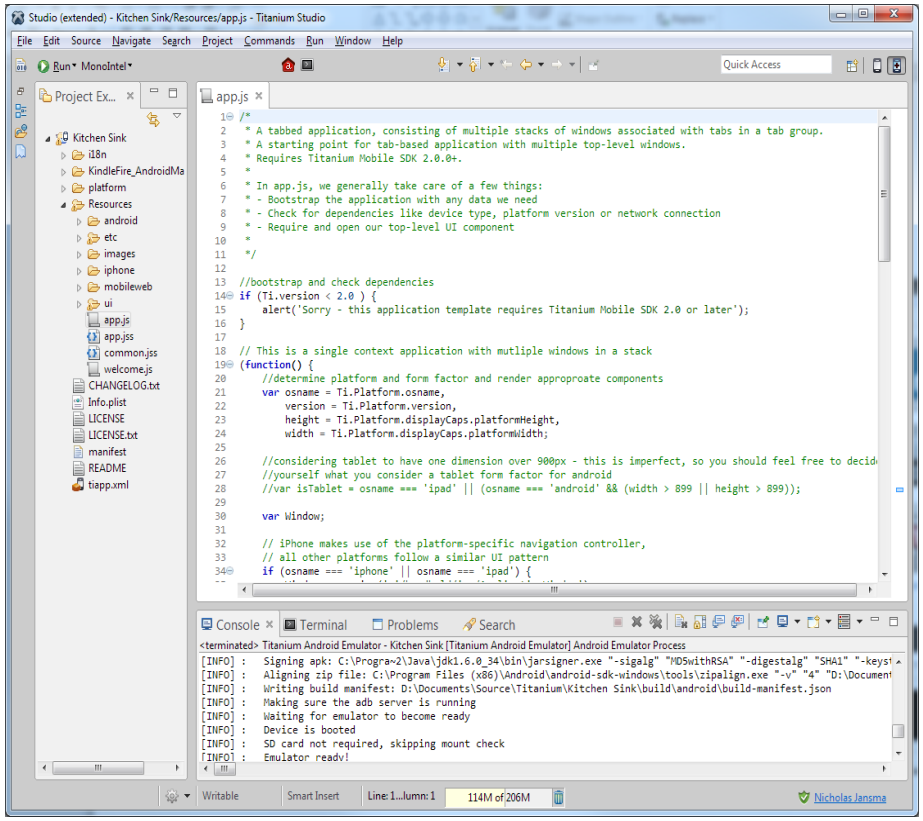
## http://www.appcelerator.com/titanium/download-titanium/




