



马哥教育  
www.magedu.com



2016全新Linux运维实战+python实战班震撼上线！  
先就业，后付款！年薪24W！学徒班还有少量名额！

首页  
最新文章  
开发  
IT技术  
职场  
业界  
极客  
创业  
访谈  
在国外

- 导航条 -

伯乐在线 > 首页 > 所有文章 > Web前端 > 前后端分离的思考与实践（三）

## 前后端分离的思考与实践（三）

2014/04/18 · [Web前端](#), [开发](#) · [Node.js](#), [Web开发](#)

分享到：

16

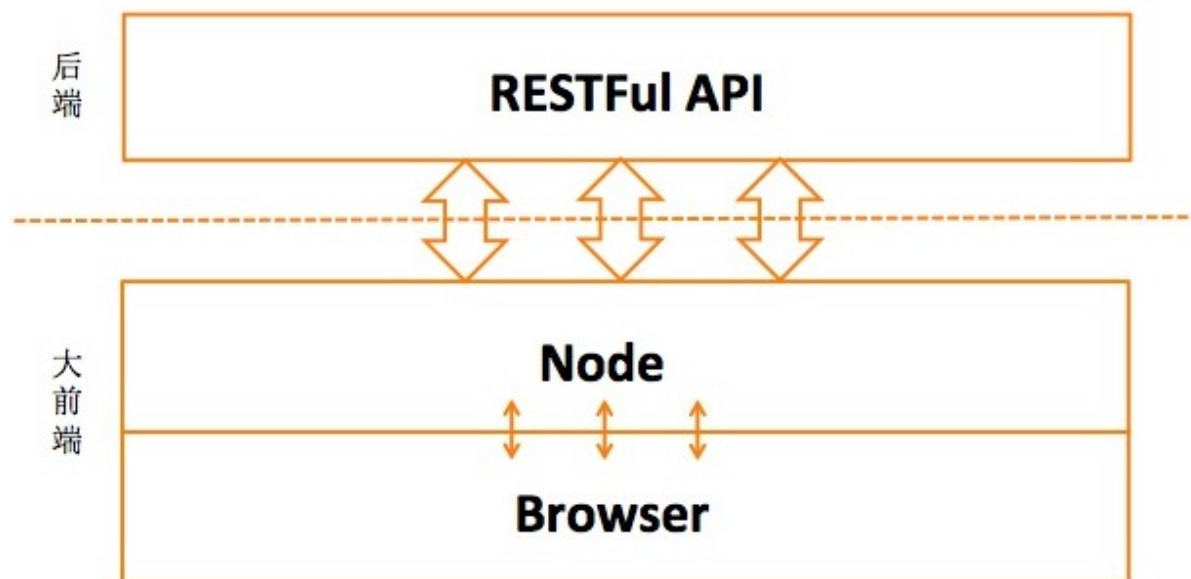
[MongoDB 在线讲座系列之MongoDB数据库...](#)  
[Gopher China 2015 上海大会](#)  
[MongoDB中文社区-北京专场（二）](#)  
[Unity 3D游戏开发之脚本系统](#)

原文出处：[淘宝UED - 善繁](#)

# Midway-ModelProxy — 轻量级的接口配置建模框架

## 前言

使用Node做前后端分离的开发模式带来了一些性能及开发流程上的优势(见[《前后端分离的思考与实践一》](#))，但同时也面临不少挑战。在淘宝复杂的业务及技术架构下，后端必须依赖Java搭建基础架构，同时提供相关业务接口供前端使用。Node在整个环境中最重要的工作之一就是代理这些业务接口，以方便前端(Node端和浏览器端)整合数据做页面渲染。如何做好代理工作，使得前后端开发分离之后，仍然可以在流程上无缝衔接，是我们需要考虑的问题。本文将就该问题做相关探讨，并提出解决方案。



