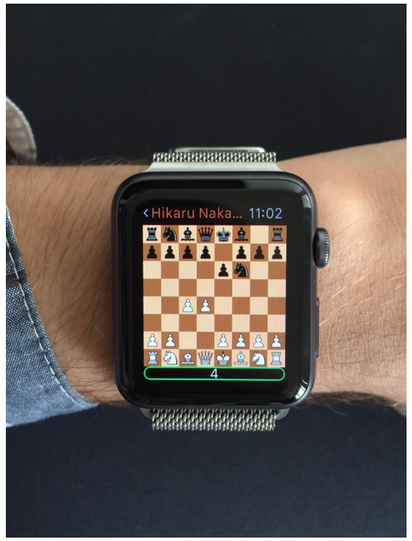
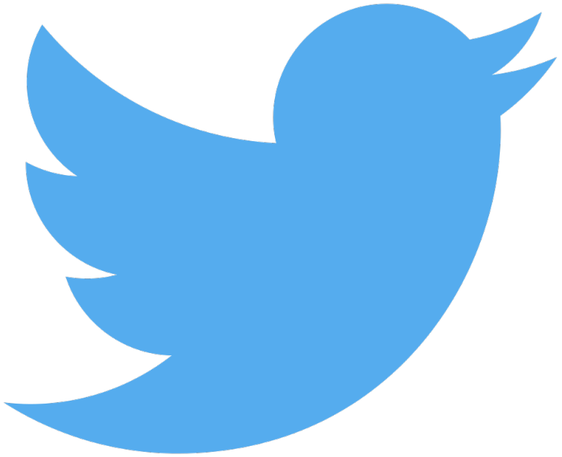
Building Fabric.app in Swift

