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：
3. 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

1. 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”

};  
}

1. 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

1. Model属性是容器类型

@class Shadow,Border,Atachment

@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

1. 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

1. Json数组转Model

//Json:

[

{“md5”:”abc”,”name”:”aefa”},

{“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}];