nimble

Chopping iOS App Delegate

Jason

Growth Session #23 - March 21-22 2019

Intro

Chopping iOS App Delegate into bite-sized pieces

Goals

Break-down **The 1** iOS app delegate

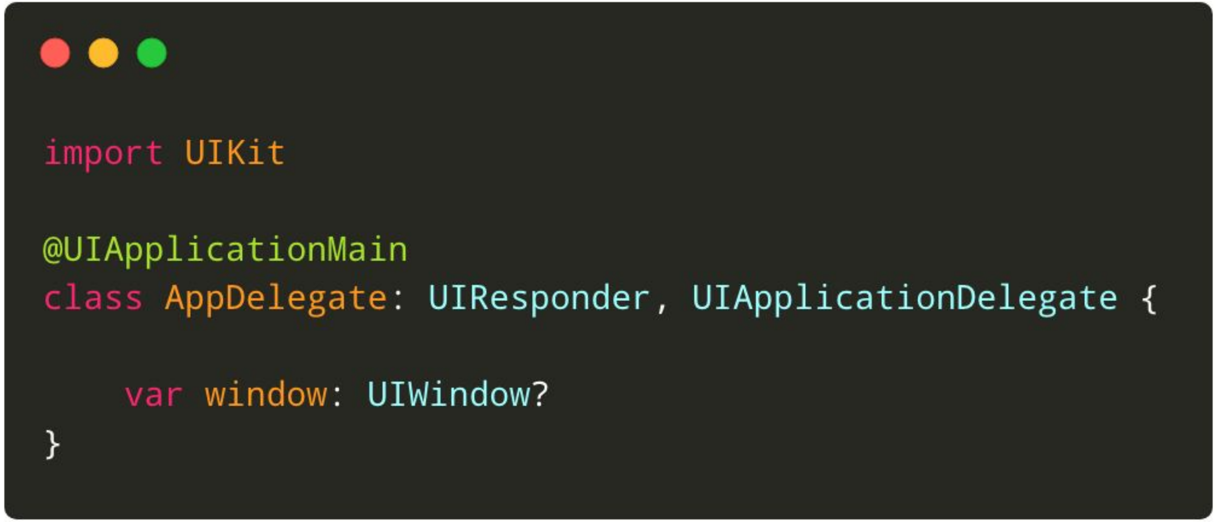
What is App Delegate

App delegate is a class which is called by the system on:

- Each app lifecycle change:
  - willFinishLaunchingWithOptions
  - didFinishLaunchingWithOptions
  - applicationDidBecomeActive
  - applicationWillResignActive
  - applicationDidReceiveMemoryWarning
- Opening universal link
- Receives push notifications
- Handling interaction with Apple Watch
- Opening files

What Is Usually Happening...

When the project is just created...



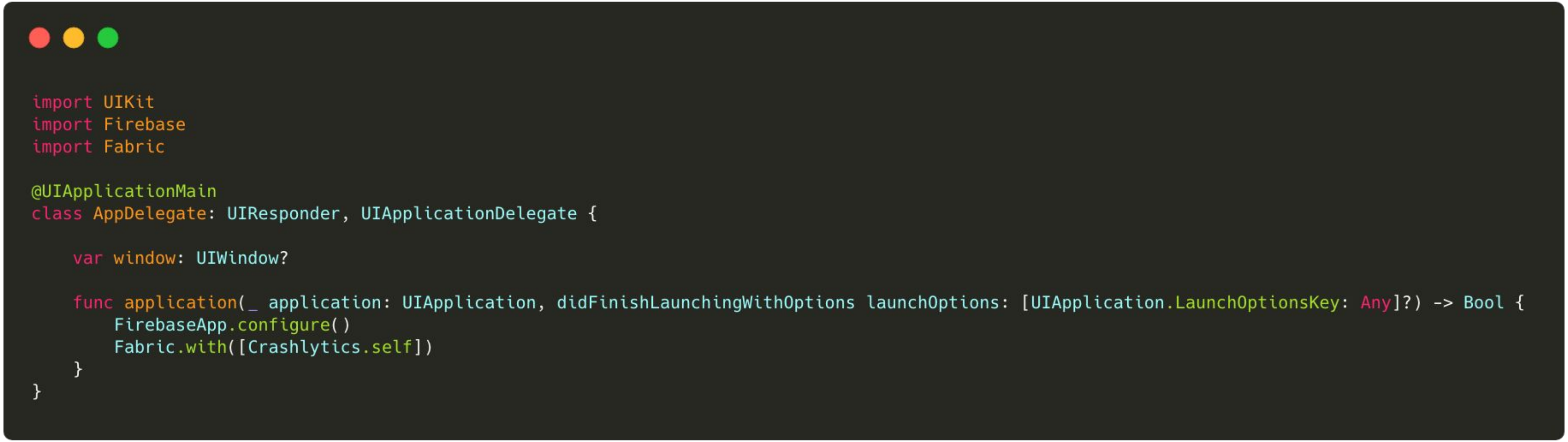
```swift
import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?
}
```

What Is Usually Happening…

Marketing team comes and asks…

What Is Usually Happening...