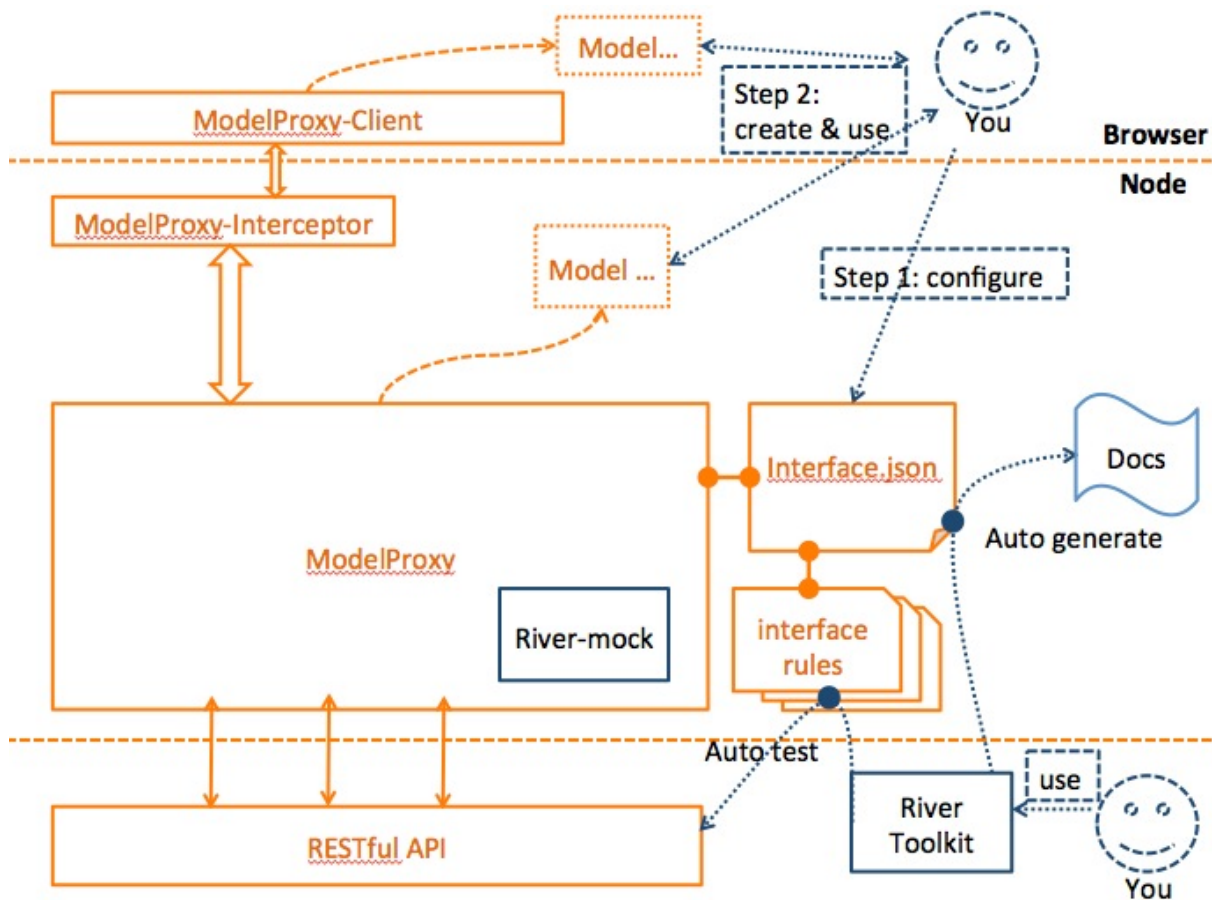
由于后端提供的接口方式可能多种多样，同时开发人员在编写Node端代码访问这些接口的方式也有可能多种多样。如果我们在接口访问方式及使用上不做统一架构处理，则会带来以下一些问题：

1. 每一个开发人员使用各自的代码风格编写接口访问代码，造成工程目录及编码风格混乱，维护相对困难。
2. 每一个开发人员编写自己的mock数据方式，开发完毕之后，需要手工修改代码移除mock。
3. 每一个开发人员为了实现接口的不同环境切换(日常，预发，线上)，可能各自维护了一些配置文件。
4. 数据接口调用方式无法被各个业务model非常方便地复用。
5. 对于数据接口的描述约定散落在代码的各个角落，有可能跟后端人员约定的接口文档不一致。
6. 整个项目分离开发之后，对于接口的联调或者测试回归成本依然很高，需要涉及到每一个接口提供者和使用者。

于是我们希望有这样一个框架，通过该框架提供的机制去描述工程项目中依赖的所有外部接口，对他们进行统一管理，同时提供灵活的接口建模及调用方式，并且提供便捷的线上环境和生产环境切换方法，使前后端开发无缝结合。ModelProxy就是满足这样要求的轻量级框架，它是Midway Framework 核心构件之一，也可以单独使用。使用ModelProxy可以带来如下优点：