## Works on Mac, Windows and Linux

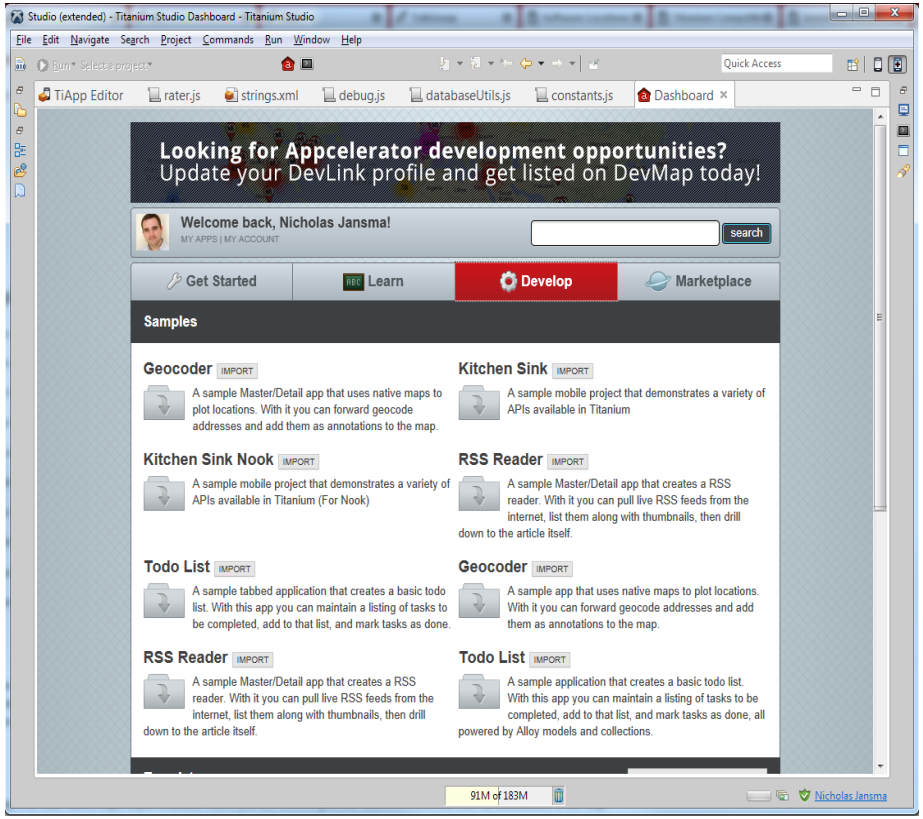
# TITANIUM STUDIO

## (based on Eclipse)



# CODE SAMPLES

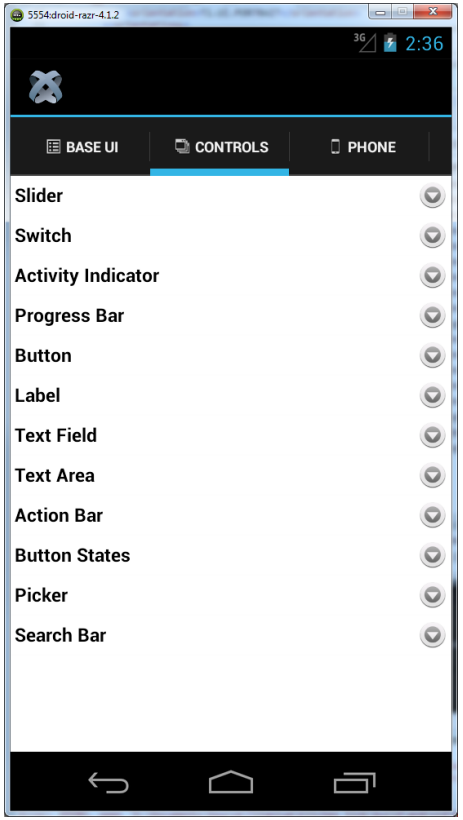
Get the "Kitchen Sink" sample from *Dashboard* | *Develop*, which has code demos for how to use all of the controls:




Also on github.com/appcelerator/KitchenSink

# KITCHEN SINK

Code samples for how to access all of the platform's native controls through the Titanium APIs (`Ti.*`)