Intro

pebble





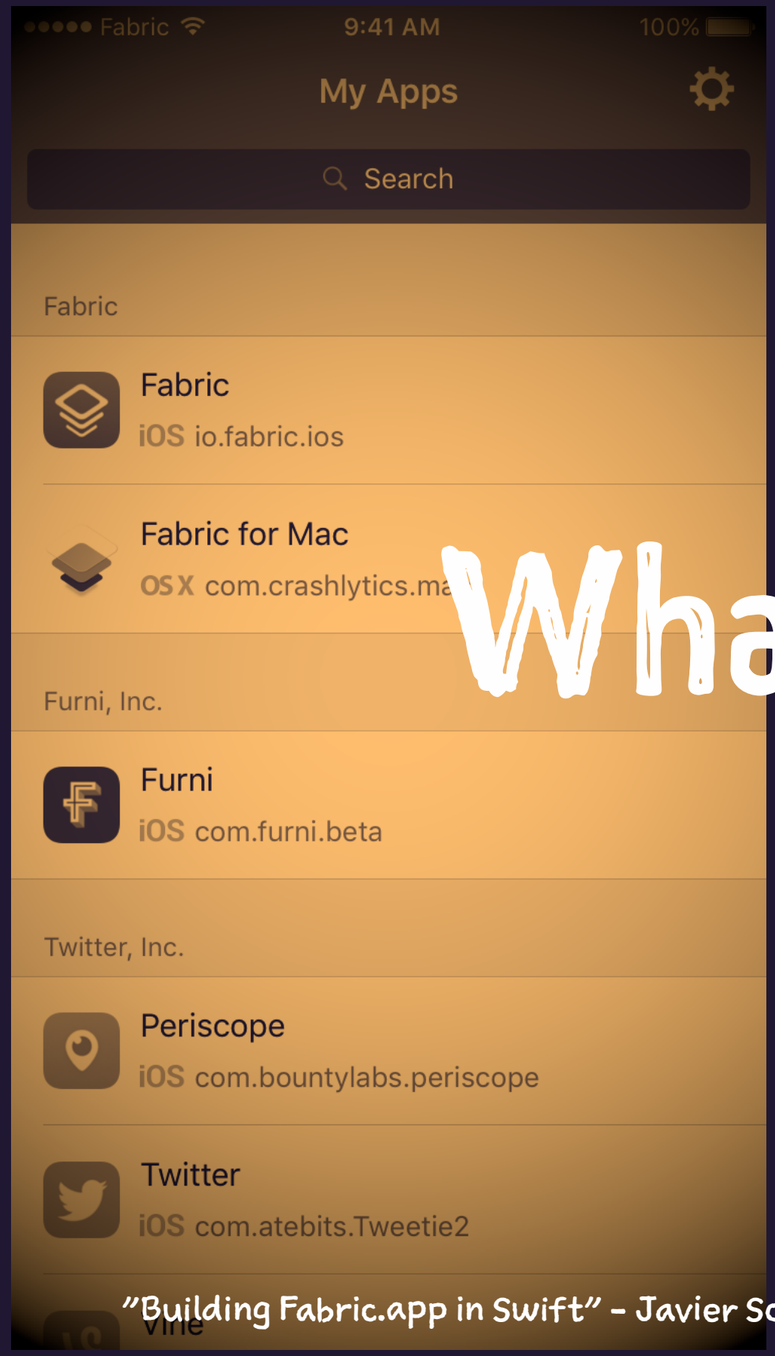
@Javi

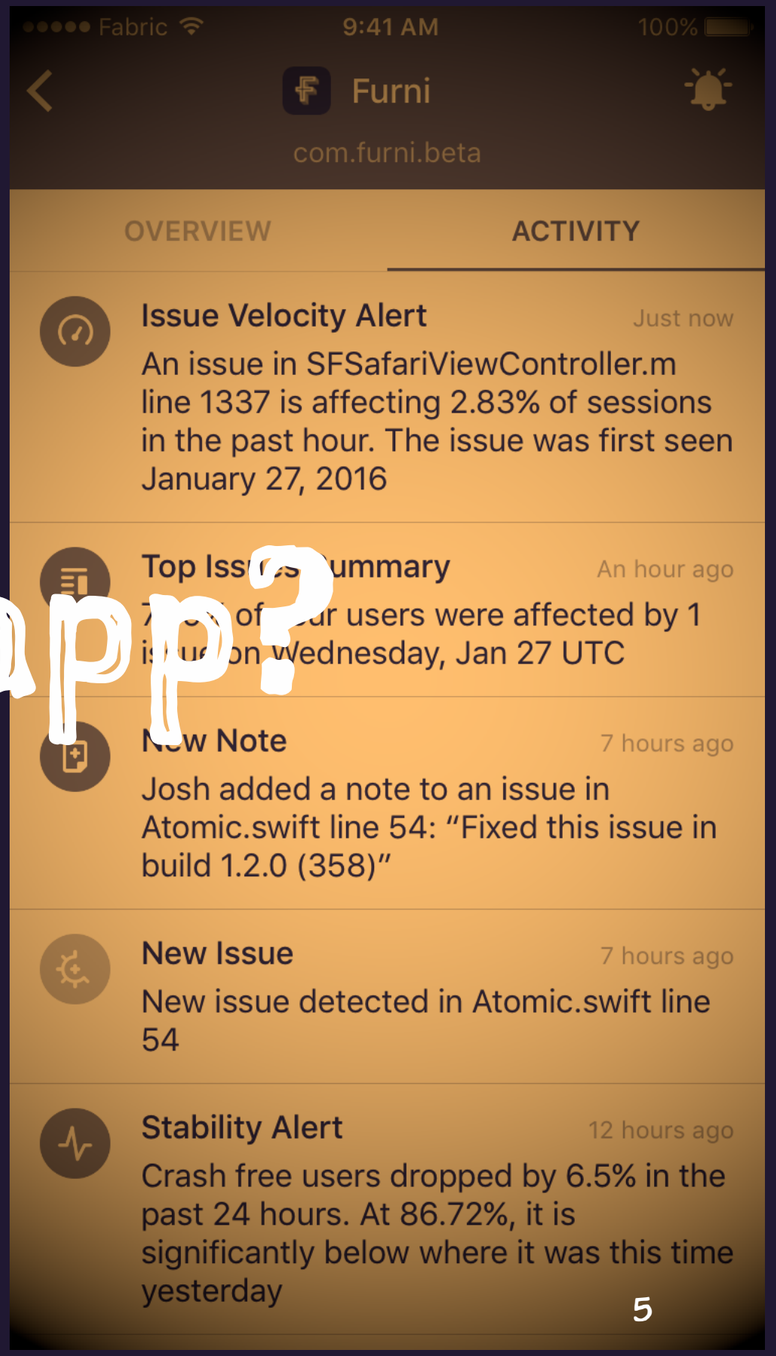

WHEN YOU LOVE SWIFT
AND HAVEN'T TOLD ANYBODY IN 10 MINUTES
memegenerator.net

Outline

- What is Fabric.app?
- Tools used
- Architecture
- Better with Swift
- Error reporting

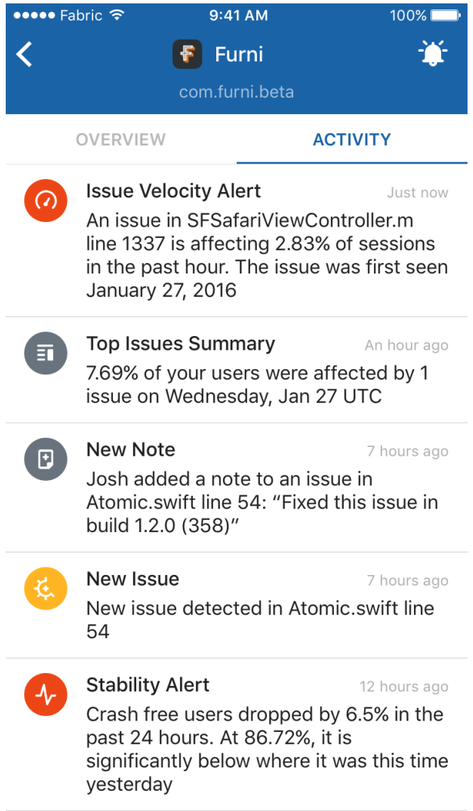
What is Fabric.app?





What is Fabric.app?

What is Fabric.app?



Tools

Dependency management

Dependency management

- SSL Pinning
- Keychain
- IPassword integration
- Graphs
- Networking, caching

Dependency management

- CocoaPods
- Carthage

CocoaPods

- More widely adopted
- Less work to integrate frameworks
- Easily link some frameworks only in Debug

Carthage

→ Precompilation == Speed

Swift Package Manager

Interface Builder

Interface Builder

- AutoLayout
- Storyboards?
- robb/Cartography

AutoLayout

AutoLayout

```
Oct 17 11:01:52 Jeffs-iPhone SpringBoard[11069] <Warning>: Unable to simultaneously satisfy constraints.
        Probably at least one of the constraints in the following list is one you don't want. Try this: (1) look at each constraint and try to figure out
