

在同事的安利下，试用了下接口管理平台YApi

这篇文章发布于 2018年05月9日，星期三，02:26，归类于 [JS实例](#)。阅读 18738 次, 今日 13 次 [12 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7564>

本文可全文转载，但需要保留原作者和出处。

业界接口管理平台有好些个，例如[easyapi](#)，[sosoapi](#)，还有本文要介绍的YApi。



站在团队角度讲，这些接口平台最大作用是团队内部接入，规范API，管理，合作等都会变得更加轻松。

站在个人角度讲，尤其对于需要写高保真页面的前端重构人员，这些接口平台可以模拟返回更加真实的数据，天然跨域支持，自己无需额外搭建服务，就能有动态特性展示。

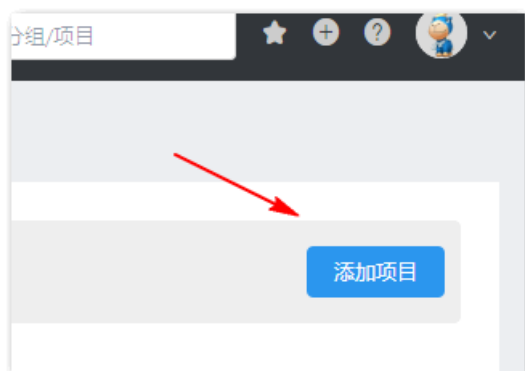
自己经常要写高保真的交互原型，因此，需要可以有相当真实的API请求数据，例如出错，请求延时模拟，极端数据处理等。同事的安利下，就开始接触YApi，试用了下，总体还行，有一些小坑，有一些局限，这里自己记录下，以后用得到。

一、接口管理平台YApi简介

这里的介绍主要站在个人使用角度而言的。

首先[注册](#)，无需邮箱验证就可以用。

进入后“新建项目”：



随便填一填，选一选，权限可以选择公开：

* 项目名称: lulu ui

* 所属分组: 个人空间

基本路径 ②: /lulu

描述:

* 权限: ☐ 私有
只有组长和项目开发者可以索引并查看项目信息

☒ 公开
任何人都可以索引并查看项目信息

+ 创建项目

张 <http://www.zhangxinxu.com>
鑫旭-鑫空间-鑫生活

然后就可以添加接口了:

这里可以随意点, 反正后期可以修改的, 例如:

确定后就进入下图所示的界面, 其中:

- “预览”作用就本身就是用来复制YApi提供的API请求地址;
- “编辑”就是编辑请求的类型、参数和返回的数据等, 如果模拟的数据偏静态, 则这个选项卡是主战场;
- “运行”要装Chrome插件, 还要翻wall, 可以忽略;
- “高级mock”是用来设定删选条件, 以及自己写JS代码根据不同参数返回不同数据的;

下面我们通过几个案例简单了解下“编辑”和“高级mock”。

二、了解“编辑”和“高级mock”

1. 无逻辑无筛选下的“编辑”

“编辑”适合静态的数据模拟, 能满足大多数的API请求需求。

“基本设置”中我们可以选择是GET请求还是POST请求:

“请求参数设置”可以设置ajax请求的查询参数, 例如, 我们写一个 `key`, 并且是必需:

“返回数据设置”这个地方有个 `json-schema`：

在我看来，应该是使用可视化的方式设计请求返回的数据。

我个人还是喜欢直接看JSON代码，因此置灰：

其中，这里的JSON代码并不是纯静止的JSON数据，支持一些特定的语法规则，可以多数据选一，随机数组个数等等，这里有张官方的图：

其中很多语法规则来自Mock.js，具体可[参见这里](#)。

Mock.js 的语法规则包括两部分：

1. 数据模板定义规范（Data Template Definition，DTD）
2. 数据占位符定义规范（Data Placeholder Definition，DPD）

前者可以实现数值递增，数值范围，自动计数，随机数量生成，正则字符生成等。后者则可以理解具有特定身份的占位符，例如 `@id` 会生成随机的 `id`，你可以理解是 `id` 的占位符，占位符关键字又很多，例如 `@email` 表示邮箱，`@name` 表示姓名等。差不多下图出现的这些单词都是占位符。

