Advanced Swift Generics

Перейдем на <T>

Max Sokolov - iOS dev @ Avito
masokolov@avito.ru
@max_sokolov

Agenda

- The problem
- Generics in Swift
- Protocols with associated types
- Declarative type-safe UIKit
- Generics in practise

```
class Matrix<T: Choiceable where T.T == T, T: Learnable, T: Fateable>
```

What for?

Generic code enables you to write **flexible**, **reusable** functions and types that can work with any type, subject to requirements that you define. You can write code that **avoids duplication** and expresses its intent in a **clear**, **abstracted** manner.

Apple docs

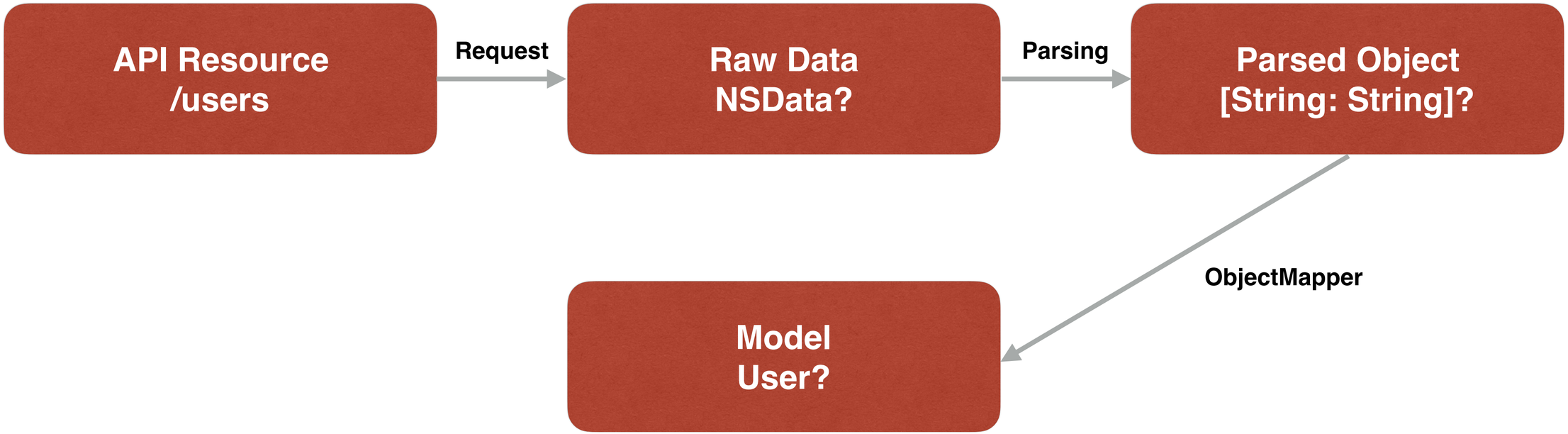

Segmentation fault: 11

Why?

- **One team** of 5 people
- **3 different apps** with shared components (mostly Swift)
- 20 private CocoaPods
- Large code base (~100k)

[go back];

Typical api request



Implementation

```
@implementation DataProvider

- (void)fetch:(NSString *)resources
 mappingClass:(Class)mappingClass
   completion:(void(^)(NSArray<id<Mappable>> *results))completion {

    // Load data...
    // Parse data...

    if ([mappingClass conformsToProtocol:@protocol(Mappable)]) {

        id result = [mappingClass initWithDictionary:response];

        completion(@[result]);
    }
}

@end
```

The problem?

```
@implementation DataProvider

- (void)fetch:(NSString *)resources
 mappingClass:(Class)mappingClass
   completion:(void(^)(NSArray<id<Mappable>> *results))completion {

    // Load data...
    // Parse data...

    if ([mappingClass conformsToProtocol:@protocol(Mappable)]) {

        id result = [mappingClass initWithDictionary:response];

        completion(@[result]);
    }
}

@end
```

RUN TIME!