1. 不同的开发者对于接口访问代码编写方式统一，含义清晰，降低维护难度。
2. 框架内部采用工厂+单例模式，实现接口一次配置多次复用。并且开发者可以随意定制组装自己的业务Model(依赖注入)。
3. 可以非常方便地实现线上，日常，预发环境的切换。
4. 内置`river-mock`和`mockjs`等mock引擎，提供mock数据非常方便。
5. 使用接口配置文件，对接口的依赖描述做统一的管理，避免散落在各个代码之中。
6. 支持浏览器端共享Model，浏览器端可以使用它做前端数据渲染。整个代理过程对浏览器透明。
7. 接口配置文件本身是结构化的描述文档，可以使用`river`工具集合，自动生成文档。也可使用它做相关自动化接口测试，使整个开发过程形成一个闭环。

## ModelProxy工作原理图及相关开发过程图览



在上图中，开发者首先需要将工程项目中所有依赖的后端接口描述，按照指定的json格式，写入 interface.json 配置文件。必要时，需要对每个接口编写一个规则文件，也即图中 interface rules 部分。该规则文件用于在开发阶段 mock 数据或者在联调阶段使用 River 工具集去验证接口。规则文件的内容取决于采用哪一种 mock 引擎（比如 mockjs, river-mock 等等）。配置完成之后，即可在代码中按照自己的需求创建自己的业务 model。

下面是一个简单的例子：

### 【例一】

- 第一步 在工程目录中创建接口配置文件 interface.json, 并在其中添加主搜接口 json 定义

```

1 {
2   "title": "pad淘宝项目数据接口集合定义",
3   "version": "1.0.0",
4   "engine": "mockjs",
5   "rulebase": "./interfaceRules/",
6   "status": "online",
7   "interfaces": [ {
8     "name": "主搜索接口",
9     "id": "Search.getItems",
10    "urls": {
11      "online": "http://s.m.taobao.com/client/search.do"
12    }
13  } ]
14 }

```

- 第二步 在代码中创建并使用 model

```

1 // 引入模块
2 var ModelProxy = require( 'modelproxy' );
3
4 // 全局初始化引入接口配置文件 （注意：初始化工作有且只有一次）

```

```

5 ModelProxy.init( './interface.json' );
6
7 // 创建model 更多创建模式请参后文
8 var searchModel = new ModelProxy( {
9     searchItems: 'Search.getItems' // 自定义方法名: 配置文件中的定义的接口ID
10 } );
11
12 // 使用model, 注意: 调用方法所需要的参数即为实际接口所需要的参数。
13 searchModel.searchItems( { q: 'iphone6' } )
14     // !注意 必须调用 done 方法指定回调函数, 来取得上面异步调用searchItems获得的数据!
15     .done( function( data ) {
16         console.log( data );
17     } )
18     .error( function( err ) {
19         console.log( err );
20     } );

```

ModelProxy的功能丰富性在于它支持各种形式的profile以创建需要业务model:

- 使用接口ID创建>生成的对象会取ID最后'.'号后面的单词作为方法名

```
1 ModelProxy.create( 'Search.getItem' );
```

- 使用键值JSON对象>自定义方法名: 接口ID

```

1 ModelProxy.create( {
2     getName: 'Session.getUserName',
3     getMyCarts: 'Cart.getCarts'
4 } );

```

- 使用数组形式>取最后'.'号后面的单词作为方法名  
下例中生成的方法调用名依次为: Cart\_getItem, getItem, suggest, getName

```
1 ModelProxy.create( [ 'Cart.getItem', 'Search.getItem', 'Search.suggest', 'Session.User.g' ] );
```

- 前缀形式>所有满足前缀的接口ID会被引入对象, 并取其后半部分作为方法名

```
1 ModelProxy.create( 'Search.*' );
```

同时, 使用这些Model, 你可以很轻易地实现合并请求或者依赖请求, 并做相关模板渲染

### 【例二】合并请求

```

1 var model = new ModelProxy( 'Search.*' );
2
3 // 合并请求 (下面调用的model方法除done之外, 皆为配置接口id时指定)
4 model.suggest( { q: '女' } )
5     .list( { keyword: 'iphone6' } )
6     .getNav( { key: '流行服装' } )
7     .done( function( data1, data2, data3 ) {
8         // 参数顺序与方法调用顺序一致
9         console.log( data1, data2, data3 );
10    } );