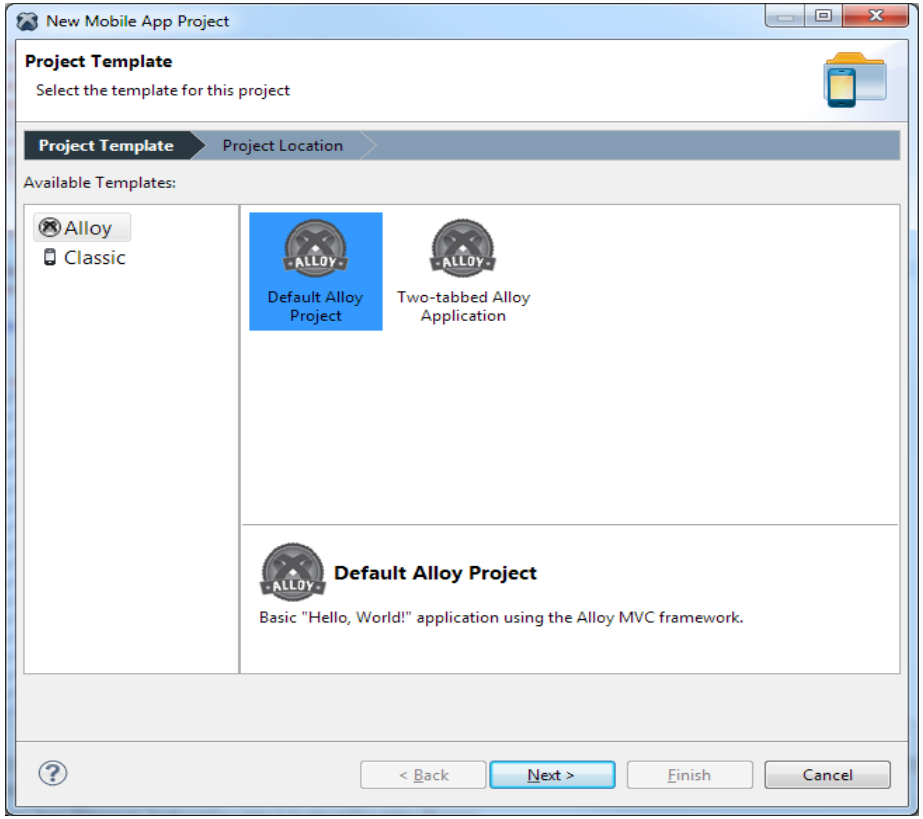


# CREATING A NEW PROJECT

*File | New | Mobile App Project*

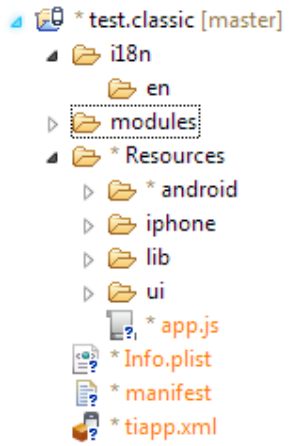
**Alloy** will use their MVC framework

**Classic** lets you build your own UI



# PROJECT STRUCTURE (CLASSIC PROJECT)

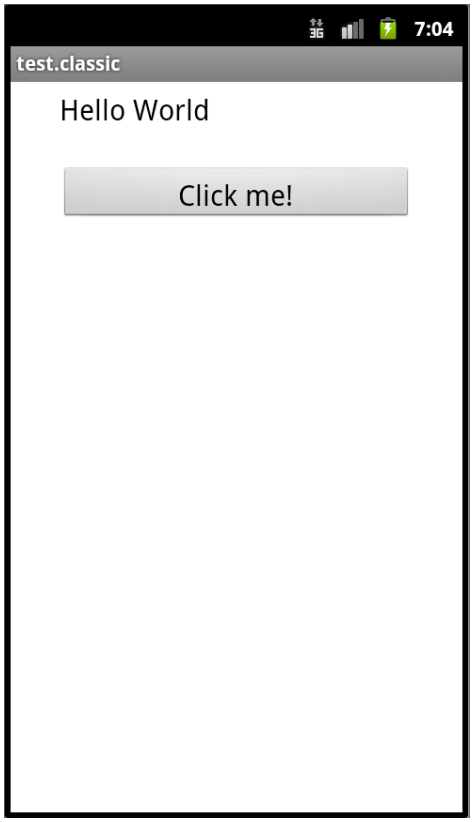
- `i18n/`: Internationalization
- `modules/`: Native modules
- `platform/`: iOS/Android specific
- `Resources`: Core project code
  - `app.js`: Startup file
  - `android/` and `iphone/`: Platform images
  - `lib/`, `ui/`, `whatever/`: Your code



# APP STARTUP

```
var win = Ti.UI.createWindow({
    title: 'Hello, World!',
    layout: 'vertical',
    backgroundColor: 'white'
});

var helloLabel = Ti.UI.createLabel({
    text: 'Hello World',
    color: 'black',
    font: { fontSize: '20sp' },
    height: '40dp',
    width: '250dp'
});
win.add(helloLabel);

var helloButton = Ti.UI.createButton({
    title: 'Click me!',
    font: { fontSize: '20sp' },
    top: '20dp',
    height: '40dp',
    width: '250dp'
});

helloButton.addEventListener('click', function() {
    alert('you clicked me!');
});
win.add(helloButton);

win.open();
```

