

## iWangKe.me

Oct 13 2014 iWangKe.me

# 为什么唱吧iOS 6.0选择了Mantle



最近唱吧iOS的6.0版本已经成功上线了。18人月的投入，2500个commit，几十万行的代码修改。唱吧iOS已经从内至外焕然一新，感谢一起并肩作战的小伙伴们。

6.0一个很重大的修改就是基于Mantle重建(新建)了Model层。这里不对Mantle作更多介绍，只分享一下使用Mantle的决策及执行过程。

## 我们遇到的问题

唱吧是一款上线2年多的App，产品形态的演进和迭代非常快。因此不可避免的遗留了各种问题：

- Model层不健全，没有统一的结构，不同工程师做法差异很大；多数是哑类型，且没有统一的序列化机制
- 业务逻辑冗余、分散、不一致
- 模块划分随意，依赖关系混乱，维护困难
- NSDictionary作为承载业务的数据类型在各处出现(sqlite, Model object, API, Notification, web, OpenURL etc.)，参数和值的正确性完全没有编译器检查，字符串很容易写错，风险延后至运行时，易产生低级bug
- 基本没有文档和注释(结合上一点，不挂debugger很难读懂代码)
- 几百个API，业务复杂，变动快，重构难；同一个API请求可能有重复和不一致
- API的一些参数和返回值，同一个参数/返回值可能存在类型差异；由于API需要向前兼容，修改API有成本

除此之外，还有其他工程上的约束：

- 不能影响现有的API，所有的事情只限于iOS端的修改
- 代码即文档，因为没有精力维护文档
- 对不同Model的持久化方式作迁移
- 避免写大段枯燥的Model的序列化/反序列化代码
- 没有时间造出足够成熟、健壮可重用的组件及撰写文档

上述的问题都是长期存在且需要解决的，否则严重影响开发效率及代码质量。11年的时候我还在做社交游戏的时候，设计并实现了一套简单的基于Objective-C Runtime的数值表Model结构及转换工具(Model<=>csv)供数值策划使用。但想写出一套成熟的方案还是有一些距离，而且也没有资源和时间作维护、测试和文档。

顺着这个思路找到了JSONModel和Mantle，前者刚刚1.0，后者在Github for Mac中广泛使用且社区更成熟(甚至Slack上有channel)，所以成为了更好的选择。

事实也证明这个选择是对的，6.0上线后，crash率比之前的版本有显示的降低，并且Mantle相关的crash占总crash的比率不到3%，大可以直接用在大型的产品上。

除了成熟稳定，Mantle基本解决了我们遇到了的所有问题。下面具体介绍一些通用性Mantle使用经验，基本的使用方法请直接移步Mantle的[README](#)。

## Property名称转换

由于API使用的开发语言与iOS所使用的Objective-C是截然不同的，所以可能将一些保留关键字作为property的名称(如id)，或者不小心override掉基类的属性(如description)。还有可能API中使用了一个很糟糕的名称，或者使用了不符合Objective-C命名规范的名称，这些我们都需要作转换。

只需要实现 MTLJSONSerializing protocol并在 +JSONKeyPathsByPropertyKey 方法中定义好新旧名称的映射关系即可，Mantle会在序列化及反序列化时对属性名进行自动的转换。

```
1  + (NSDictionary *)JSONKeyPathsByPropertyKey {
2      return @{
3          @"identifier": @"id",
4          @"displayDiscription": @"description",
5          @"thisIsANewShit": @"newShit",
6          @"creativeProduct": @"copyToChina",
7          @"betterPropertyName": @"m_wired_propertyName"
8      }
9  }
```

好了很多吧？没错，只需要定义一次名称的映射关系就可以了，Mantle负责model与JSON之间的双向转换。不需要将这种逻辑写得到处都是，并且还得维护它的一致性。

## Property的类型映射

iOS中处理URL使用的是NSURL类型，但JSON只支持基本的字符串，Mantle可以自动帮你转换成NSURL。

```
1  + (NSValueTransformer *)URLJSONTransformer {
2      return [NSValueTransformer valueTransformerWithName:MTLURLValueTransf
3  }
```

NSValueTransformer负责在不同类型间进行双向转换，请读者研究一下Mantle的实现方式。在此前提下，留给读者一个问题(其实这是一个真实的故事，类似的故事还有很多，详见iOS应用开发之十大坑队友)：

假设我们有一个entity，名字且叫KTVConcreteEntity吧，它有一个属性名字叫entityID，类型是NSInteger。问题来了，entityID可能在另外一个API的response中是字符串类型，在不直接修改Mantle的源码的前提下怎么搞？欢迎在下方留言讨论。

## 空标量异常

有的时候API的response会有空值，比如 copyToChina 可能不是每次都有的，JSON是这样儿的：

```
1  {  
2      "copyToChina": null  
3  }
```

Mantle在这种情况下会将newShit转换为nil，但如果是标量如NSInteger怎么办？KVC会直接raise `NSInvalidArgumentException`。

Mantle是基于KVC给property赋值的，KVC提供了 `-(void)setNilValueForKey:(NSString *)key` 方法，让我们为nil指定一个合理的替代值，我们来看一下此方法的解释：

Invoked by setValue:forKey: when it's given a nil value for a scalar value (such as an int or float).

