

The Great Swift Migration

Self Intro

- Name: Josh Kaplan
- Company: GMO Pepabo
- App: minne
- Strengths: English
- Hobbies: Collecting Xcode versions

30,000 lines of Obj-C

Why?

- Future proofing
- Safety
- Productivity with features like generics, etc
- Security

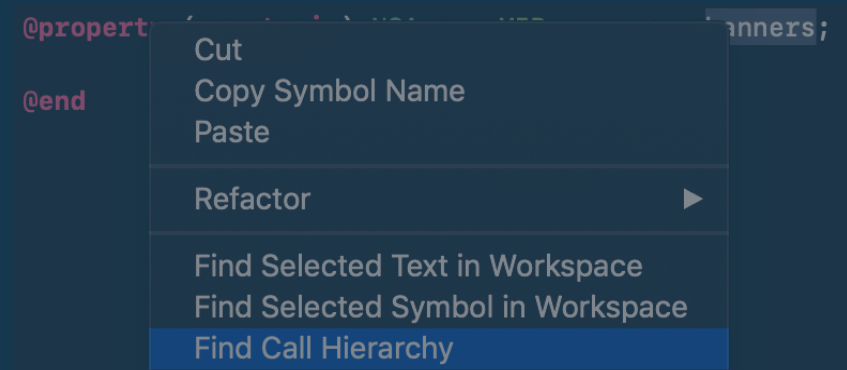
How?

Modernize Objective-C


```
@property (nonatomic, nonnull) NSArray<MICreator *> *creators;  
@property (nonatomic, nonnull) MICartPayment *payment;  
@property (nonatomic, nonnull) MIAPIClient *client;  
@property (nonatomic, nullable) MICoupon *coupon;
```

Check nullability in current Swift code

```
let num: Int? = nil
"\(num)" // String interpolation produces a debug description for an optional value
```



Choosing which areas to migrate

- First, modernize models
- Then, ViewControllers and ViewModels/Presenters
 - Work on feature/module basis
- Avoid converting code that is then used in obj-c
- Cells and smaller views can wait

Details

Steps

1. Move properties in m file to header and add annotations
2. Make Swift file and make extension
3. Migrate non-lifecycle methods and commit for each method
4. Make temporary method for lifecycle and IBAction method contents
5. Migrate lifecycle methods and properties
6. Update storyboard

Points

- Only *minor* refactoring
- Writing tests first helps
- Communicate with team to prevent conflicts
- Things like major renaming should be in separate PRs to make review easier
- Set goals and monitor progress with tools like *tokei*

Bonus: *Migrating to SwiftUI*

- Small components
- Use Xcode Preview now
- Fewer dependencies
- Swift 🤗