iOS for Android Developers

with Swift

David Truxall, Ph.D.

http://bit.ly/androidToIos

About Me

@davetrux

blog.davidtruxall.com

You

- Know Java
- Know Android
- Don’t know Swift
- Don’t know iOS
- Need a Mac

Goal

Learn basic iOS concepts for someone familiar with Android (using Swift)

Why?

- Do I hate Android now?
- Neither platform is “the winner”
- Clients want both platforms
- You can make more money
- Swift is the new hotness in mobile
- Objective-C

Agenda

1. Brief language intro
2. Project/Tool Structure
3. App Architecture
4. Coding Demo

Swift

- Object-oriented AND Functional
- C family
- Cleaner, simpler, safer than Objective-C
- Modern features

Compare

```
- (NSString*) concatenateString:(NSString*)stringA withString:(NSString*)stringB
{
    NSString *finalString = [NSString stringWithFormat:@"%@%@", stringA, stringB];
    return finalString;
}
```

```
func concatenateString(stringA: String, stringB: String) ->String {
    let result = stringA + stringB
    return result
}
```

Swift Features

- Closures
- Tuples and multiple return values
- Generics
- Structs that support methods
- Functional programming patterns

Language

Java/Android	Swift/iOS
`import com.package.name;`	`import frameworkname`
`int counter;`	`var counter :Int`
`static final int LEVELS = 8;`	`let levels = 8`
`private`	`private`
`public`	`public`
-	`internal (*)`
`protected`	-

Objects

Java/Android	Swift/iOS
`class Foo extends Bar {}`	`class Foo : Bar`
`interface Baz{}`	`protocol Baz`
`class Foo implements Baz{}`	`class Bar : Baz {}`
`Foo();`	`init()`
`void doWork(String arg){}`	`func doWork(arg: String) -> Void`
`Foo item = new Foo();`	`var item : Foo = Foo()`
`item.doWork(arg);`	`item.doWork(arg)`

Optionals

? - Has a value or no value at all (nil)

! - Implicitly Unwrapped Optional

Swift OO

```
class VideoMode {
    var resolution : Resolution = Resolution()
    var interlaced = false
    let frameRate = 60.0
    var name: String?

    func setUpMode(modeName: String) -> Void {<do stuff>}
}
```

Swift Functional

```
func addTwoInts(a: Int, b: Int) -> Int {
    return a + b
}

var addFunction: (Int, Int) -> Int = addTwoInts

func printMath(mathFunction: (Int, Int) -> Int, a: Int,
b: Int) {
    println("Result: \(mathFunction(a, b))")
}
printMath(addTwoInts, 3, 5)
```

Xcode

- Free
- It's an IDE
- Click not double-click
- Virtual file organization

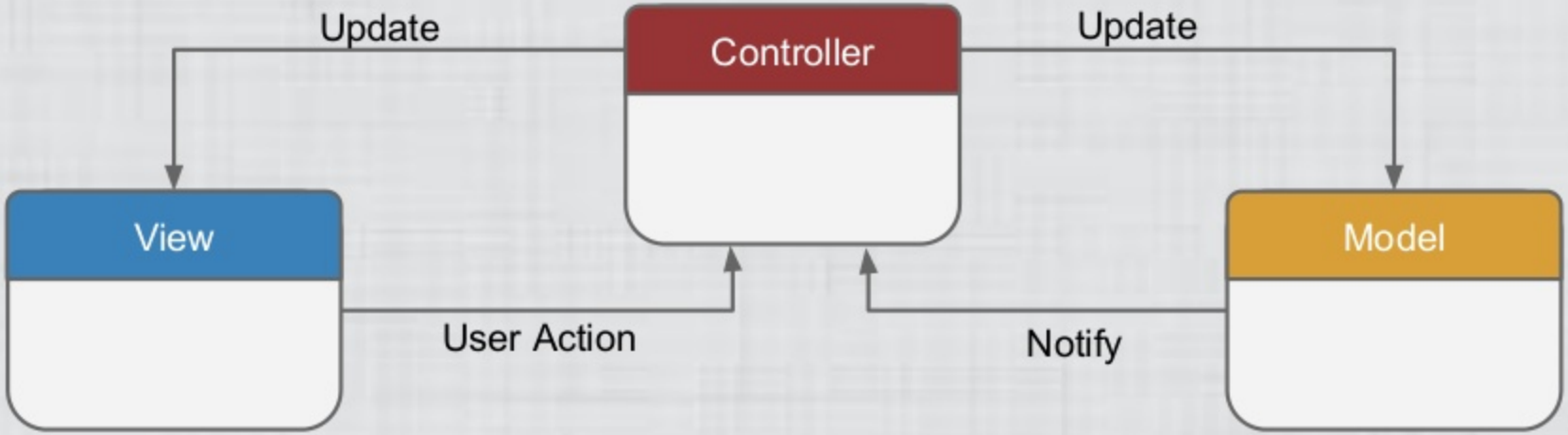
Demo

Application Architecture

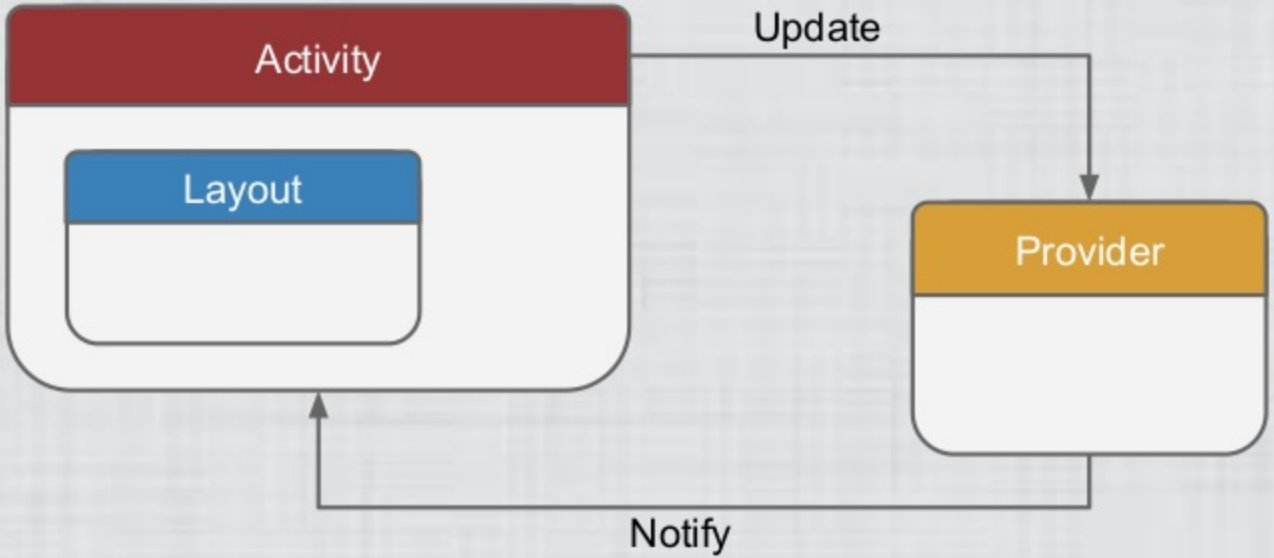
Model - View - Controller

UI Organization and Plumbing

Model - View -Controller



Android != Model-View-Controller



iOS Model - View -Controller