We add two lines of code



```swift
import UIKit
import Firebase
import Fabric

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        FirebaseApp.configure()
        Fabric.with([Crashlytics.self])
    }
}
```

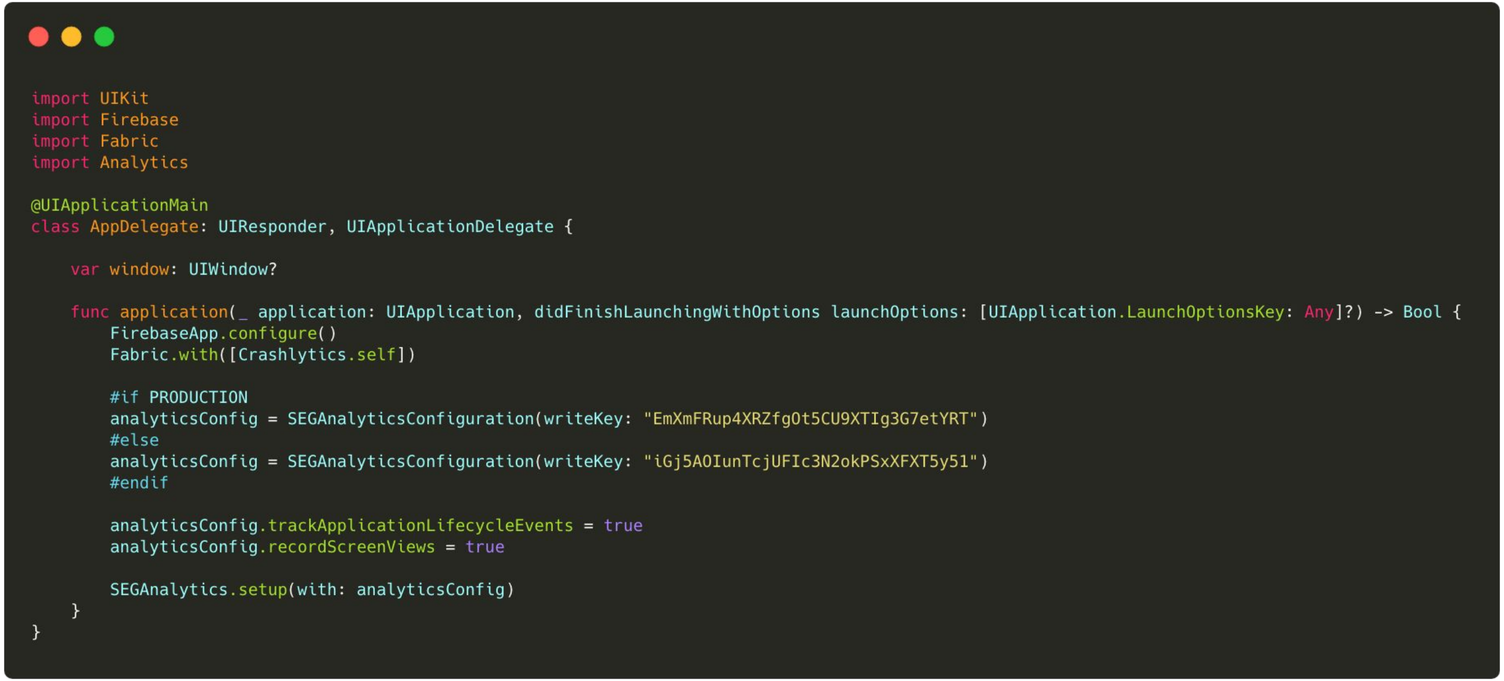
What Is Usually Happening...

Marketing team comes again and ask...

Segment

What Is Usually Happening...