What run time means?

```
DataProvider *provider = [DataProvider new];
[provider fetch:@"/users"
   mappingClass:[NSString class] // compiler doesn’t warn you here...
     completion:^(NSArray<User *> *results) {

}];
```

You are doing it wrong with Swift

```
let dataProvider = DataProvider()
dataProvider.fetch("/resource", mappingClass: User.self) { result in

    // code smell...
    guard let users = result.values as? [User] else {
        return
    }
}
```

<T>

Demo

UIKit<T>?

Next generation table views

UITableView powered by generics

90% of our screens are table views

●●●○○ actiDealer 11:14 AM 100%

Моя база

16 сентября в 11:57

Audi Q5, 2014

2.0 AMT (177 л.с.), полн., диз.

5 990 000 ₽

XML

20:57

Toyota Allex, 2003

1.6 AT (102 л.с.), перед., бен.

720 000 ₽

XML

Авито

27 августа в 10:36

Mercedes-Benz GL-klasse II (X166), 2009

2.0 MT (135 л.с.), задн., бен.

49 000 000 ₽

XML

9 мая в 13:20

Range Rover Discovery, 2007

Газ, полн.

19 720 000 ₽

XML

●●●○○ actiDealer 11:14 AM 100%

16 сентября в 11:57

Audi Q5, 2014

5 990 000 ₽

Публикации

Авито **Опубликовано**

Престижное авто **Ожидание**

Статистика

Avito

14 янв — 16 фев

3872 просмотра

126 запросов контакта

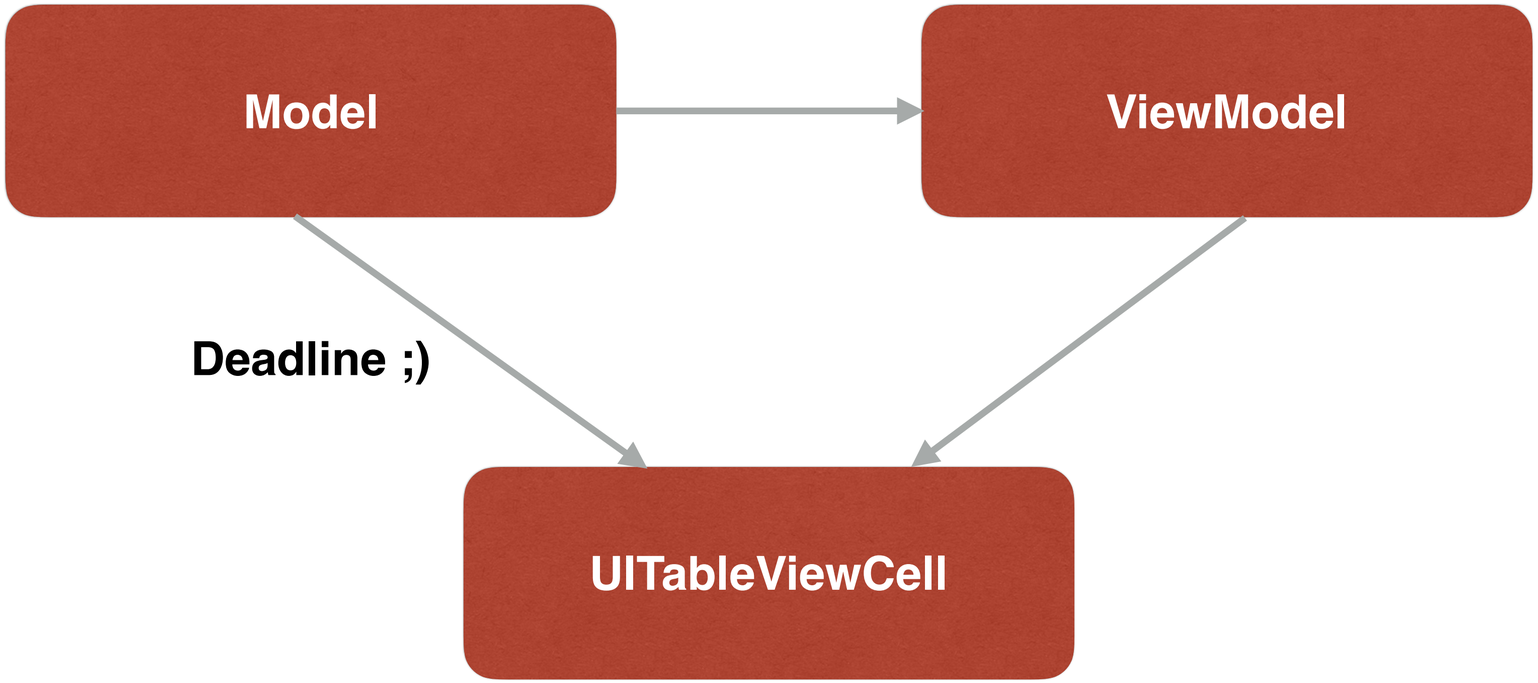
Параметры

Год	**2014**
Пробег, км	**1 800**
Двигатель	**2.0 AMT (177 л.с.)**
Топливо	**дизель**
Привод	**полный**

The problem with UITableView

- Boilerplate
- Painful when data sources are complex and dynamic
- **Not type safe** (dequeueReusableCellWithIdentifier etc.)

What table actually needs?



<DataType, CellType>

What if…

Line of code

```
let rowBuilder = TableRowBuilder<String, MyTableViewCell>(items: ["1", "2", "3"])
```

Demo

Tablet.swift

Open Source

https://github.com/maxsokolov/Tablet.swift

Tablet.swift

```
import Tablet

let row = TableRowBuilder<String, MyTableViewCell>(items: ["1", "2", "3"])
    .action(.click) { (data) in

    }
    .action(.willDisplay) { (data) in

    }

let section = TableSectionBuilder(headerView: nil, footerView: nil, rows: [row])

tableDirector += section
```

Generalization

NibLoadable

```
class MyView: UIView, NibLoadable {

}
```

```
let view = MyView.nibView() // view is MyView
```

NibLoadable / generic protocol

```
protocol NibLoadable {

    associatedtype T: UIView