which you don't expect; (2) find the code that added the unwanted constraint or constraints and fix it. (Note: If you're seeing
NSAutoresizingMaskLayoutConstraints that you don't understand, refer to the documentation for the UIView property translatesAutoresizingMaskIntoConstraints)
    (
        "<NSLayoutConstraint:0x131668130 UILabel:0x1316852a0'SIRI SUGGESTIONS'.leading == SPUISearchTableHeaderView:0x1316b3150.leadingMargin>",
        "<NSLayoutConstraint:0x1316c39b0 H:[UILabel:0x1316852a0'SIRI SUGGESTIONS']-(NSSpace(8))-[UIButton:0x1316c0b10'Show More']>",
        "<NSLayoutConstraint:0x1316c3a00 SPUISearchTableHeaderView:0x1316b3150.trailingMargin == UIButton:0x1316c0b10'Show More'.trailing>",
        "<NSLayoutConstraint:0x131403e30 UIButtonLabel:0x131687e10'Show More'.width <= UIButton:0x1316c0b10'Show More'.width>",
        "<NSLayoutConstraint:0x1316845e0 'UIView-Encapsulated-Layout-Width' H:[SPUISearchTableHeaderView:0x1316b3150(0)]>"
    )

    Will attempt to recover by breaking constraint
    <NSLayoutConstraint:0x131403e30 UIButtonLabel:0x131687e10'Show More'.width <= UIButton:0x1316c0b10'Show More'.width>
```