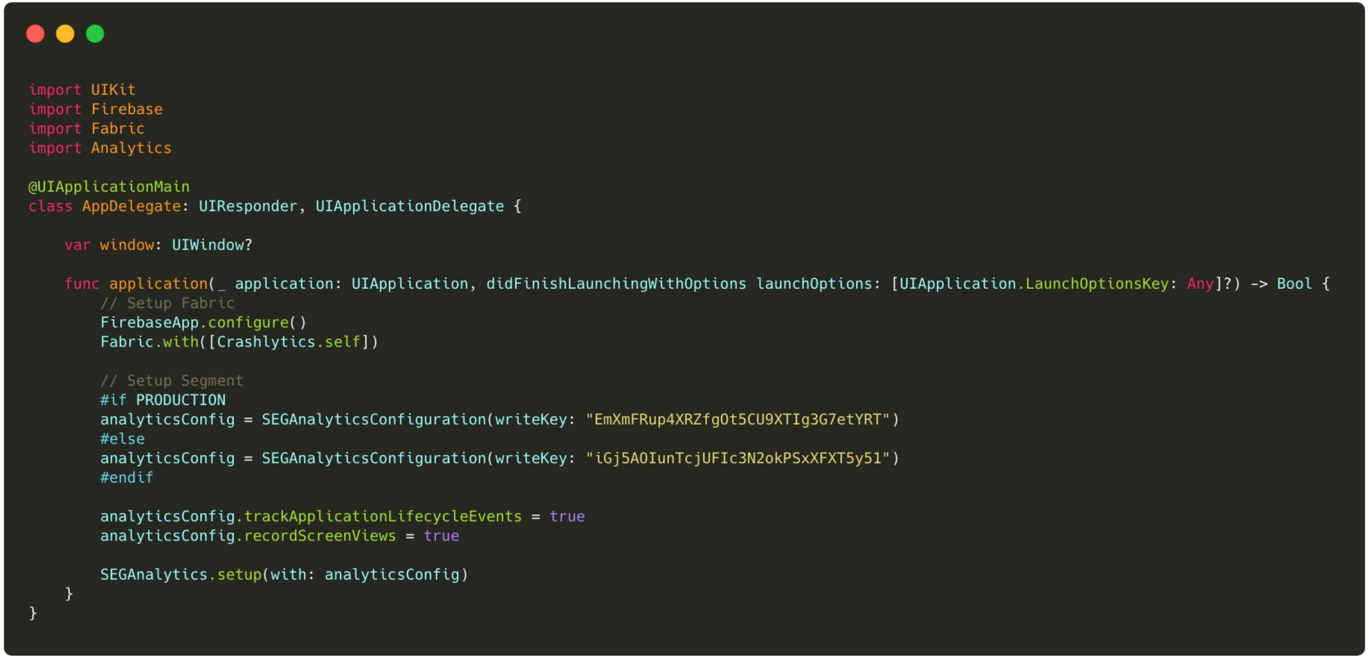
Add multiple lines of code



```
import UIKit
import Firebase
import Fabric
import Analytics

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        FirebaseApp.configure()
        Fabric.with([Crashlytics.self])

        #if PRODUCTION
        analyticsConfig = SEGAnalyticsConfiguration(writeKey: "EmXmFRup4XRZfgOt5CU9XTIg3G7etYRT")
        #else
        analyticsConfig = SEGAnalyticsConfiguration(writeKey: "iGj5AOIunTcjUFIc3N2okPSxXFXT5y51")
        #endif

        analyticsConfig.trackApplicationLifecycleEvents = true
        analyticsConfig.recordScreenViews = true

        SEGAnalytics.setup(with: analyticsConfig)
    }
}
```

What Is Usually Happening...