实例展示

例如，有下面的mock JSON数据：

```
{
  "error": 0,
  "data|0-3": [{
    "id": "@id",
    "value": "@name "
  },
  {
    "id": "@id",
    "value": " @name @word"
  },
  {
    "id": 5,
    "value": "<img src=\"icon_del.png\" align=absmiddle>支持小图标"
  },
  {
    "id": 8,
    "value": "<a href=\"?key=\">点击跳转</a>"
  }
]
```

表示 `"data"` 后面的数组里面数据随机循环0~3遍，注意，这里是循环0~3遍，不是 `"data"` 后面的数组随机选择0~3项。

因此，这个模板返回的 `"data"` 后面的数组数据可能是0个，也有可能是12个。

例如，某一次请求的返回数据：

```
{
  "error": 0,
  "data": [
    {
      "id": "630000199003261731",
      "value": "Christopher Hall "
    },
    {
      "id": "220000201107085094",
      "value": " Donald Thomas smse"
    },
    {
      "id": 5,
      "value": "<img src=\"icon_del.png\" align=absmiddle>支持小图标"
    },
    {
      "id": 8,
      "value": "<a href=\"?key=\">点击跳转</a>"
    }
  ]
}
```

模板中 `"data"` 数组循环了一次，数组中前两项的 `id` 随机和 `value` 随机，因为使用了 `@id` 占位符和 `@name` `@word` 占位符。

眼见为实，您可以狠狠地点击这里：<https://l-ui.com/content/apis/datalist.html>

其中“基于ajax动态返回数据的autocomplete功能”的这个模拟Autocomplete功能所发出去请求就是用的上面的模板，每输入一个内容，下拉列表的个数可能0个，可能4个，也可能8个……

2. 可以返回查询数据或请求数据

有时候我们希望API返回的数据就是我们提交的数据，或者包含提交的数据，YApi也是支持的。

```
{
  // 请求的url是/path?name=xiaoming，返回的name字段是xiaoming
  "name": "${query.name}",
  // 请求的requestBody type=1,返回的type字段是1
  "type": "${body.type}"
}
```

其中，`${query.name}` 你可以看成是GET请求的 `name` 参数值，`${body.type}` 你可以看成是POST请求提交的 `type` 参数数据。

实例展示

假设我们再新增一个接口，名为“lulu表单提交”，选择POST请求，同时设置 `tel` 和 `code` 两个必需字段，表示手机号和验证码：

同时模板数据如下：

```
{
  "error|0-1": 0,
  "msg": "一定概率出现的错误",
  "data": {
    "tel": "${body.tel}",
    "code": "${body.code}"
  }
}
```

则 `"error"` 参数50%概率返回值是1，也就是认为出错；同时，原封不动返回提交的电话号码和验证码。

眼见为实，您可以狠狠地点击这里：<https://l-ui.com/content/example/form.html>

上面的模板就是给这个文档页面的表单用的，点击提交按钮，50%概率出现下图所示错误提示：



50%概率则是成功提示，并且提示语中包含提交的手机号，这个手机号就是YApi返回的。



3. 具有筛选功能甚至复杂逻辑的“高级mock”

“高级mock”有两个选项卡，一个是“期望”，一个是“脚本”。

其中，“期望”意思是，当你输请求参数是这个的时候，我给你返回一个特殊的数据。例如，手机号11234567890是管理员账号，管理员返回数据要和普通成员不一样，这时候我们就可以使用期望来处理。