Storyboards

UIStoryboard

```
class UIStoryboard {
    func instantiateViewControllerWithIdentifier(_ identifier: String) -> UIViewController
}
```

UIStoryboard

```
let vc = mainStoryboard.instantiateViewControllerWithIdentifier("ApplicationOverviewID") as! ApplicationOverviewVC
vc.userSession = ...
vc.applicationID = ...
```

UIStoryboard

```
final class ApplicationOverviewVC {
    // I hope you like "!"s...
    var userSession: UserSession!
    var applicationID: String!

    func viewDidLoad() {
        super.viewDidLoad()

        self.userSession.foo()
    }
}
```

~~Storyboards~~

```
final class ApplicationOverviewVC {
    let userSession: UserSession
    let applicationID: String

    init(userSession: UserSession, applicationID: String) {
            self.userSession = userSession
            self.applicationID = applicationID

            super.init(nibName: ..., bundle: ...)
    }

    func viewDidLoad() {
        super.viewDidLoad()

        // Compile-time guarantee of object fully initialized
        self.userSession.foo()
    }
}
```

Fastlane

Fastlane

Fabric app

👕 Standard Setup

- Deploy updates through Crashlytics Beta.
- Submit to the AppStore with screenshots and metadata.
- Code generation for xib names, identifiers, etc using `R.swift`

- https://github.com/fastlane/examples

ReactiveCocoa

ReactiveCocoa

```
self.accountService.muteAllNotifications(untilTime: date)
```

ReactiveCocoa

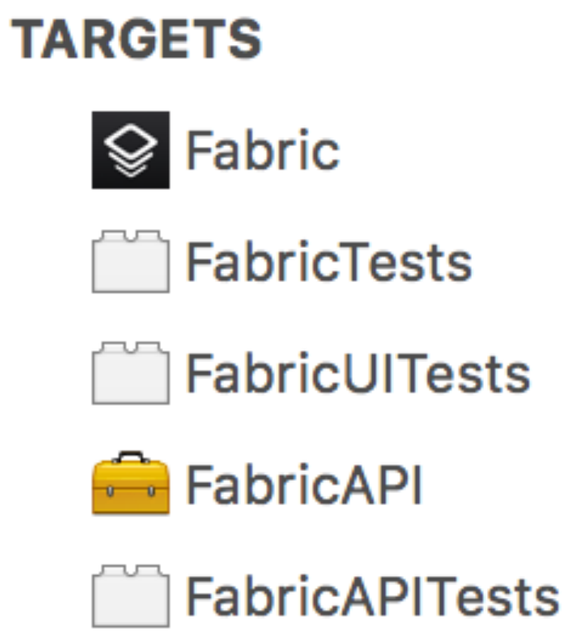
```
self.accountService.muteAllNotifications(untilTime: date)
    .continueWhenApplicationIsBackgrounded(taskName: "Muting notifications")
```

Tools

Architecture

Architecture

FabricAPI.framework



Architecture

- ~~Massive View Controllers~~
- View Models
- Glue-code View Controllers
- Table View Data sources

Architecture

`GenericTableViewDataSource`

Architecture

GenericTableViewDataSource

```
protocol TableSectionType {
    associatedtype AssociatedTableRowType: TableRowType

    var rows: [AssociatedTableRowType] { get }

    var title: String? { get }
}

protocol TableRowType { }
```

Architecture

GenericTableViewDataSource

```
final class GenericTableViewDataSource
<Elements,
SectionType: TableSectionType,
RowType: TableRowType
where SectionType.AssociatedTableRowType == RowType,
SectionType: Equatable,
RowType: Equatable>: NSObject, UITableViewDataSource
```