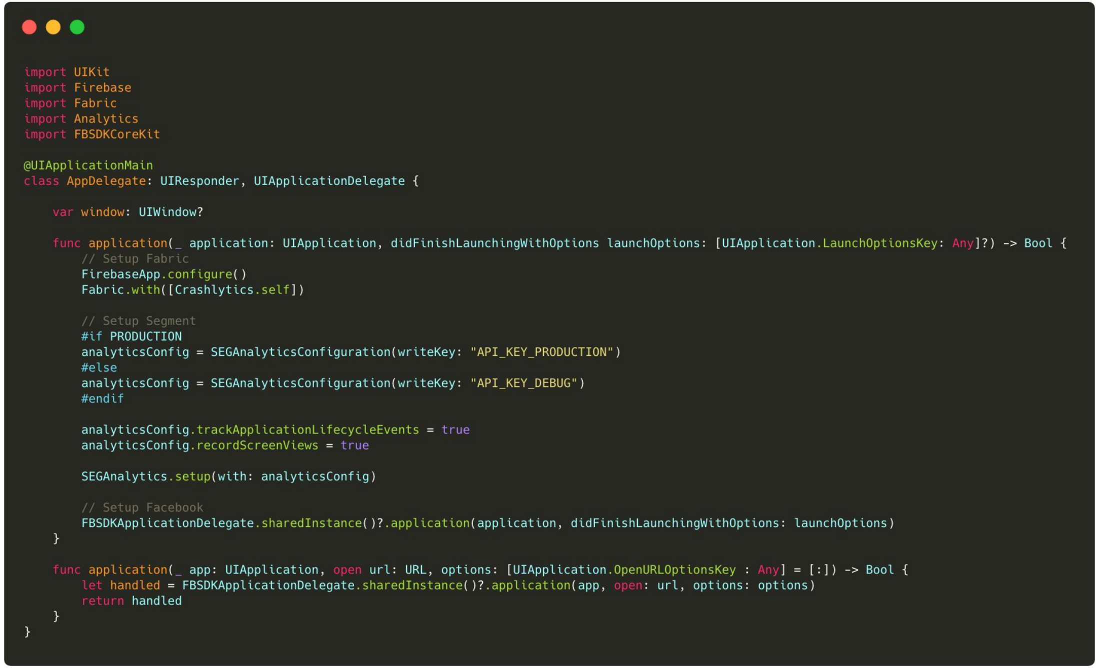
Add multiple lines of code again



```swift
import UIKit
import Firebase
import Fabric
import Analytics

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Setup Fabric
        FirebaseApp.configure()
        Fabric.with([Crashlytics.self])

        // Setup Segment
        #if PRODUCTION
        analyticsConfig = SEGAnalyticsConfiguration(writeKey: "EmXmFRup4XRZfg0t5CU9XTIg3G7etYRT")
        #else
        analyticsConfig = SEGAnalyticsConfiguration(writeKey: "iGj5AOIunTcjUFIc3N2okPSxXFXT5y51")
        #endif

        analyticsConfig.trackApplicationLifecycleEvents = true
        analyticsConfig.recordScreenViews = true

        SEGAnalytics.setup(with: analyticsConfig)
    }
}
```

What Usually Happening...

We need Facebook sign in



```
import UIKit
import Firebase
import Fabric
import Analytics
import FBSDKCoreKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Setup Fabric
        FirebaseApp.configure()
        Fabric.with([Crashlytics.self])

        // Setup Segment
        #if PRODUCTION
        analyticsConfig = SEGAnalyticsConfiguration(writeKey: "API_KEY_PRODUCTION")
        #else
        analyticsConfig = SEGAnalyticsConfiguration(writeKey: "API_KEY_DEBUG")
        #endif

        analyticsConfig.trackApplicationLifecycleEvents = true
        analyticsConfig.recordScreenViews = true

        SEGAnalytics.setup(with: analyticsConfig)

        // Setup Facebook
        FBSDKApplicationDelegate.sharedInstance()?.application(application, didFinishLaunchingWithOptions: launchOptions)
    }

    func application(_ app: UIApplication, open url: URL, options: [UIApplication.OpenURLOptionsKey : Any] = [:]) -> Bool {
        let handled = FBSDKApplicationDelegate.sharedInstance()?.application(app, open: url, options: options)
        return handled
    }
}
```

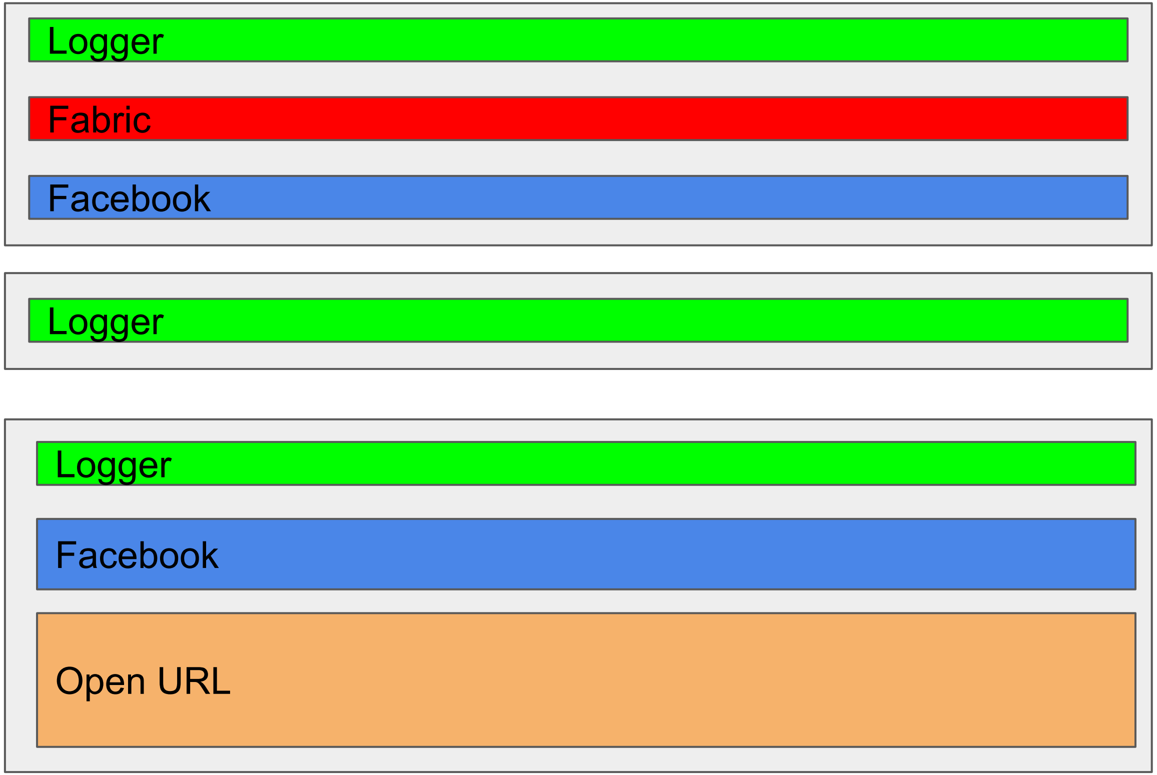
What Is Usually Happening...