Subclasses can override this method to handle the request in some other way, such as by substituting 0 or a sentinel value for nil and invoking setValue:forKey: again or setting the variable directly. The default implementation raises an `NSInvalidArgumentException`.

对于标量来讲，多数情况下合理的值即为0，我们来看下代码：

```
1  @interface MTLModel (KTVNullableScalar)  
2  @end  
3  @implementation MTLModel (KTVNullableScalar)  
4  - (void)setNilValueForKey:(NSString *)key {  
5      [self setValue:@0 forKey:key]; // For NSInteger/CGFloat/BOOL  
6  }  
7  @end  
8
```

问题完美解决，再也不需要到处写无聊的if/else了。

## 其它重要特性

Mantle为我们带来的方便不胜枚举：

- 实现了 `NSCopying` protocol，子类可以直接copy是多么爽的事情
- 实现了 `NSCoding` protocol，跟NSUserDefaults说拜拜
- 提供了 `-isEqual:` 和 `-hash` 的默认实现，model作NSDictionary的key方便了许多
- 简单且把一件事情做好，不掺杂网络相关的操作

如此强大优雅的设计，让我不得不向Github的工程师们致敬！

## 写在后面

篇幅所限，只介绍了几个典型的问题，欢迎大家讨论。但如果你的App的代码规模只有几万

行，或者API只有十几个，或者没有遇到我们这些遗留问题，我建议还是不要引入了，杀鸡用指甲刀就够了，杀不动多磨磨找准要害。Anyway，Mantle的实现和思路是值得每位iOS工程师学习和借鉴的。

附小广告一则：唱吧iOS团队诚招iOS工程师，推荐成功即奖励6000元现金或iPhone 6一部，详见这篇blog。

#ChangBa #iOS

## NEWER

唱吧客户端团队的Git实践

## OLDER

基于Core Animation的KTV歌词视图的平滑实现

26 条评论 iWangKe.me

1 登录 ▾

♥ Recommend 2  分享

按评分高低排序 ▾



加入讨论...



**Stephen Zhuang** · 9个月前

@interface ZXAccount : BaseModel<mtljsonserializing>

@property (nonatomic , strong) ZXUser \*user;  
@end

ZXAccount \*account = [MTLJSONAdapter modelOfClass:[ZXAccount class]  
fromJSONDictionary:JSON error:&error];  
这样子出来，为什么user是个NSDictionary?

1 ^ | ▾ · 回复 · 分享 ▾



**Liu Jiawei** · 3个月前

刚接手了一个项目的重构，看了下一共14W行，单纯从工程规模上，有引入Mantle的必要么？

^ | ▾ · 回复 · 分享 ▾



**Ke 管理员** → **Liu Jiawei** · 3个月前

其实引入Mantle并不复杂，但44W行的代码如果没有一个好的管理

 It depends, Mantle不是唯一的于段; 但14W行的code如果没有model层会有很大的问题。

^ | v · 回复 · 分享 ·



**fly2never** · 3个月前

按照最新的Mantle(1.5.4), setNilValueForKey已经不需要了, Mantle会自动转换为标量的默认值, 不会异常. 另外多keypath映射到同一个属性的问题也解决了. JSONKeyPathsByPropertyKey支持传递一个NSArray映射多个值, 参见 <https://github.com/Mantle/Mant...>

^ | v · 回复 · 分享 ·



**fly2never** → fly2never · 3个月前

看了一下,多keypath在2.0实现的, 只兼容iOS8了

^ | v · 回复 · 分享 ·



**Ke 管理员** → fly2never · 3个月前

我们目前确实没有多keypath的需要

^ | v · 回复 · 分享 ·



**jbas** · 6个月前

请问如何用mantle,把model转成json?