# DEMO



# DEMO


7:05
test.classic
Hello World
Click me!
Alert
you clicked me!
OK

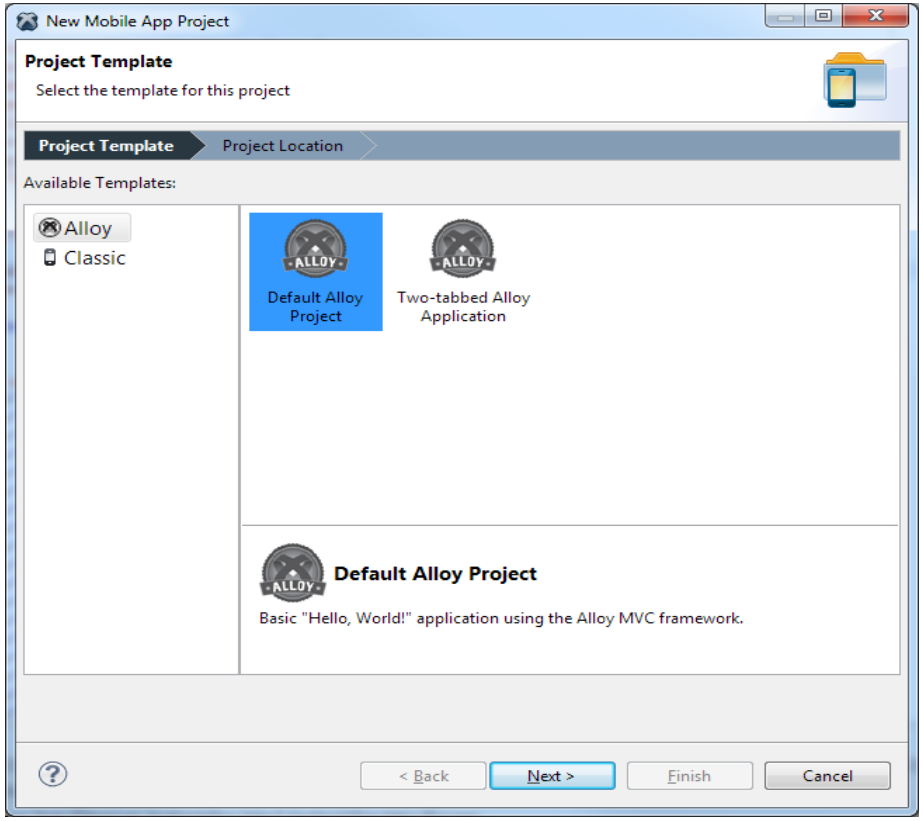
# ALLOY

- Alloy is an open-source **model-video-controller** (MVC) framework for Titanium
- Alloy provides a simple model for **separating** your user interface, business logic and data models
- Alloy uses **XML and CSS** to create and style your views
- Alloy is fully integrated into **Titanium Studio**

# GETTING STARTED

## First, install Alloy via Node.js's NPM:

```
sudo npm install -g alloy
```



# CODE

## app/views/index.html:

```
<Alloy>
    <Window class="container">
        <Label id="label">Hello World</Label>
        <Button id="button" onClick="doClick">Click me!</Button>
    </Window>
</Alloy>
```

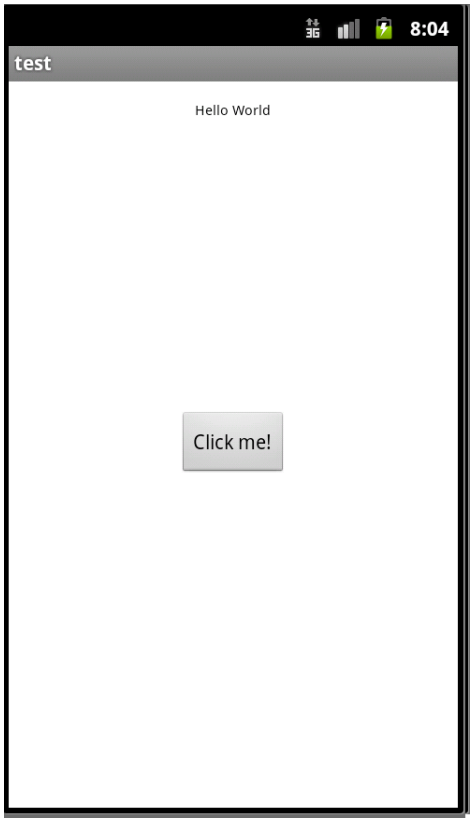
## app/controllers/index.js:

```
function doClick(e) {
    alert('you clicked me!');
}

$.index.open();
```

## app/styles/index.tss:

```
".container": {
    backgroundColor:"white"
},
"#label": {
    top: 20,
    width: Ti.UI.SIZE,
    height: Ti.UI.SIZE,
    color: "#000"
}
```

# DEMO



# TITANIUM MOBILE APIS

- AJAX / web services
- In-App Purchases
- Geolocation
- Camera
- Calendar, Contact
- Media, Photo Gallery
- Gestures, Accelerometer
- Maps
- Analytics
- Social Sharing (Facebook, Yahoo, etc)
- Extensible with your own native iOS/Android packages

# PROS

- One codebase for two+ platforms
- You'll (theoretically) spend less time than if you write two native apps
- Maintenance on one codebase should be easier in the long run
- Native interface controls: your app looks native, not like web controls
- Might be able to reuse your JavaScript in other parts of your project
- As time progresses, code re-use percentage increases

