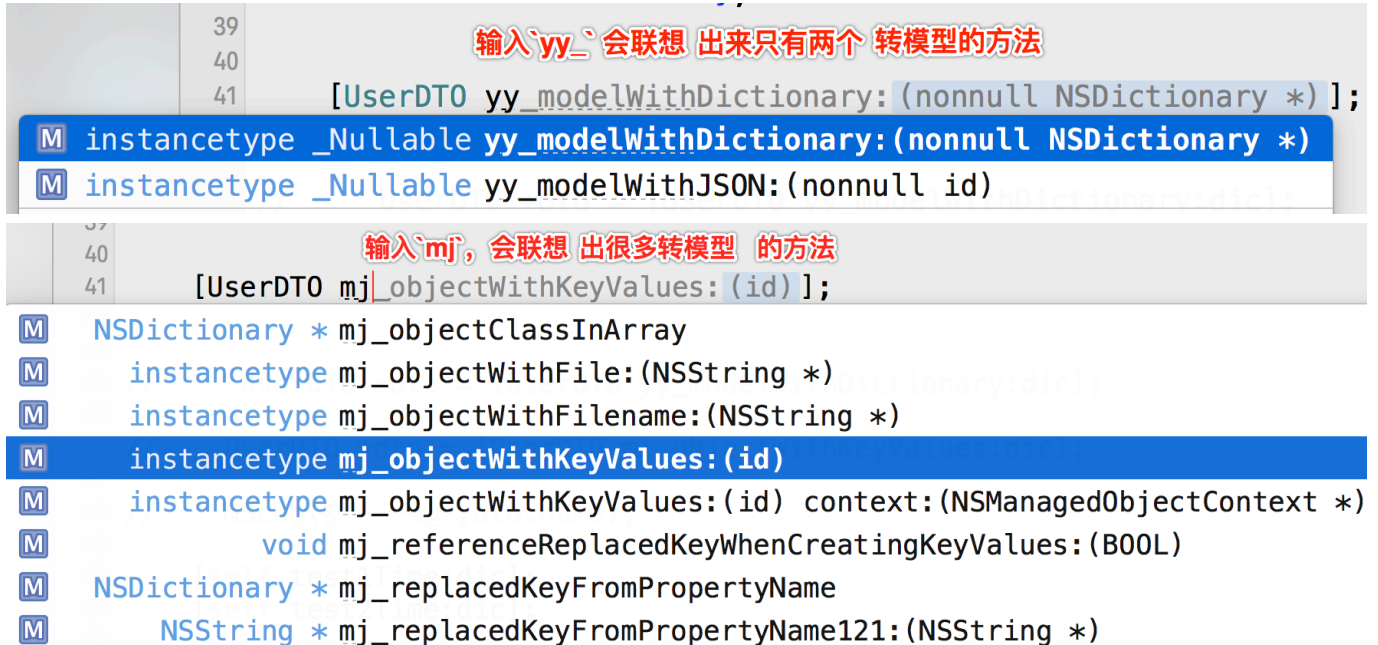


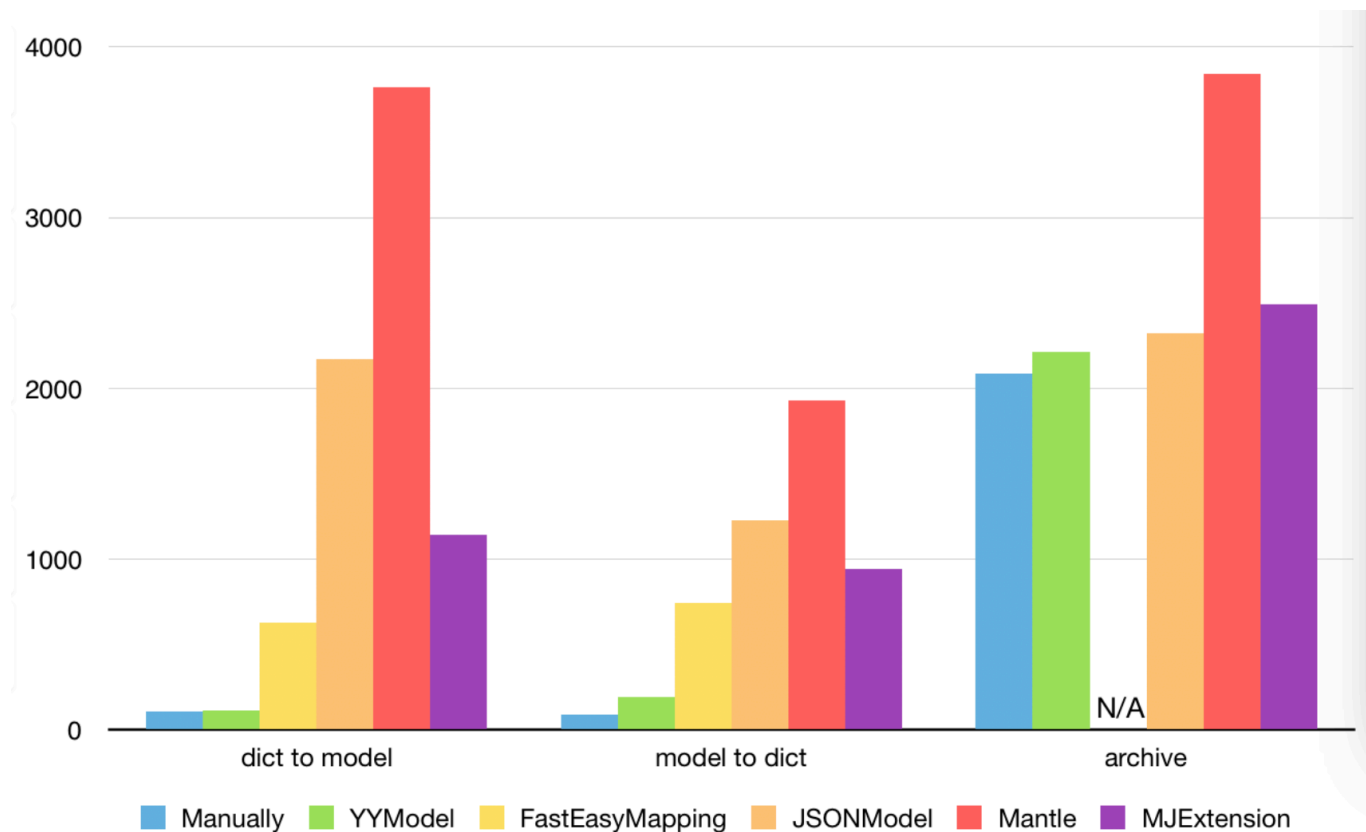
YYModel 与 MJExtension优缺点

- YYModel是一个比MJExtension较轻量级的一个字典转模型框架。MJExtension框架自带的文件多大21个，而YYModel自带的文件只有5个。如，输入yy会联想出来两个转模型的方法，而输入mj会联想出很多转模型的方法。不说了，直接上图：



通过字典来创建一个模型

- YYModel 效率要比 MJExtension 高。例如简单的循环10000次，MJExtension:0.12S,YYModel:0.02S
- MJExtension兼容性更好，涉及的逻辑判断远远大于YYModel。正常的网络模型转换，不会上千次、上万次，除非是本地数据库转换，100次以内，差距是几乎可以忽略的。
- 类型安全：YYModel在转换过程中所有的类型都会被验证，以确保类型安全。
- 高性能：YYModel转换效率接近手写代码。
- 性能比较图：



YYModel的底层核心是运用runtime获取类结构体中Ivars，进行KVC操作，然后根据不同情况进行分别处理。

YYModel用法

1.字典转模型-----

```
// 将 JSON (NSData,NSString,NSDictionary) 转换为Model:
User *user = [User yy_modelWithJSON:json];
```

2.模型转字典-----

```
// 将 Model 转换为 JSON 对象:
NSDictionary *json = [user yy_modelToJSONObject];
```

3.JSON/Dictionary 中的对象类型与 Model 属性不一致时，YYModel 将会进行如下自动转换。自动转换不支持的值将会被忽略，以避免各种潜在的崩溃问题。-----

JSON/Dictionary	Model
NSString	NSNumber,NSURL,SEL,Class
NSNumber	NSString
NSString/NSNumber	C number (BOOL,int,float,NSUInteger,UInt64,...)
NSString	NSDate parsed with these formats: yyyy-MM-dd

```

        yyyy-MM-dd HH:mm:ss
        yyyy-MM-dd'T'HH:mm:ss
        yyyy-MM-dd'T'HH:mm:ssZ
        EEE MMM dd HH:mm:ss Z yyyy
NSDate      NSString formatted with ISO8601:
            "YYYY-MM-dd'T'HH:mm:ssZ"
NSNumber    struct (CGRect,CGSize,...)
NSNull      nil,0
"no","false",...  @(NO),0
"yes","true",...  @(YES),1