Adding universal link support

What Is Usually Happening...

Handling Push Notifications

What Is Usually Happening...



Logger

Fabric

Facebook

Logger

Logger

Facebook

Open URL

What Is Usually Happening...

The 1 iOS project’s App Delegate has

What Is Usually Happening...

438 lines of code

What Is Usually Happening...

Red Planet iOS project’s App Delegate has

What Is Usually Happening...

509 lines of code and...

What Is Usually Happening…

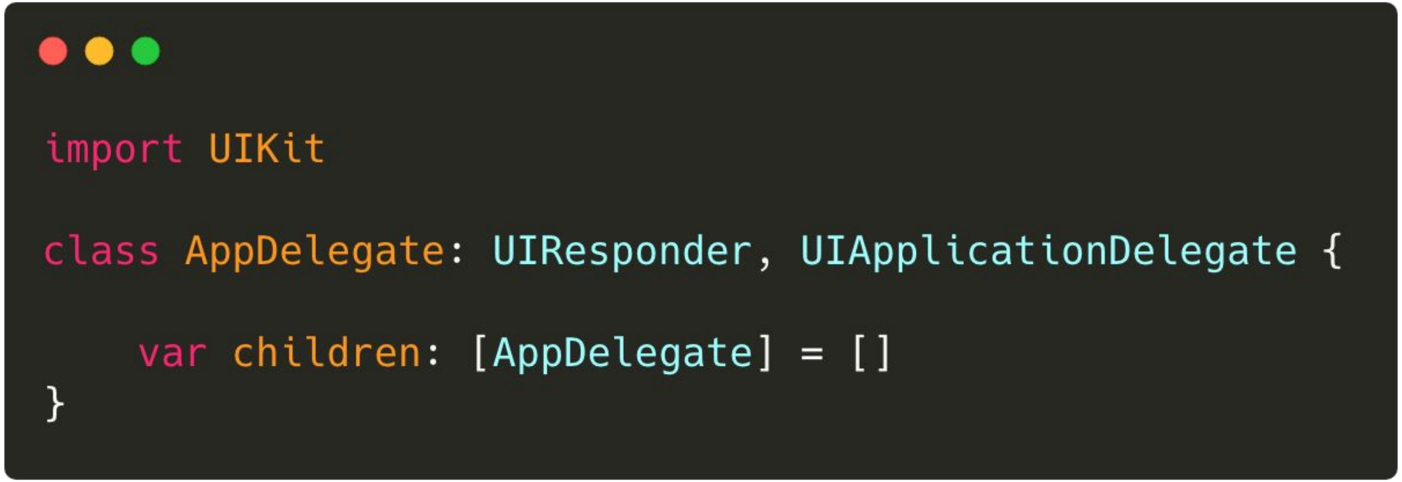
These classes are very fragile and sensitive…

What Is Usually Happening...

they are too big to fight...