    static func nibView() -> T?
}
```

```
extension NibLoadable where Self: UIView {

    static func nibView() -> Self? {

        return NSBundle(forClass: self)
            .loadNibNamed(String(self), owner: nil, options: nil)
            .first as? Self
    }
}
```

NibLoadable / generic protocol

```
protocol NibLoadable {

    associatedtype T: UIView

    static func nibView() -> T?
}
```

```
extension NibLoadable where Self: UIView {

    static func nibView() -> Self? {

        return NSBundle(forClass: self)
            .loadNibNamed(String(self), owner: nil, options: nil)
            .first as? Self
    }
}
```

NibLoadable / generic func

```
protocol NibLoadable {

    static func nibView<T: UIView>(viewType type: T.Type) -> T?
}


extension NibLoadable where Self: UIView {

    static func nibView<T: UIView>(viewType type: T.Type) -> T? {

        return NSBundle.mainBundle()
            .loadNibNamed(String(type), owner: nil, options: nil)
            .first as? T
    }

    static func nibView() -> Self? {

        return nibView(viewType: self)
    }
}
```

NibLoadable / generic func

```
protocol NibLoadable {

    static func nibView<T: UIView>(viewType type: T.Type) -> T?
}


extension NibLoadable where Self: UIView {

    static func nibView<T: UIView>(viewType type: T.Type) -> T? {

        return NSBundle.mainBundle()
            .loadNibNamed(String(type), owner: nil, options: nil)
            .first as! T
    }

    static func nibView() -> Self? {

        return nibView(viewType: self)
    }
}
```

Resume

- Currently, SDK/API’s are not generic
- Compiler is not perfect
- Can’t be used with Objective-C

- Generics help to know about potential issues earlier
- Compile-time type-safety makes your code better

Links

Covariance and Contravariance
https://www.mikeash.com/pyblog/friday-qa-2015-11-20-covariance-and-contravariance.html

Protocols with Associated Types, and How They Got That Way
http://www.slideshare.net/alexis_gallagher/protocols-with-associated-types-and-how-they-got-that-way

Why Associated Types?
http://www.russbishop.net/swift-why-associated-types

Tablet.swift
https://github.com/maxsokolov/Tablet.swift


THANK YOU!

QUESTIONS?

Max Sokolov - iOS dev @ Avito
https://github.com/maxsokolov
@max_sokolov