Architecture

GenericTableViewDataSource

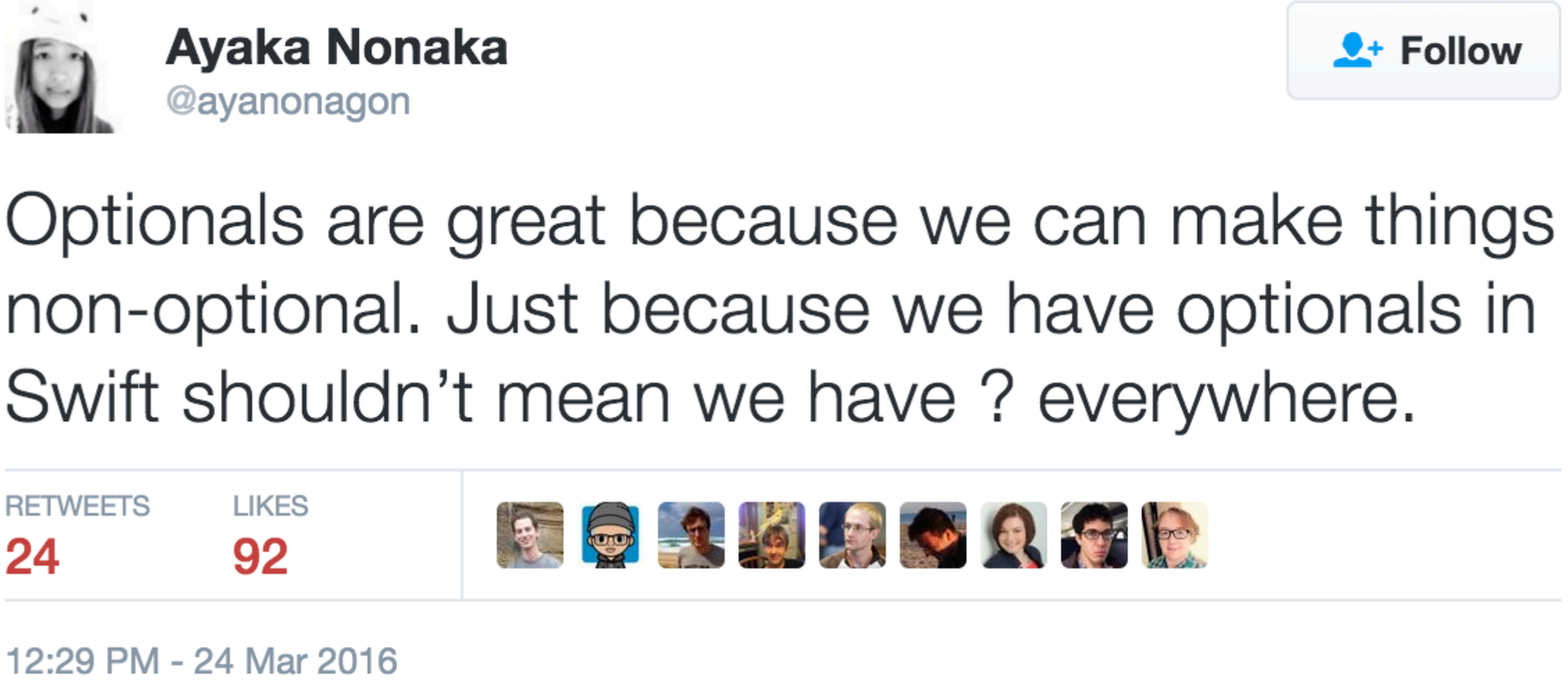
```
enum ProjectIssueRow: TableRowType, Equatable {
    case Loading
    case NoIssues
    case ErrorLoadingIssues
    case ProjectIssue(Issue)
}

return GenericTableViewDataSource(
        tableView: tableView,
        tableViewData: observableProperty, // Observable<Elements>
        computeSections: { elements in ... }, /// Pure function from `Elements` to `[SectionType]`
        configureRow: { row, indexPath in ... } /// Function from `RowType` to `UITableViewCell`
        )
```