Child App Delegate

App delegate with child app delegates



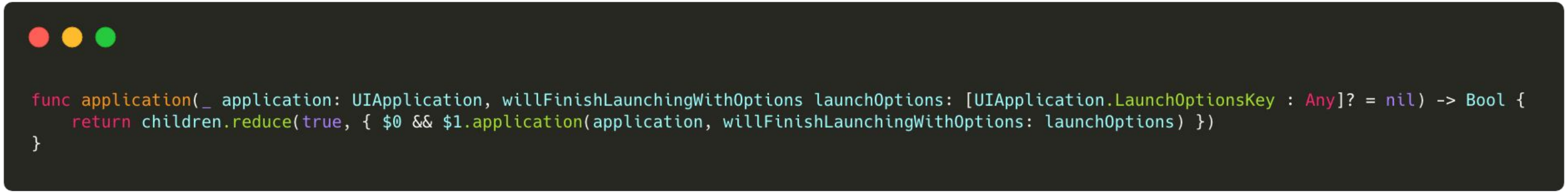
```
import UIKit

class AppDelegate: UIResponder, UIApplicationDelegate {

    var children: [AppDelegate] = []
}
```

Iterate Child App Delegate

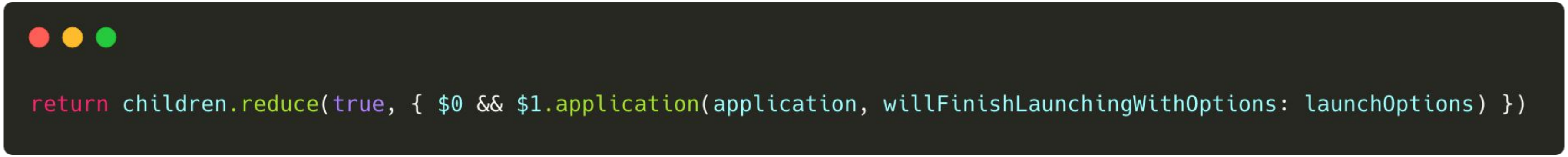
There are **3** types of App Delegate methods

Iterate Child App Delegate

Returning Value



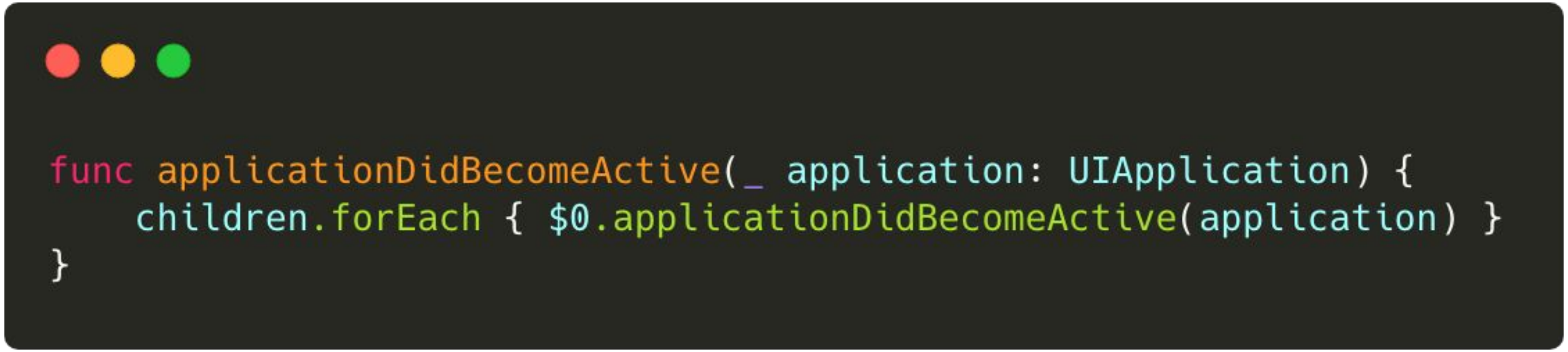
```
func application(_ application: UIApplication, willFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey : Any]? = nil) -> Bool {
    return children.reduce(true, { $0 && $1.application(application, willFinishLaunchingWithOptions: launchOptions) })
}
```



```
return children.reduce(true, { $0 && $1.application(application, willFinishLaunchingWithOptions: launchOptions) })
```

Iterate Child App Delegate

Invocation



```swift
func applicationDidBecomeActive(_ application: UIApplication) {
    children.forEach { $0.applicationDidBecomeActive(application) }
}
```