^ | v · 回复 · 分享 ·



**Jlang Yanyan** · 6个月前

我就想仰天长问, 为什么北美区IOS系统下的app store里没有唱吧啊~~~~~我也想没事了自己唱唱的啊~~~~~看着国内小伙伴们有事没事上传他们唱的歌~~都没我唱的好听好不好啊~~~已哭, 求安慰。

^ | v · 回复 · 分享 ·



**Ke 管理员** → Jlang Yanyan · 3个月前

只有国内的版权

^ | v · 回复 · 分享 ·



**Stephen Zhuang** · 9个月前

大神, mantle只支持NSNumber类型, 不支持单独的BOOL, NSInteger类型吗, 还有mantle的array property是怎样工作的, 我没有找到例子, 刚刚试了jsonmodel, 发现jsonmodel的array解析有问题啊- -

^ | v · 回复 · 分享 ·



**Ke 管理员** → Stephen Zhuang · 9个月前

1. 支持

2. See <https://github.com/Mantle/Mant...>

^ | v · 回复 · 分享 ·



**Stephen Zhuang** → Ke · 9个月前



那mantle是怎么知道我的array里放的是哪个model的?

1 ^ | v · 回复 · 分享 ›



**blankyao** → Stephen Zhuang · 8个月前

[NSValueTransformer

mtl\_JSONArrayTransformerWithModelClass:[XXModel class]];

^ | v · 回复 · 分享 ›



**Ke 管理员** → blankyao · 8个月前

快成support forum了==

^ | v · 回复 · 分享 ›



**Dev\_Ho** · 9个月前

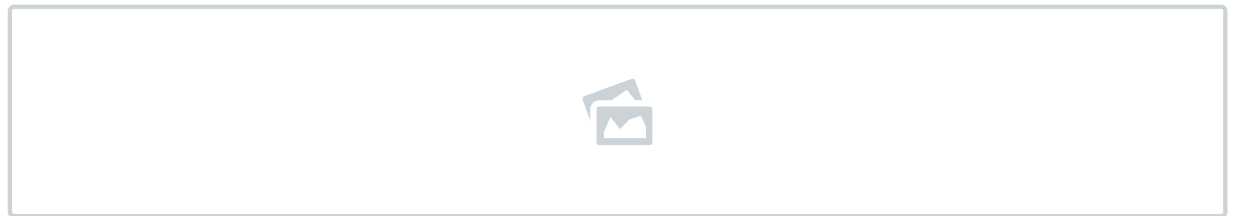
感谢分享 ^\_^.

^ | v · 回复 · 分享 ›



**Shuang** · 9个月前

11年的csv



^ | v · 回复 · 分享 ›



**Ke 管理员** → Shuang · 9个月前

哈爽爷，你还留着呢，来唱吧送iPhone6

^ | v · 回复 · 分享 ›



**Joe** · 9个月前

高质量的博文~

^ | v · 回复 · 分享 ›



**zeng** · 9个月前

可不可以这样搞？如果不修改Mantle的源码，新建一个基于NSValueTransformer的category然后重写+ (void)load方法，(每个category中的load方法都会被调用)里面实现可以模仿着Mantle中NSValueTransformer+MTLPredefinedTransformerAdditions.h的搞法，多加一个转换的ValueTransformerName及实现，最后在model中如果某个属性有返回值类型不统一的情况就加一个keyJSONTransformer转换，我写了个可以解决这种情形

^ | v · 回复 · 分享 ›



**fly2never** → zeng · 3个月前

但是对于多keypath怎么解决呢

^ | v · 回复 · 分享 ›





**Ke 管理员** → zeng • 9个月前

嗯就是这样

Best Regards

^ | v • 回复 • 分享



**zeng** → Ke • 9个月前

可不可以自动检测和转换呢？就是不需要写keyJSONTransformer转换，property中声明了属性的类型，不知道runtime时能不能自省一个属性的类型？

^ | v • 回复 • 分享



**Ke 管理员** → zeng • 9个月前

非常难而且没必要，你觉得呢

Best Regards

^ | v • 回复 • 分享



**zeng** → Ke • 9个月前

我也觉得，上次我们老大问我能不能这样，我就蒙了。主要还是服务器端要用测试用例来保障类型一致，然后定义接口格式时也得一致，前端总不能每个值都得判断

^ | v • 回复 • 分享



**Ke 管理员** → zeng • 9个月前

所以没必要解决一个不是问题的问题

Best Regards

^ | v • 回复 • 分享



**Machackx** • 9个月前

谢谢分享 :)

^ | v • 回复 • 分享

在 **IWANGKE.ME** 上还有.....

这是什么？

**关于IDP不得不说的事情**

1条评论 • 2年前

**由唱吧的音效改版谈Audio Unit的使用**

1条评论 • 3个月前

Search

**IWANGKE.ME**

© 2015 iWangKe.me

Modify from Apollo theme, designed by SANOGRAPHIX.NET

<http://www.iwangke.me/2014/10/13/Why-Changba-iOS-choose-Mantle/>