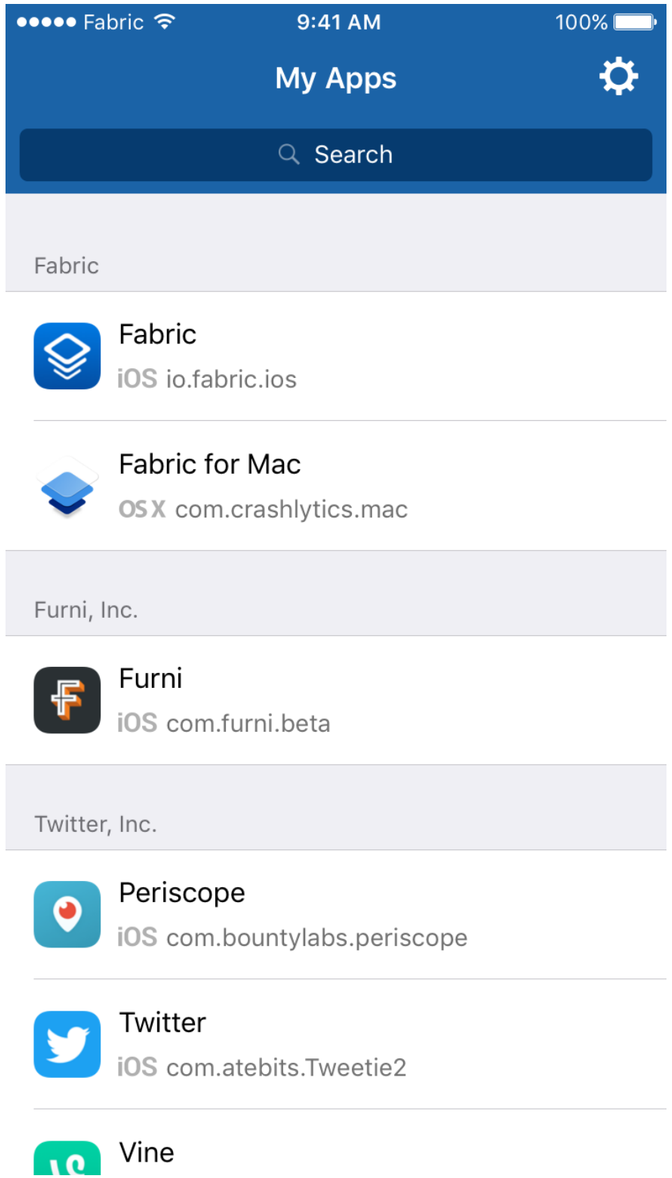
Better with Swift

Better with Swift

- Nullability
- Type-safe JSON parsing
- Code generation

Nullability




Fabric
9:41 AM
100%
My Apps
Search
Fabric
Fabric
iOS io.fabric.ios
Fabric for Mac
OS X com.crashlytics.mac
Furni, Inc.
Furni
iOS com.furni.beta
Twitter, Inc.
Periscope
iOS com.bountylabs.periscope
Twitter
iOS com.atebits.Tweetie2
Vine

Nullability

```
final class ApplicationListViewController: BaseFabricTableViewController
```

Nullability

```
final class ApplicationListViewController: BaseFabricTableViewController {
 override viewDidLoad() {
     super.viewDidLoad()

     let session = UserSession.currentUserSession

     if let session = session session {
         session.requestApplications()...
     }
     // or...
     session!.requestApplications()...
 }
```

Nullability

```
final class ApplicationListViewController: BaseFabricTableViewController {
 init(viewModel: ApplicationListViewModel)
}

final class ApplicationListViewModel {
 init(fabricAPI: AuthenticatedFabricAPI)
}

public final class AuthenticatedFabricAPI {
    public init(authResponse: AuthResponse)
}

public final class AuthResponse {
     let accessToken: String
}
```

Ayaka Nonaka
@ayanonagon

Follow

Optionals are great because we can make things non-optional. Just because we have optionals in Swift shouldn't mean we have ? everywhere.