Iterate Child App Delegate

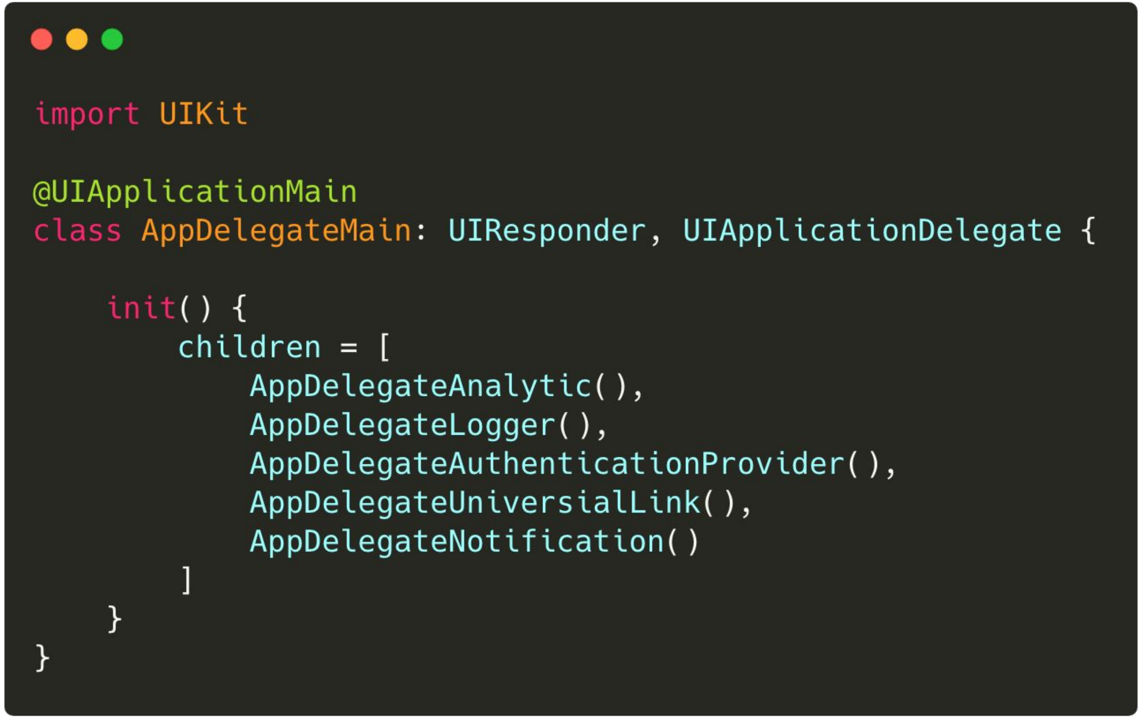
Completion block

Structure

AppDelegateMain

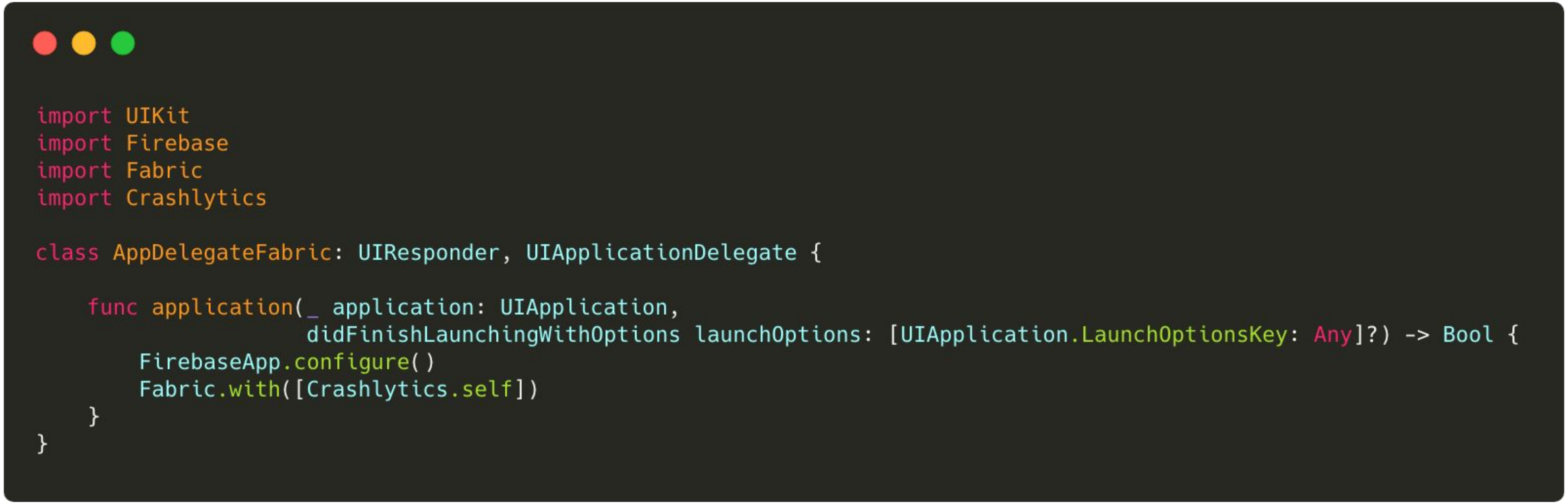
- AppDelegateAnalytic
  - AppDelegateFabric
  - AppDelegateSegment
  - AppDelegateUrbanAirship
- AppDelegateLogger
- AppDelegateAuthenticationProvider
  - AppDelegateFacebook
  - AppDelegateGitHub
- AppDelegateUniversialLink
- AppDelegateNotification

