

前一个月，我接触到了Mantle,由于项目采用的是MVC的设计模式，选用好的model也是至关重要的。先介绍下Mantle的使用吧。

首先定义好数据模型：

```
@property (nonatomic, copy) NSString *address;

@property (nonatomic, copy) NSString *contact_info;

@property (nonatomic, copy) NSString *contact_name;

@property (nonatomic, assign) NSInteger id;

@property (nonatomic, copy) NSString *open_time;

@property (nonatomic, copy) NSString *project_id;

@property (nonatomic, copy) NSString *project_name;

@property (nonatomic, copy) NSString *project_situation;

@property (nonatomic, copy) NSString *title_name;

@property (nonatomic, copy) NSString *url;
```

然后.m文件中实现如下方法：

```
+(NSDictionary *)JSONKeyPathsByPropertyKey{

    return @{};

}
```

使用

```
_biddingModel=[MTLJSONAdapter modelOfClass:
[TTBiddingModel class] fromJSONDictionary:dicerror:nil];
```

会什么会return呢？后面我会讲到。

参考了《为什么唱吧iOS6.0选择了Mantle》<http://www.cocoachina.com/ios/20141016/9931.html>

JSONKeyPathsByPropertyKey这个协议方法是将返回json中的内容序列化

```
+ (NSDictionary *)JSONKeyPathsByPropertyKey {
    return @{
        @"identifier": @"id",
        @"displayDescription": @"description",
        @"thisIsANewShit": @"newShit",
        @"creativeProduct": @"copyToChina",
        @"betterPropertyName": @"m_wired_propertyName"
    };
}
```

对属性名进行自动转换。

so为什么我会return{}呢，我们来看一看mantle的源代码
首先它定义了这样一个属性

```
@property (nonatomic, copy, readonly) NSDictionary *JSONKeyPathsByPropertyKey;
```

此时_JSONKeyPathsByPropertyKey就是我们刚刚用到的

```
+ (NSDictionary *)JSONKeyPathsByPropertyKey {};
```

```
_JSONKeyPathsByPropertyKey = [[modelClass JSONKeyPathsByPropertyKey] copy];
```

init 方法里面

```
- (id)initWithModel:(MTLModel<MTLJSONSerializing> *)model {  
    NSParameterAssert(model != nil);  
    self = [super init];  
    if (self == nil) return nil;  
    _model = model;  
    _modelClass = model.class;  
    _JSONKeyPathsByPropertyKey = [[model.class JSONKeyPathsByPropertyKey] copy];  
    return self;  
}
```

这里我cocopods的Mantle是1.5版本的，而最新的2.0.2版本舍去掉了如下方法：

```
- (NSString *)JSONKeyPathForPropertyKey:(NSString *)key {  
    NSParameterAssert(key != nil);  
    id JSONKeyPath = self.JSONKeyPathsByPropertyKey[key];  
    if ([JSONKeyPath isEqual:NSNull.null]) return nil;  
    if (JSONKeyPath == nil) {  
        return key;  
    } else {  
        return JSONKeyPath;  
    }  
}
```

所以，这就是我为什么会在协议方法里return {}，这里应该很明白了吧，序列化时，它会自动匹配property定义的属性。有没有很方便呢？