Towards Safer, Better, Faster Code

Towards Safer, Better, Faster Code

(almost)

Constraints

- Code style (white space or new lines...).
- Frustrated with Swift build time.
- Stringly typed resource (storyboard, nib, image names, fonts)
- Time is limited

Towards Safer, Better, Faster Code

- **R.swift** -> safer
- **SwiftLint** -> better
- **Fastlane** -> faster

SwiftLint

"A tool to enforce Swift style and conventions. https://realm.io"

https://github.com/realm/SwiftLint

SwiftLint

Demo

SwiftLint

Command Line

```
$ swiftlint help
Available commands:

   autocorrect  Automatically correct warnings and errors
   help         Display general or command-specific help
   lint         Print lint warnings and errors for the Swift files in the current directory (default command)
   rules        Display the list of rules and their identifiers
   version      Display the current version of SwiftLint
```

SwiftLint

- Disable rules in code

Rules can be disabled with a comment inside a source file with the following format:

```
// swiftlint:disable <rule1> [<rule2> <rule3>...]
```

The rules will be disabled until the end of the file or until the linter sees a matching enable comment:

```
// swiftlint:enable <rule1> [<rule2> <rule3>...]
```

SwiftLint

For example:

```
// swiftlint:disable:next force_cast
let noWarning = NSNumber() as! Int
let hasWarning = NSNumber() as! Int
let noWarning2 = NSNumber() as! Int // swiftlint:disable:this force_cast
let noWarning3 = NSNumber() as! Int
// swiftlint:disable:previous force_cast
```

SwiftLint

Reference:
+ https://github.com/realm/SwiftLint
+ https://github.com/realm/SwiftLint/blob/master/Rules.md [good read]

R.swift

"Get strong typed, autocompleted resources like images, fonts and segues in Swift projects"

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

R.swift

- Why use this?

It makes your code that uses resources:
+ Fully typed, less casting and guessing what a method will return
+ Compile time checked, no more incorrect strings that make your app crash at runtime
+ Autocompleted, never have to guess that image name again

R.swift

before

```swift
let someController = UIStoryboard(name: "Foo", bundle: nil).instantiateViewController(withIdentifier: "SomeController") as? SomeController
```

after

```swift
let someController = R.storyboard.foo.someController // someController is typed SomeController
```

R.swift

- Supported types

R.swift currently supports these types of resources:
+ Images
+ Fonts
+ Resource files
+ Colors
+ Localized strings
+ Storyboards
+ Segues
+ Nibs
+ Reusable cells

R.swift

Demo

Fastlane

"fastlane is the easiest way to automate beta deployments and releases for your iOS and Android apps. Ã°Å¸Å¡â‚¬ It handles all tedious tasks, like generating screenshots, dealing with code signing, and releasing your application."

https://docs.fastlane.tools/

Fastlane

Demo

Fastlane

- Install

```
$ brew cask install fastlane
```

Fastlane

Gettings started guide

- iOS: https://docs.fastlane.tools/getting-started/ios/setup/
- Android: https://docs.fastlane.tools/getting-started/android/setup/

Fastlane

fastfiles from other companies

https://github.com/fastlane/examples

Towards Safer, Better, Faster Code

- **R.swift** -> safer
- **SwiftLint** -> better
- **Fastlane** -> faster

Summary

Pro:

- Faster build process (fastlane)
- Able Integrate CI and CD (fastlane)
- Fastlane support many tools
- Multi platform build (fastlane gym)
- Help write better Swift code (SwiftLint with the Rules.md guide)
- We may learn new things (SwiftLint)

Summary

Cons:

- May decrease build time: since we may want to integrate into Xcode via build phases, so more build phases means longer build time
- Diving deep will get complicated. Fastlane: for advanced build process, example: build -> upload appstore / app manager -> notify on slack... But we can search for example on Github.

Other tools

- **Sourcery**: Meta-programming for Swift, stop writing boilerplate code. https://github.com/krzysztofzablocki/Sourcery
- **SwiftGen**: The Swift code generator for your assets, storyboards, Localizable.strings, … — Get rid of all String-based APIs! https://github.com/SwiftGen/SwiftGen
- **Reusable**: A Swift mixin for reusing views easily and in a type-safe way (UITableViewCells, UICollectionViewCells, custom UIViews, ViewControllers, Storyboards…) https://github.com/AliSoftware/Reusable

Sample code: https://github.com/vinhnx/sbf_sample

Thanks!