Example



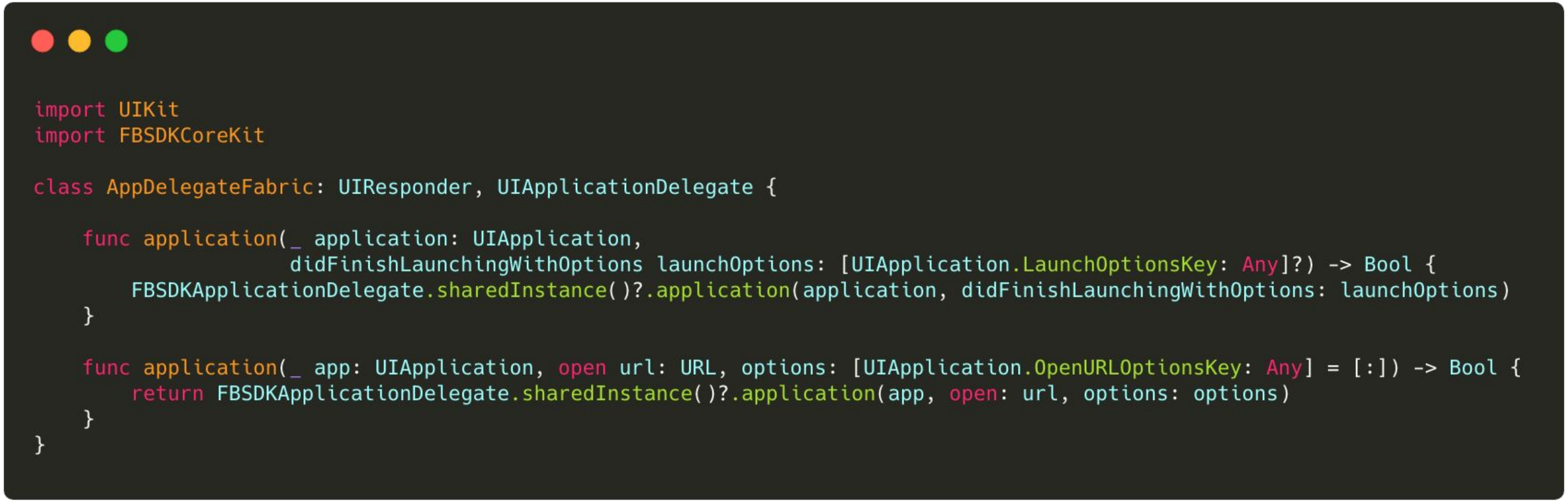
```swift
import UIKit

@UIApplicationMain
class AppDelegateMain: UIResponder, UIApplicationDelegate {

    init() {
        children = [
            AppDelegateAnalytic(),
            AppDelegateLogger(),
            AppDelegateAuthenticationProvider(),
            AppDelegateUniversialLink(),
            AppDelegateNotification()
        ]
    }
}
```

Example



```
import UIKit
import Firebase
import Fabric
import Crashlytics

class AppDelegateFabric: UIResponder, UIApplicationDelegate {

    func application(_ application: UIApplication,
                     didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        FirebaseApp.configure()
        Fabric.with([Crashlytics.self])
    }
}
```

Example



```
import UIKit
import FBSDKCoreKit

class AppDelegateFabric: UIResponder, UIApplicationDelegate {

    func application(_ application: UIApplication,
                     didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        FBSDKApplicationDelegate.sharedInstance()?.application(application, didFinishLaunchingWithOptions: launchOptions)
    }

    func application(_ app: UIApplication, open url: URL, options: [UIApplication.OpenURLOptionsKey: Any] = [:]) -> Bool {
        return FBSDKApplicationDelegate.sharedInstance()?.application(app, open: url, options: options)
    }
}
```

Conclusion

- Easier debugging and configuration
- Enforce better practice
- Better testability
- Reusability
- Abstraction

Next Steps

- Publish to CocoaPods
- Implementing App Delegates?

Adequate abstractions and modulizations increase productivity dramatically

Thanks!

Contact Nimble

nimblehq.co
hello@nimblehq.co

Bangkok

399 Interchange 21 Sukhumvit Road, Unit
#2402-03, Klong Toei, Wattana, Bangkok
10110, Thailand

Singapore

28C Stanley St, Singapore 068737

Hong Kong

20th Floor, Central Tower
28 Queen's Road, Central, Hong Kong

nimble