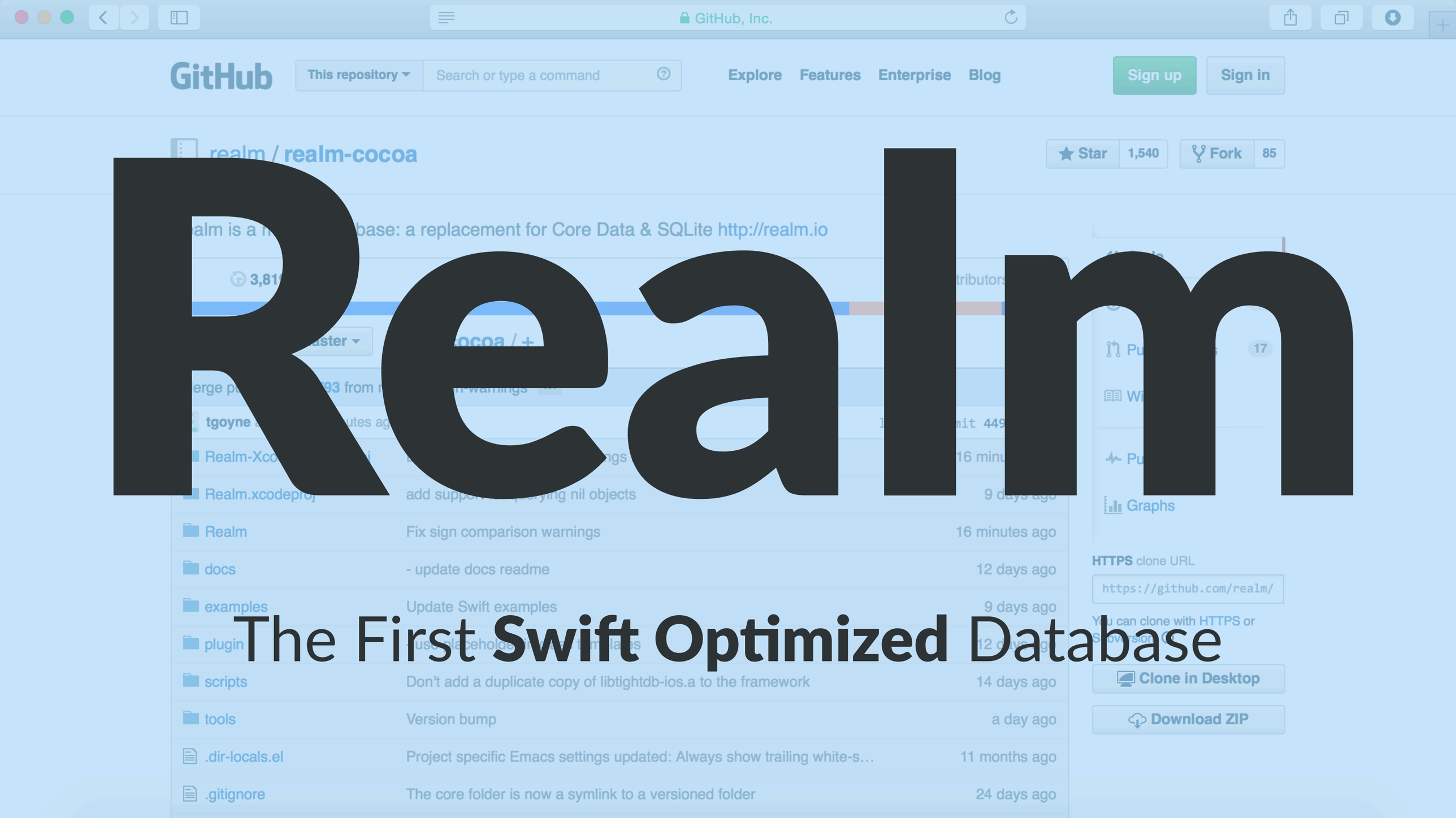
GitHub, Inc.
GitHub
This repository
Search or type a command
Explore
Features
Enterprise
Blog
Sign up
Sign in
realm / realm-cocoa
Star 1,540
Fork 85
Realm
The First Swift Optimized Database
Realm
Fix sign comparison warnings
16 minutes ago
docs
- update docs readme
12 days ago
examples
Update Swift examples
9 days ago
scripts
Don't add a duplicate copy of libtightdb-ios.a to the framework
14 days ago
tools
Version bump
a day ago
.dir-locals.el
Project specific Emacs settings updated: Always show trailing white-s…
11 months ago
.gitignore
The core folder is now a symlink to a versioned folder
24 days ago
Graphs
HTTPS clone URL
https://github.com/realm/
Clone in Desktop
Download ZIP

What is Realm?

- **Fast, embedded database** (zero-copy, not an ORM)
- **Used in apps with *millions* of users**
- **NoSQL**
- **Full ACID transactions**
- **Well defined threading model**
- **Cross-platform C++ core with many language bindings** (only Objective-C & Swift released)

Open Source*

github.com/realm/*realm-cocoa*

* Bindings 100% open source, C++ core being released 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[urls.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
}()
```

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()
  dynamic var employees = ArrayProperty(Employee)
}
```

Using Realm

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

// Transactions
defaultRealm().transaction() {
  realm.add(company)
}

// Querying objects
let companies = objects(Company)
companies[0]?.name // => Realm (generics)
let ftEmployees = objects(Employee).filter(.fullTime == true || .name == "John")
```

Work In Progress

- Change notifications/Live Results Sets
- Primary Keys/Upsert/JSON Mapping
- Delete Rules
- Bi-directional relationships
- Sync
- Open Source Core
- Android

Resources

- 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)**

Questions?

@simjp, jp@realm.io