RETWEETS 24 LIKES 92

12:29 PM - 24 Mar 2016

Nullability

```
final class ApplicationListViewModel {
    var applications: [Application]?
}
```

Nullability

```
enum DataLoadState<T> {
    case Loading
    case Failed
    case Loaded(T)
}

final class ApplicationListViewModel {
    var applications: DataLoadState<[Application]> = .Loading
}
```

Type-safe JSON parsing

JSON Parsing Anti-Patterns

```
public struct Application {
    public var ID: String?
    public var name: String?
    public var bundleIdentifier: String?

    public mutating func decode(j: [String: AnyObject]) {
            self.ID = j["id"] as? String
            self.name = j["name"] as? String
            self.bundleIdentifier = j["identifier"] as? String
    }
}
```

JSON Parsing Anti-Patterns

```
public struct Application {
    public let ID: String
    public let name: String
    public let bundleIdentifier: String

    public static func decode(j: [String: AnyObject]) -> Application? {
            guard let ID = j["id"] as? String,
                  let name = j["name"] as? String,
                  let bundleIdentifier = j["identifier"] as? String else { return nil }

        return Application(
            ID: ID,
            name: name,
            bundleIdentifier: bundleIdentifier
        )
    }
}
```

Type-safe JSON parsing

```
import Decodable /// https://github.com/Anviking/Decodable

public struct Application: Decodable {
    public let ID: String
    public let name: String
    public let bundleIdentifier: String

    public static func decode(j: AnyObject) throws -> Application {
        return try Application(
            ID: j => "id",
            name: j => "name",
            bundleIdentifier: j => "identifier"
        )
    }
}
```

Code Generation

Code Generation

```
/// Swift < 2.2
UIBarButtonItem(barButtonSystemItem: .Done, target: self, action: Selector("buttonTapped"))

// Swift 2.2
UIBarButtonItem(barButtonSystemItem: .Done, target: self, action: #selector(ViewController.buttonTapped))
```

Code Generation

```
super.init(nibName: "ViewControllerNibName", bundle: nil)

let nib = UINib(nibName: "NibName", bundle: nil)

tableView.registerNib(nib, forCellReuseIdentifier: "ReuseIdentifier")

let cell = tableView.dequeueReusableCellWithIdentifier("ReuseIdentifier", forIndexPath: indexPath) as! MyTableViewCell

let image = UIImage(named: "ImageName")!
```

Code Generation: R.swift

https://github.com/mac-cain13/R.swift

```
lane :code_gen do
  root_directory = "#{File.expand_path(File.dirname(__FILE__))}/../"

  sdk_path = sh("xcodebuild -version -sdk iphoneos9.2 Path")

  sh("#{root_directory}/Pods/R.swift/rswift -p \"#{root_directory}/Fabric.xcodeproj\" -t Fabric
end
```

Code Generation: R.swift

https://github.com/mac-cain13/R.swift

- Nibs
- Reuse Identifiers
- Image names in asset catalogs
- Other file names in the bundle

Code Generation: R.swift

https://github.com/mac-cain13/R.swift

```
super.init(nibResource: R.nib.myViewController)

class MyTableViewCell: UITableViewCell, ReusableNibTableViewCell {
    /// This will even fail to compile if it's not the right cell
    static let nibResource = R.nib.myTableViewCell
}

tableView.registerReusableNibCell(MyTableViewCell)

let cell = MyTableViewCell.dequeueFromTableView(tableView, indexPath)

let image = R.image.imageName
```

Error Reporting

Error Reporting

```
do {
    try fileManager.createDirectoryAtPath(path, withIntermediateDirectories: true, attributes: nil)
    /// ...
}
catch {
    print("Error: \(error)")
}
```

Error Reporting

```
do {
    try fileManager.createDirectoryAtPath(path, withIntermediateDirectories: true, attributes: nil)
    /// ...
}
catch {
    Crashlytics.sharedInstance().recordError(error, withAdditionalUserInfo: userInfo)
}
```

Error Reporting

Questions?

- @Javi
- javi@twitter.com
- fabric-app-ios@twitter.com

Thanks