```

4.Model 属性名和 JSON 中的 Key 不相同-----

```

// JSON:
{
    "n":"Harry Pottery",
    "p": 256,
    "ext" : {
        "desc" : "A book written by J.K.Rowing."
    },
    "ID" : 100010
}

// Model:
@interface Book : NSObject
@property NSString *name;
@property NSInteger page;
@property NSString *desc;
@property NSString *bookID;
@end

@implementation Book
//返回一个 Dict, 将 Model 属性名对映射到 JSON 的 Key。
+ (NSDictionary *)modelCustomPropertyMapper {
    return @{@"name" : @"n",
            @"page" : @"p",
            @"desc" : @"ext.desc",
            @"bookID" : @"[@"id",@"ID",@"book_id"]}];
}

```

5.Model 包含其他 Model-----

```

{
    "author":{
        "name":"J.K.Rowling",
        "birthday":"1965-07-31T00:00:00+0000"
    },
    "name":"Harry Potter",
    "pages":256
}

```

```

}

// Model: 什么都不用做, 转换会自动完成
@interface Author : NSObject
@property NSString *name;
@property NSDate *birthday;
@end

@implementation Author
@end

@interface Book : NSObject
@property NSString *name;
@property NSUInteger pages;
@property Author *author; //Book 包含 Author 属性
@end

@implementation Book
@end

@end

```

6. 容器类属性-----

```

@class Shadow, Border, Attachment;

@interface Attributes
@property NSString *name;
@property NSArray *shadows; //Array<Shadow>
@property NSSet *borders; //Set<Border>
@property NSMutableDictionary *attachments; //Dict<NSString, Attachment>
@end

@implementation Attributes
// 返回容器类中的所需要存放的数据类型 (以 Class 或 Class Name 的形式)。
+ (NSDictionary *)modelContainerPropertyGenericClass {
    return @{@"shadows" : [Shadow class],
            @"borders" : Border.class,
            @"attachments" : @"Attachment" };
}
@end

```

7. 黑名单与白名单-----效率提升问题, -----

```

@interface User
@property NSString *name;
@property NSUInteger age;
@end

@implementation Attributes

```

```

// 如果实现了该方法, 则处理过程中会忽略该列表内的所有属性。黑名单, 故名思议, 黑名单中的属性不会参与字典转模型
+ (NSArray *)modelPropertyBlacklist {
    return @[@"test1", @"test2"];
}
// 如果实现了该方法, 则处理过程中不会处理该列表外的属性。白名单使用比较极端, 你用了之后, 只有白名单中的属性会参与字典转模型, 其他属性都不参与。不推荐使用。
+ (NSArray *)modelPropertyWhitelist {
    return @[@"name"];
}
@end

```

8. 数据校验与自定义转换-----

```

// JSON:
{
    "name": "Harry",
    "timestamp" : 1445534567    //时间戳
}

// Model:
@interface User
@property NSString *name;
@property NSDate *createdAt;
@end

@implementation User
// 当 JSON 转为 Model 完成后, 该方法会被调用。
// 你可以在这里对数据进行校验, 如果校验不通过, 可以返回 NO, 则该 Model 会被忽略。
// 你也可以在这里做一些自动转换不能完成的工作。
- (BOOL)modelCustomTransformFromDictionary:(NSDictionary *)dic {
    NSNumber *timestamp = dic[@"timestamp"];
    if (![timestamp isKindOfClass:[NSNumber class]]) return NO;
    _createdAt = [NSDate dateWithTimeIntervalSince1970:timestamp.floatValue];
    return YES;
}

// 当 Model 转为 JSON 完成后, 该方法会被调用。
// 你可以在这里对数据进行校验, 如果校验不通过, 可以返回 NO, 则该 Model 会被忽略。
// 你也可以在这里做一些自动转换不能完成的工作。
- (BOOL)modelCustomTransformToDictionary:(NSMutableDictionary *)dic {
    if (!_createdAt) return NO;
    dic[@"timestamp"] = @(n.timeIntervalSince1970);
    return YES;
}
@end

```

- 在YYModel盛行的时候，其他插件都可以不用，但是ESJsonFormat必不可缺.
- ESJsonFormat 的安装:<https://github.com/czhen09/ESJsonFormatForMac>
- 界面功能如下:
 1. 支持输入Url，发送请求获取Json数据、.h文件、.m文件内容;
 2. 支持直接在Json中输入Json数据进行转换;
 3. 参数和接口分离;
 4. 支持.h、.m文件内容的复制清除;
 5. 支持Swift和OC语言;
 6. 缺陷：如果JSON里"birthday":"1965-07-31T00:00:00+0000"输出的只会是NSString类型的，不会是NSDate类型的
 7. 不会在model里对每个属性进行解释
- 使用:程序下载下来，跑起来之后，之后在Dock中保留就可以了;