

YYModel-Json 解析

1. 简述：YYModel 是 ibireme 郭曜源开发的 YYKit 工具库的组件之一，是一个高性能的 ios Json 模型框架。功能和 JsonModel 差不多，但有着更好的性能和容错性
2. 用法：

(1) 简单： Json 转 Model

```
//JsonString:{
    “uid” :123456,
    “name” :” Harry” ,
    “created” :” 1965-07-31”
}

//jsonDic:@{
    @” uid” :@” 123456” ,
    @ “name” :@” Harry” ,
    @” created” :@” 1965-07-31”
}

//Model:

@interface User:NSObject

@property UInt64 uid;
@property NSString *name;
@property NSDate *created;

@end

// 从 Json 转到 Model

User *user1 = [User yy_modelWithJson:jsonString];
User *user1 = [User yy_modelWithDictionary:jsonDic];
User *user1 = [User yy_modelWithData:jsonData];

//从 Model 转为 json

NSDictionary *jsonDic = [user1 yy_modelToJSONObject];
```

```
NSData *jsonData = [user1 yy_modelToJSONData];
```

```
NSString *jsonString = [user1 yy_modelToJSONString];
```

注: model 属性命名与 json 的 key 值必须一致

(2) 将 Model 的属性匹配到 JSON/NSDictionary 键值

```
//JsonString:
```

```
{
    "n" : " Harry Pottery" ,
    "p" :256,
    "ext" :{
        "desc" : " A book written by J.K.Rowing"
    }
}
```

```
}
```

```
//JsonDictionary
```

```
@{
    @" n" :@" Harry Pottery" ,
    @" p" :256,
    @" ext" :@{
        @" desc" ;@"  A book written by J.K.Rowing"
    }
}
```

```
}
```

```
//Model:
```

```
@interface Book:NSObject
```

```
@property NSString *name;
```

```
@property NSInteger page;
```

```
@property NSString *desc;
```

```
@end
```

```
@implementation Book
```

```

+ (NSDictionary *)modelCustomPropertyMapper {
    return @{@"name" : @"n",
            @"page" : @"p",
            @"desc" : @"ext.desc"
    };
}

```

(3) Model 属性关联到其他 Mode

//JSON:

```

{
    "author" : {
        "name" : "J.K. Rowling",
        "birthday" : "1995-07-31"
    }
    "name" : "Harry Potter",
    "pages" : 256
}

```

//Dict

```

@{
    @"author" : @{
        @"name" : @"J.K. Rowling",
        @"birthday" : @"1995-07-31"
    }
    "name" : "Harry Potter",
    "pages" : 256
}

```

//Model

```

@interface Author:NSObject

```

```

@property NSString name;

@property NSData *birthday;

@end

@interface Book:NSObject

@property NSString *name;

@property NSUInteger pages;

@property Author *author;

@end

```

注：默认什么都不做，YYModel 可自动转换，前提：所有属性名和 Json 里 key 一致

(4) Model 属性是容器类型

```

@class Shadow, Border, Attachment

@interface Attributes

@property NSString *name;

@property NSArray *shadows;

@property NSSet *borders;

@property NSMutableArray *attachments;

@end

@implementation Attributes

+ (NSDictionary *)modelContainerPropertyGenericClass{

    return @{

        @" shadows" :[Shadow class],

        @" borders" :Border.class,

        @" attachments" :@" Attachment"

    }

}

```

注：通过实现协议中 `modelContainerPropertyGenericClass`，返回 `Model` 属性容器中需要存放的对象类型，`YYModel` 会自动处理。对象类型可以是 `Class` 或 `Class name`

(5) 黑名单白名单

```
@interface User

@property NSString *name;
@property NSUInteger age;

@end

@implementation User

+ (NSArray *)modelPropertyBlacklist {
    return @[@" age" ];
}

+ (NSArray *)modelPropertyWhitelist {
    return @[@" name" ];
}
```

如果一个 `Model` 需忽略某个属性，可实现协议中的 `modelPropertyBlacklist` 来返回属性名列表，`YYModel` 会在处理过程忽略掉这些属性

如果一个 `Model` 只需处理个别属性，可实现协议中的 `modelPropertyWhitelist` 来返回属性名列表，`YYModel` 在处理中只会处理列表内属性

(6) Json 数组转 Model

```
//Json:

[
    { "md5" : " abc" , " name" : " aeefa" },
    { "md5" : " cde" , " name" : " faf" }
]

NSArray *updateInfo = [NSJsonSerialization JSONObjectWithData:data
options:...Leaves];

//Model
```

```

@interface MYUpdateInfoResponed:NSObject

@property (nonatomic, strong) NSArray *updateInfo;

+ modelWithDictionary:(NSDictionary *)dictionary:(NSDictionary
*)dictionary;

@end

@interface updateInfo:NSObject

@property NSString *name;

@property NSString *md5;

@property NSString *url;

@end

@implementation MYUpdateInfoResponed:NSObject

+modelWithDictionary:{

    return [self yy_modelWithDictionary:dictionary];

}

+modelContainerPropertyGenericClass{

    return @{@"” updateInfo” :[updateInfo class]};

}

@end

```

外部调用

```

MYUpdateInfoResponed *updateInfoResponed = [MYUpdateInfoResponed
modelWithDictionary:@{@"” updateInfo” :updateInfoArr}];

```