“脚本”则是可以自己写JavaScript逻辑，自己创造数据，想返回什么就是什么，可以说是数据伪造的终极解决方案。

实例展示

假设我们再新增一个接口，名为“lulu富交互列表”，选择GET请求，直接进行“高级mock”处理。

这个实例要先看页面，具有复杂逻辑，真分页lulu富交互列表：<https://l-ui.com/content/example/table.html>

这个页面点击分页数据都是真的分页，搜索，批量删除都有功能，甚至还有彩蛋，全部都是“高级mock”搞定的，“期望”和“脚本”都使用了，我们一个一个看下。

首先，输入框搜索 `error` 会演示出错效果：



相关的“期望”设置如下：

然后，回到初始态，我们选择任意列表，然后批量删除，则列表会为空：



相关的“期望”设置如下，设置期望条件，当 `action` 为 `'delete'`，返回空数据：



最后，看看一开始的真分页，和搜索随机列表数量效果是怎样处理的，就像平常写JavaScript语言一样，

在“脚本”这个输入框中写我们的逻辑即可，只要保证最后mockJSON数据是我们希望返回的数据即可！

完整代码如下：

```
var data = [{
    "id": 1,
    "title": "浅议星星的花朵",
    "time": "2015-07-20 16:23",
    "comment": 7,
    "deleted": false
}, /* ... 这里省略中间的47项 ... */ {
    "id": 48,
    "title": "产品细节中的情感化设计",
    "time": "2015-06-01 21:14",
    "comment": 0,
    "deleted": false
}];

var current = params.current;
var every = params.every;

var begin = 0;
var end = 0;

var total = data.length;

if (!params.key) {
    // 正常分页
    begin = (current - 1) * every;
    end = every;
} else {
    begin = Math.round(25 * Math.random());
    end = Math.round(20 * Math.random());

    total = end;
}

var dataGet = data.slice(begin, begin + end);

mockJson = {
    error: 0,
    data: {
        total: total,
        data: dataGet
    }
}
```

截图：



也就是默认写死差不多50条列表数据（可以通过“编辑模板”和占位符迅速生成），然后根据请求的参数进行逻辑处理。

其中 `params.xxxx` 可以获得无论是POST还是GET请求发送的 `xxxx` 参数的值，于是，我们可以从容地书写逻辑。

例如，本例子中，如果有 `params.key` 这个参数值，认为是搜索，则返回随机个数的列表；如果没有 `p`

arams.key，则认为是初始载入，则根据当前的分页页面，和用户选择的每页列表条数准确返回列表数目，于是一个高保真的富交互列表效果就出来了，各种体验和交互问题就能很好地测试了。

4. 三者的优先级关系

请求我们模拟数据时，规则匹配优先级：期望 > 自定义脚本 > 普通模板。

如果前面匹配到数据，后面的则不返回数据。

三、YApi的局限

不支持https请求，如果大家原型页面是https协议的，则YApi不能直接使用，可以搞个https页面反向代理下。或者使用EasyApi，按照同时说法，这个是支持https，不过我个人并没有测试。

四、结束语

本文洋洋洒洒这么长，主要展示个体在开发高保真HTML页面时候，如何借助API接口平台工具，让我们的页面的交互体验做到尽善尽美，更健壮，让设计师和开发人见人爱。

但个体涉及的知识点只是YApi中较小的一部分，更多内容，尤其团队内网部署等还需要多多参考API文档：<https://yapi.ymfe.org/index.html>

一点小分享，自己备忘同时希望可以帮到对API平台有兴趣的小伙伴。

最后，感谢去哪儿网移动大前端技术中心对本工具的开源贡献。

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击[这里](#)。有话要说？点击[这里](#)。



« windows系统下批量删除OS X系统.DS_Store文件

小tips: 纯CSS实现蜡烛、火焰以及熄灭后烟雾效果 »

猜你喜欢

- jQuery powerSwitch万能slide(切换)插件
- 翻译 - CSS Sprites:实用技术还是生厌之物?
- 翻译 - 高性能网站需避免的7个错误
- 小tip: base64:URL背景图片与web页面性能优化
- 梳理: 提高前端性能方面的处理以及不足
- 从天猫某活动视频不必要的3次请求说起
- jQuery boxy弹出层对话框插件中文演示及讲解
- jQuery图片文本滚动切换插件jCarousel中文翻译与详解
- 自己写的无图片版jQuery zxxbox弹出框插件
- 网页制作辅助工具-jQuery标尺参考线插件
- Snap.svg-SVG实战学习必修课-实例与文档讲解

分享到: 1

标签: API, HTTP请求, mock, mock.js, yapi, 动态api

发表评论（目前12条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

- youtugif**说道:
2018年10月14日 16:05

easyapi这个太菜了，随随便便点几下就出现403.jsp

[回复](#)
- 上天**说道:
2018年09月10日 10:10

这玩意不支持同post路径不同参数,返回不同内容的mock....没法搞了....要搞的话需要建好多个项目...我还是用moco.jar自己玩吧...

