

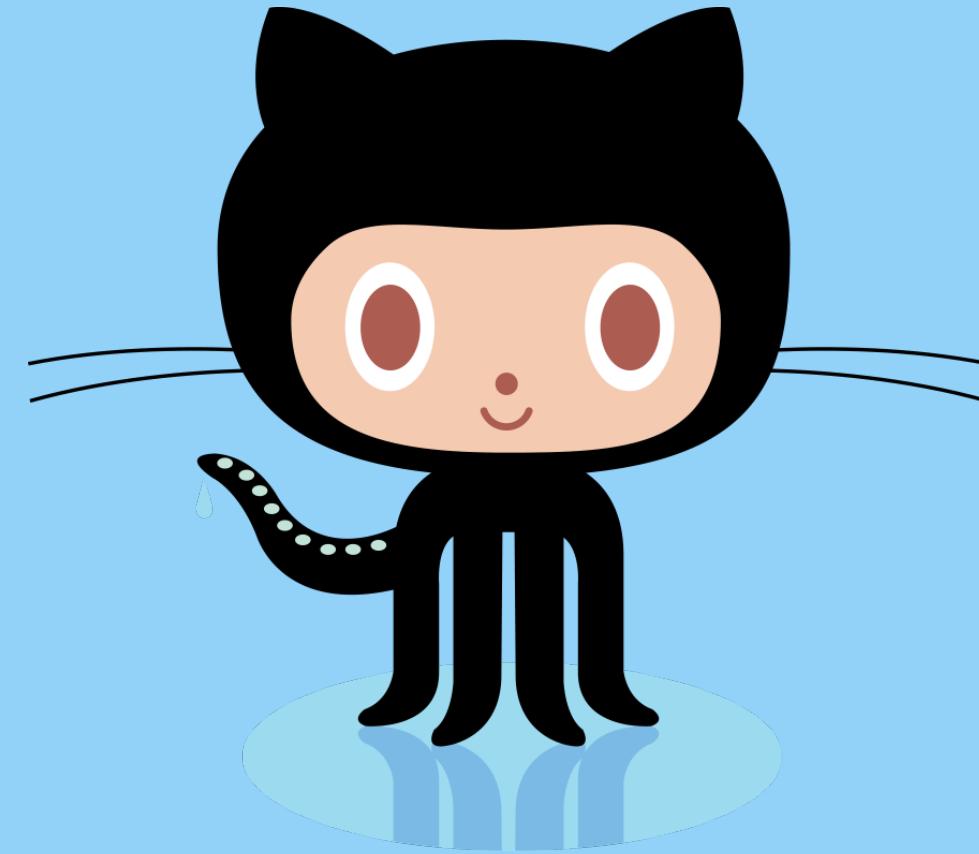
The First Swift Optimized Database

Making
compile-time type safety
play nice with
runtime magic

What is Realm?

- **Fast, zero-copy, embedded database**
- **Used in apps with *millions* of users**
- **NoSQL**
- **Full **ACID** transactions**
- **Well defined threading model**
- **Cross-platform C++ core with many language bindings**
(currently Objective-C, Swift & Android)

Open Source*



github.com/realm/realm-cocoa

* Bindings 100% open source, C++ core will launch as Apache 2

Core Data in Swift

Core Data goes against Swift's strong type safety and generics features.

```
let myPersonClass: AnyClass =  
    NSClassFromString("MyGreatApp.Person")
```

Apple had to invent special language
constructs in Swift to support Core
Data!!

Like the `@dynamic` attribute in Objective-C, the `@NSManaged` attribute informs the Swift compiler that the storage & implementation of a property will be provided at runtime.

Managed context in Core Data

```
@lazy var managedObjectContext: NSManagedObjectContext = {
    let modelURL = NSBundle.mainBundle().URLForResource("SwiftTestOne", withExtension: "momd")
    let mom = NSManagedObjectModel(contentsOfURL: modelURL)
    ZAssert(mom != nil, "Error initializing mom from: \(modelURL)")

    let psc = NSPersistentStoreCoordinator(managedObjectModel: mom)

    let urls = NSFileManager.defaultManager().URLsForDirectory(.DocumentDirectory, inDomains: .UserDomainMask)
    let storeURL = (urls[url.endIndex-1]).URLByAppendingPathComponent("SwiftTestOne.sqlite")

    var error: NSError? = nil

    var store = psc.addPersistentStoreWithType(NSSQLiteStoreType, configuration: nil, URL: storeURL, options: nil, error: &error)
    if (store == nil) {
        println("Failed to load store")
    }
    ZAssert(store != nil, "Unresolved error \(error?.localizedDescription), \(error?.userInfo)\nAttempted to create store at \(storeURL)")

    var managedObjectContext = NSManagedObjectContext()
    managedObjectContext.persistentStoreCoordinator = psc

    return managedObjectContext
}()
```

TOOLS

SwiftData

Simple and Effective SQLite Handling in Swift

github.com/ryanfowler/SwiftData

Made by Ryan Fowler, [@ryanfowler19](https://twitter.com/ryanfowler19)

```
if let err = SD.createTable("Cities",
    withColumnNamesAndTypes: ["Name": .StringVal,
                                "Population": .IntVal]) {
    //there was an error during this function, handle it here
} else {
    //no error, the table was created successfully
}
```

QueryKit

Swift CoreData query language

querykit.org

Made by Kyle Fuller, [@kylef](https://twitter.com/kylef)

```
Person.queryset(context).filter(Person.name == "Kyle").delete()
```

```
// Filtering
queryset.filter(Person.name == "Kyle")
queryset.exclude(Person.age < 21)
queryset.exclude(Person.isEmployed)
```

Realm in Swift

- **Generics**
- **Type Safety**
- **Default Values**
- **Swift Objects**
- **No Code Generation**
- **Zero Copy**

Realm Models

```
class Employee: Object {  
    dynamic var name = "" // you can specify defaults  
    dynamic var startDate = NSDate()  
    dynamic var salary = 0.0  
    dynamic var fullTime = true  
}
```

```
class Company: Object {  
    dynamic var name = ""  
    dynamic var ceo: Employee? // optional. who needs CEO's?!  
    let employees = List<Employee>()  
}
```

Using Realm

```
let company = Company() // Using Realm Objects  
company.name = "Realm" // etc...
```

```
defaultRealm().write { // Transactions  
    defaultRealm().add(company)  
}
```

```
// Queries  
let companies = objects(Company)  
companies[0].name // => Realm (generics)  
let ftJacks = objects(Employee) // "Jack"s who work full time  
    .filter(.fullTime == true && .name == "Jack")
```

Realm

Under the hood



Thou shalt do

no
evil

Thou shalt do

no

a convenient amount of

evil

Evil things Realm does

1. Introspecting Swift from Swift (👍)
2. Introspecting Swift from ObjC (😱)
3. Overriding property accessors (😎)
4. Emoji properties in ObjC (😬)
5. Generic properties in ObjC (🚀 !)

another time....

Resources

- This talk ([github.com](#))
- Realm on GitHub ([github.com](#))
- Realm's WIP Swift branch ([github.com](#))
- Writing Swift Classes With Objective-C Behavior ([apple.com](#))
- The Core Data Stack in Swift ([cimgf.com](#))

Thank You!

Meetup().questions?.askThem!
JP Simard, *@simjp, realm.io*