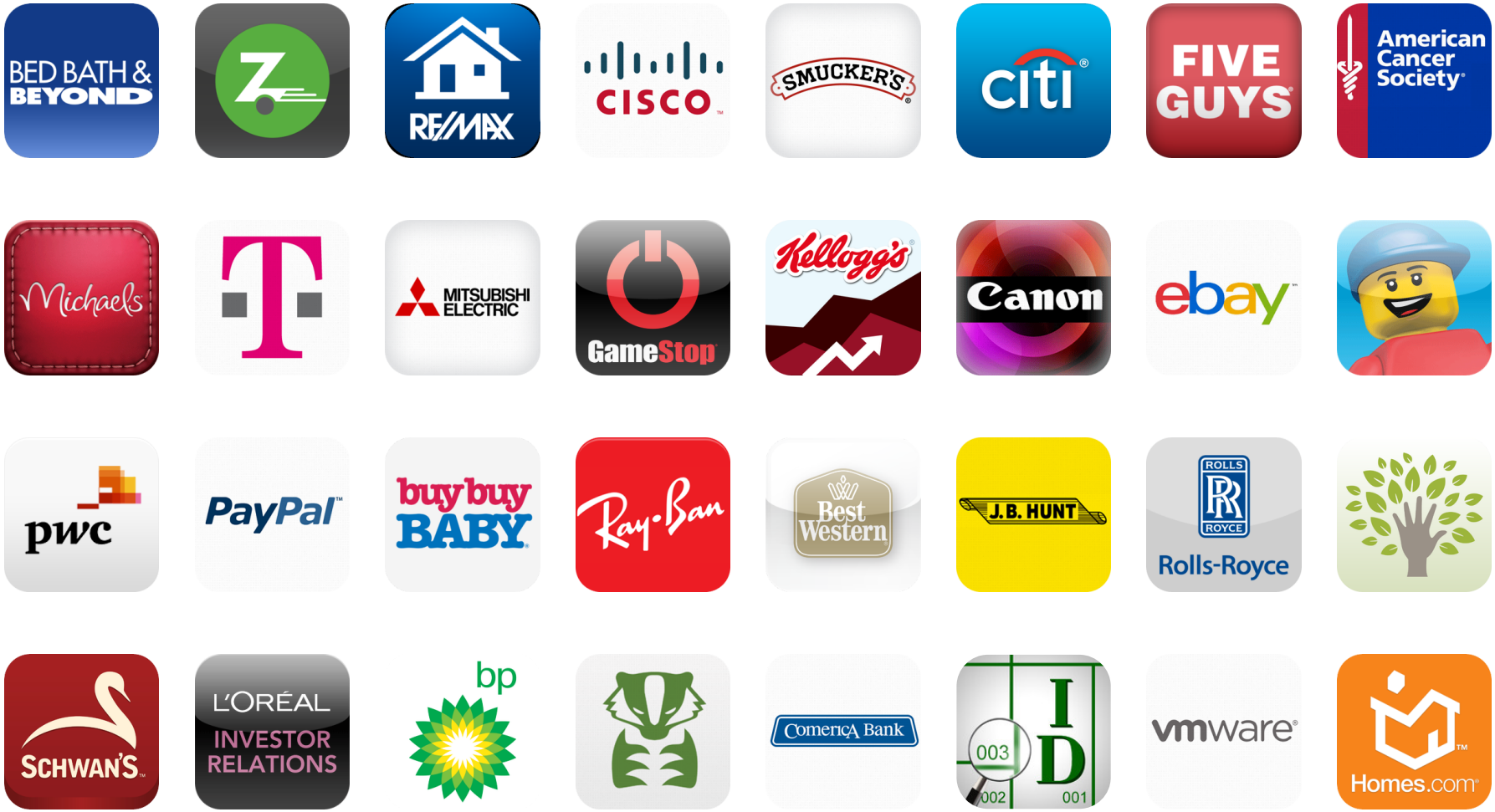
# PROS

- JavaScript is great for rapid prototyping, and works really well with Titanium
- Titanium is open-source: github.com/appcelerator/titanium_mobile
- The platform is mature and stable
- Appcelerator is well funded
- SDK and API documentation, tutorials and samples have improved dramatically over the last year
- 610,000 developers, ISVs and partners
- 238,512,651 Devices are running apps built with Titanium (as of this morning)