[回复](#)
- panhezeng**说道:
2018年08月20日 18:32

和easy mock比起来太弱了
<https://easy-mock.com/project/5ad70e6aa675954fc238c322>
这个比yapi的期望和高级脚本牛多了好不好
而且easy-mock编辑器是支持mock语法提示和自动补齐的

[回复](#)
- leo**说道:
2018年06月20日 15:26

使用eoLinker的飘过

[回复](#)
- trance**说道:
2018年05月20日 20:01

试试easy-mock https 的前端模拟数据好帮手



回复

狐狸说道:

2018年06月5日 17:03

easy-mock 没有“期望”和“自定义脚本”功能吧



回复

panhezeng说道:

2018年08月20日 18:36

easy mock的自定义响应了解一下.<https://easy-mock.com/docs#zi-ding-yi-xiang-ying>

yapi的自定义脚本和期望,竟然无法获得路由参数,也是醉了.只有query和body,而且自定义脚本竟然不支持mock模板,无语,

而且编辑json不能格式化,不能提示补全



回复

6. Jelton说道:

2018年05月19日 09:10

我们公司现在也用这个...



回复

7. 不得不错说道:

2018年05月17日 09:39

张大使用过easy-mock吗, 与这个挺像, 可以使用yaml定义接口。



回复

8. 广州网站建设说道:

2018年05月15日 22:28

与时俱进, 学习中



回复

9. 农业微信群孵化说道:

2018年05月14日 17:20

我也来跟着大站长学学前端技术



回复

10. 占位置说道:

2018年05月11日 17:10

战墙



回复

最新文章

- » 常见的CSS图形绘制合集
- » 粉丝群第1期CSS小测点评与答疑
- » 分享三个纯CSS实现26个英文字母的案例
- » 小tips: 纯CSS实现打字动画效果
- » CSS/CSS3 box-decoration-break属性简介
- » CSS :placeholder-shown伪类实现Material Design占位符交互效果
- » 从天猫某活动视频不必要的3次请求说起

- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的?](#)
- » [周知: CSS -webkit-伪元素选择器不再导致整行无效](#)

今日热门

- » [常见的CSS图形绘制合集](#) (179)
- » [粉丝群第1期CSS小测点评与答疑](#) (113)
- » [未来必热: SVG Sprite技术介绍](#) (111)
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) (85)
- » [让所有浏览器支持HTML5 video视频标签](#) (83)
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) (80)
- » [CSS3下的147个颜色名称及对应颜色值](#) (78)
- » [小tips: 纯CSS实现打字动画效果](#) (73)
- » [写给自己看的display: flex布局教程](#) (70)
- » [分享三个纯CSS实现26个英文字母的案例](#) (70)

今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) (76)
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) (64)
- » [看, for..in和for..of在那里吵架!](#) (60)
- » [是时候好好安利下LuLu UI框架了!](#) (47)
- » [原来浏览器原生支持JS Base64编码解码](#) (35)
- » [妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现](#) (33)
- » [炫酷H5中序列图片视频化播放的高性能实现](#) (31)
- » [CSS scroll-behavior和JS scrollToView让页面滚动平滑](#) (30)
- » [windows系统下批量删除OS X系统.DS_Store文件](#) (26)
- » [写给自己看的display: flex布局教程](#) (26)

猜你喜欢

- [jQuery powerSwitch万能slide\(切换\)插件](#)
- [翻译 - CSS Sprites:实用技术还是生厌之物?](#)
- [翻译 - 高性能网站需避免的7个错误](#)
- [小tip: base64:URL背景图片与web页面性能优化](#)
- [梳理: 提高前端性能方面的处理以及不足](#)
- [从天猫某活动视频不必要的3次请求说起](#)
- [jQuery boxy弹出层对话框插件中文演示及讲解](#)
- [jQuery图片文本滚动切换插件jCarousel中文翻译与详解](#)
- [自己写的无图片版jQuery zxxbox弹出框插件](#)
- [网页制作辅助工具-jQuery标尺参考线插件](#)
- [Snap.svg-SVG实战学习必修课-实例与文档讲解](#)