```

### 【例三】依赖请求

```

1 var model = new ModelProxy( {
2   getUser: 'Session.getUser',
3   getMyOrderList: 'Order.getOrder'
4 } );
5 // 先获得用户id, 然后再根据id号获得订单列表
6 model.getUser( { sid: 'fdkaldjfgsakls0322yf8' } )
7   .done( function( data ) {
8     var uid = data.uid;
9     // 二次数据请求依赖第一次取得的id号
10    this.getMyOrderList( { id: uid } )
11      .done( function( data ) {
12        console.log( data );
13      } );
14  } );

```

此外ModelProxy不仅在Node端可以使用，也可以在浏览器端使用。只需要在页面中引入官方包提供的modelproxy-client.js即可。

【例四】浏览器端使用ModelProxy

```

1 <!-- 引入modelproxy模块，该模块本身是由KISSY封装的标准模块-->
2 <script src="modelproxy-client.js" ></script>

```

```

1 <script type="text/javascript">
2   KISSY.use( "modelproxy", function( S, ModelProxy ) {
3     // !配置基础路径，该路径与第二步中配置的拦截路径一致！
4     // 且全局配置有且只有一次！
5     ModelProxy.configBase( '/model/' );
6
7     // 创建model
8     var searchModel = ModelProxy.create( 'Search.*' );
9     searchModel
10      .list( { q: 'ihpone6' } )
11      .list( { q: '冲锋衣' } )
12      .suggest( { q: 'i' } )
13      .getNav( { q: '滑板' } )
14      .done( function( data1, data2, data3, data4 ) {
15        console.log( {
16          "list_ihpone6": data1,
17          "list_冲锋衣": data2,
18          "suggest_i": data3,
19          "getNav_滑板": data4
20        } );
21      } );
22  } );
23 </script>

```

同时，ModelProxy可以配合Midway另一核心组件[Midway-XTPL](https://github.com/purejs/midway-xtpl)一起使用，实现数据和模板以及相关渲染过程在浏览器端和服务端的全共享。关于ModelProxy的详细教程及文档请移步<https://github.com/purejs/modelproxy>

## 总结

ModelProxy以一种配置化的轻量级框架存在，提供友好的接口model组装及使用方式，同时很好的解决前后端开发模式分离中的接口使用规范问题。在整个项目开发过程中，接口始终只需要定义描述一次，前端开发人员即可引用，同时使用River工具自动生成文档，形成与后端开发人员的契约，并做相关自动化测试，极大地优化了整个软件工程开发过程。

【注】River 是阿里集团研发的前后端统一接口规范及相关工具集合的统称

【附】相关文章列表



1. [《前后端分离的思考与实践（一）》](#)
2. [《前后端分离的思考与实践（二）》](#)

 赞 5 收藏 评论

## 相关文章

- [基于 JavaScript 的操作系统你听说过吗？](#)
- [前后端分离的思考与实践（二）](#)
- [Web 开发在 2015 年及未来的发展趋势](#)
- [前后端分离的思考与实践（一）](#)
- [我为什么向后端工程师推荐Node.js](#)
- [为重负网络优化 Nginx 和 Node.js](#)

## 可能感兴趣的话题

- [女生搞开发能干一辈子吗？有没有发展前景啊？](#) · [Q 44](#)
- [中秋节不发福利也就算了，晚上加班测试，中秋节当天也要来加班，我也是醉了...](#) · [Q 56](#)
- [预算 6K 以内，求推荐开发用笔记本电脑](#) · [Q 73](#)
- [昨晚加班到4点多，4.30左右才到家，好久没发帖了，来发个帖聊聊](#) · [Q 51](#)
- [后端转前端，转型期真心痛苦](#) · [Q 12](#)
- [2017完美世界研发部笔试题\(取经\)](#) · [Q 12](#)

登录后评论

新用户注册

直接登录



文章

输入搜索关键字

搜索



- [本周热门文章](#)
- [本月热门文章](#)
- [热门标签](#)

## 0 [无休止加班的成因](#)

- 1 [我见过最有趣的代码注释，都在这里了...](#)
- 2 [HTTPS 科普扫盲帖](#)
- 3 [PM 叫你去改一个 Bug，后来.....](#)