# CONS

- Need to learn a new platform / SDK / quirks
- Knowing the ins & outs of native iOS / Android will help
- It's necessary to have `if(iOS){}` and `if(android){}`
- Performance isn't 100% of a native app (but better than running in a web control)

# WHO USES TITANIUM

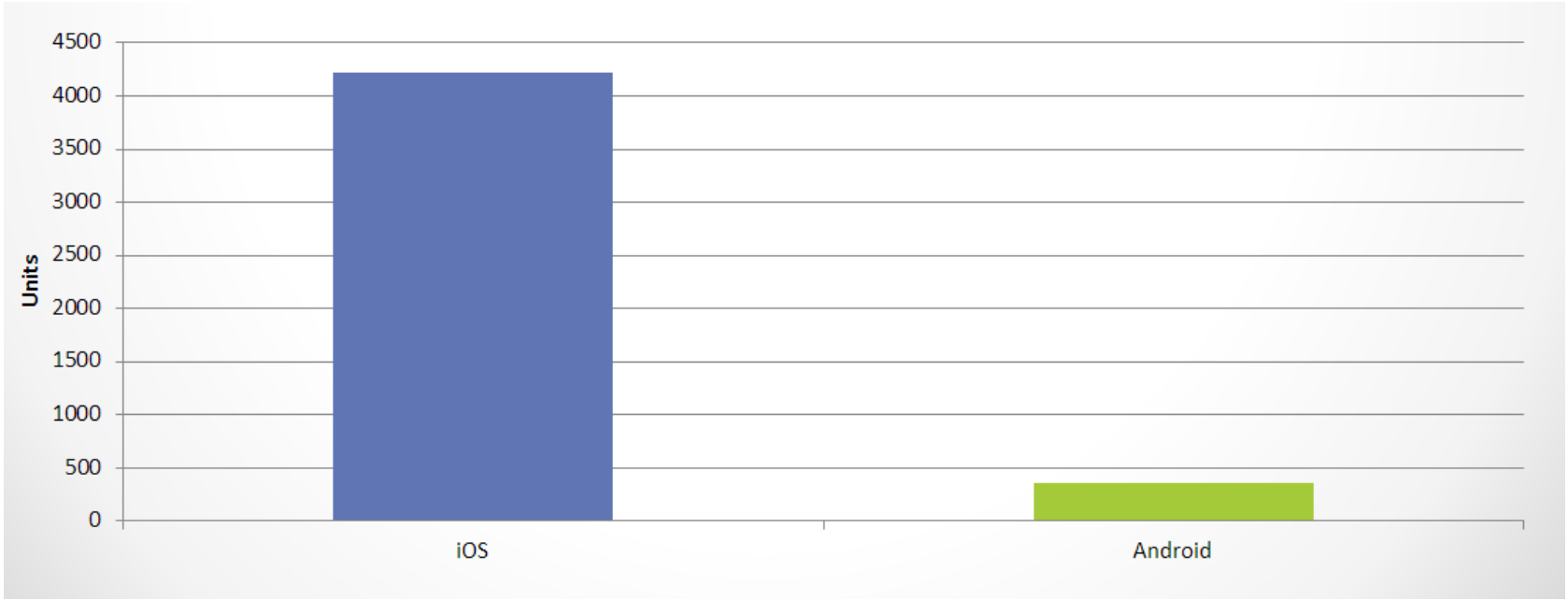


# LESSONS LEARNED

- Your first Titanium app, from development to release will take longer than you expect.
- Once you are over the first hurdle, subsequent apps & updates will be faster
- It takes time to ramp-up on good JavaScript patterns: CommonJS modules, Crockford-isms, etc
- iOS simulator is a lot faster to test on. For Android, use Genymotion!
- For community support, you'll need to use a combination of the Appcelerator API Docs, Q&A site, videos and StackOverflow

# LESSONS LEARNED

- You won't double your sales just by releasing on both platforms




- Setting up your testers to use your app (prior to release) is harder than expected

# LINKS

- Appcelerator: appcelerator.com
- Alloy MVC: appcelerator.com/platform/alloy/
- Titanium Studio: appcelerator.com/titanium/titanium-studio/
- Code samples: docs.appcelerator.com/titanium/3.0/#!/guide/Titanium_Sample
- Videos: vimeo.com/appcelerator

Thanks to Nic Jansma for the foundation of this presentation - nicj.net - @NicJ