- 4 [在谷歌招工程师，我看重的是这些](#)
- 5 [HTTPS 工作原理和 TCP 握...](#)
- 6 [TCP/IP之TCP协议：流量控制...](#)
- 7 [LOL设计模式之「策略模式」](#)
- 8 [.Net 高效开发之不可错过的实用工具](#)
- 9 [跟我一起写shell补全脚本（Zsh篇）](#)



业界热点资讯

[更多 »](#)



[甲骨文准备将NetBeans交给Apache管理](#)

11 小时前 · 2



[让计算机崩溃的简短代码](#)

3 天前 · 20



[PHP 7.1 新特性一览](#)

14 小时前 · 3



[redis3.2新功能 - GEO地理位置命令介绍](#)

1 天前 · 4



[JS 又是第一编程语言：GitHub 2016 年度报告](#)

3 天前 · 33 · 3

精选工具资源

[更多资源 »](#)



[android-remote-notifications: 从远程JSON文件拉取...](#)

## [Android](#), [通知](#)



## [PHP CPP](#): 一个开发PHP扩展的C++库 [PHP](#), [扩展](#)



## [Apache Mahout](#): 经典机器学习算法库 [Java](#), [机器学习](#)










## [WilliamChart](#): 优美和直观的图表库 [Android](#), [图表](#)



## [Singularity](#): 方便部署和操作的Mesos框架 [Java](#), [集群管理](#)

### 最新评论

-  Re: [我见过最有趣的代码注释，都在这...](#)  
12不是很信
-  Re: [PM 叫你去改一个 Bug, 后来...](#)  
现在初三，因为这个原因，理想就是不受雇于人，有一家自己的（小）公司。这种活法太恶心
-  Re: [你应该知道的计算机网络知识](#)  
同意, 因为一般一台机子只配有一个网络适配器(网卡), 所以也就默认一个MAC地址对应一台唯一的电脑...
-  Re: [133 行代码实现质感地形](#)  
应该感谢博主搜集了这么好的资源!! 辛苦啦--
-  Re: [133 行代码实现质感地形](#)  
谢谢提醒。已经在正文更新了 :)
-  Re: [PM 叫你去改一个 Bug, 后来...](#)  
很赞的文章，翻译得也非常赞。解决客户的问题，这个是最大的价值，当然也不必要放弃原则：  
P
-  Re: [133 行代码实现质感地形](#)  
你的连接错啦!! 现在修改了地址的 <http://demos.playfuljs.com/terrai...>





Re: [加入伯乐在线专栏作者，扩大知名...](#)  
MD 的教程，网络上有的，你自己搜搜。

## 关于伯乐在线博客

在这个信息爆炸的时代，人们已然被大量、快速并且简短的信息所包围。然而，我们相信：过多“快餐”式的阅读只会令人“虚胖”，缺乏实质的内涵。伯乐在线内容团队正试图以我们微薄的力量，把优秀的原创文章和译文分享给读者，为“快餐”添加一些“营养”元素。

### 快速链接

[网站使用指南»](#)  
[问题反馈与求助»](#)  
[加入我们»](#)  
[网站积分规则»](#)  
[网站声望规则»](#)

## 关注我们

新浪微博：[@伯乐在线官方微博](#)

RSS：[订阅地址](#)

推荐微信号



程序员的那些事



UI设计达人



极客范

### 合作联系

Email: [bd@jobbole.com](mailto:bd@jobbole.com)

QQ: 2302462408 (加好友请注明来意)

## 更多频道

[小组](#) - 好的话题、有启发的回复、值得信赖的圈子  
[头条](#) - 分享和发现有价值的内容与观点  
[相亲](#) - 为IT单身男女服务的征婚传播平台  
[资源](#) - 优秀的工具资源导航  
[翻译](#) - 翻译传播优秀的外文文章  
[文章](#) - 国内外的精选文章  
[设计](#) - UI, 网页, 交互和用户体验  
[iOS](#) - 专注iOS技术分享  
[安卓](#) - 专注Android技术分享  
[前端](#) - JavaScript, HTML5, CSS  
[Java](#) - 专注Java技术分享  
[Python](#) - 专注Python技术分享

© 2016 伯乐在线

[文章](#) [小组](#) [相亲](#) [加入我们](#) [反馈](#)

