

API 管理平台操作手册

1 引言

1.1 编写目的

方便前端和后端同学更好的使用 API 管理平台。

1.2 项目背景

- 1、前后端同学合作时，定义接口不能规范化，增加了大量沟通时间。
- 2、前端同学调试代码时，必须依赖后端系统返回测试数据，大大降低开发效率。
- 3、测试同学在测试过程中产生 bug，不能确认是由于前端还是后端代码引发的，提 bug 时需要转发再转发。

1.3 参考资料

mock: <http://mockjs.com/>

2 软件概述

2.1 目标

能让前后端的小伙伴们能更愉快的一起写代码。

2.2 功能

- 1、后端同学编写接口文档，详细说明接口的参数信息。
- 2、前端同学查看接口文档，并通过平台自动模拟生成后端返回的测试数据。
- 3、校验前端同学请求的参数是否正确。
- 4、校验后端同学返回的结果是否正确。

3 运行环境

3.1 硬件

- 不详

3.2 软件

- linux
- nodejs
- mongodb
- mock.js

4 使用说明

4.1 用户管理

4.2 团队管理

4.3 项目管理

4.4 PRD 管理

4.5 API 管理

4.5.1 进入 API 管理平台

- 1、在我的 PRD 页面里，点击 **api**，

创建PRD

操作	版本号	主要功能	项目类型	阶段	提测时间	上线时间	产品	自测	JIRA地址	备注
api详情更新删除	1.2	交易进度子任务		已上线	2016-07-08	2016-07-08				

< 1 >

2、进入 api 管理平台页面。

全部团队 / 微信企业号 / sec_house_order / PRD: 1.2

当前项目URL: http://10.7.249.152:3888/fete_api/cnw8dt
当前PRD URL: http://10.7.249.152:3888/fete_api/cnw8dt/kwk7npL

新建API

描述	链接	方法
显示问题内容	/feedback/getContent	GET
获取过户付款方式	/weixinEnt/order/queryPayWay	GET
查询订单可新增的子任务	/weixinEnt/order/queryAvaliableAddSubTaskResult	GET
获取订单已有子任务列表	/weixinEnt/order/queryOrderTask	GET

4. 5. 2 新增 API 接口

1、点击 新建API 按钮，新增一条空的 API：

全部团队 / 微信企业号 / sec_house_order / PRD: 1.2

当前项目URL: http://10.7.249.152:3888/fete_api/cnw8dt
当前PRD URL: http://10.7.249.152:3888/fete_api/cnw8dt/kwk7npL

新建API

描述	链接	方法
-	/	-
显示问题内容	/feedback/getContent	GET
获取过户付款方式	/weixinEnt/order/queryPayWay	GET
查询订单可新增的子任务	/weixinEnt/order/queryAvaliableAddSubTaskResult	GET
获取订单已有子任务列表	/weixinEnt/order/queryOrderTask	GET

2、点击新增行，左侧弹出编辑页面编辑

全部团队 / 微信企业号 / sec_ho

当前项目URL: http://10.7.249.152:3

当前PRD URL: http://10.7.249.152:3

描述

-

显示问题内容

获取过户付款方式

查询订单可新增的子任务

获取订单已有子任务列表

新建API

标题

一句话描述

输入数据格式

1 {}

返回数据格式

1 {
2 {"status": 1,
3 "data": {
4 "test": ""
5 }
6 }

1 {}

mock->

属性	数据类型	含义	mock规则
status	Number		
- data	Object		
test	String		

修改说明

接口修改说明

接口修改日志

上一条

确定

取消

下一条

输入

- 标题：接口的中文说明
- URL：请求地址，以“/”开头，并且不包括工程名
- Method：接口请求方式：GET, POST, PUT, DELETE
- 输入数据格式：接口的请求参数，json 格式
- 返回数据格式：返回的数据格式，json 格式
- 表格：含义：属性说明。
- 表格：mock 规则：参考 <http://mock.js.com/>，输入 mock 规则。
- 写入方法：从“|”开始写到最后，

```
Mock.mock({
  "string|1-10": "*"
})
```

1、字符串: `}}` 写成: `| 1-10:" *`, 省略掉冒号左边的引号!

```
Mock.mock({
  "array|1-10": [
    {
      "name|+1": [
        "Hello",
        "Mock.js",
        "!"
      ]
    }
  ]
})
```

2、集合对象: `set` 写成: `data: {1-10: , (最后的“:”也要保留)`
`name: {+1: [“Hello”, “Mock.js”, “!”]}`

3、对象：Object 类型，只需要写“:”

```
Mock.mock({
  "array|1-10": [
    "Mock.js"
  ]
})
```

4、数组值：Array 类型：如 `1-10: ["mock.js"]`

Mock.mock({
 "boolean|1": true
})
5、Boolean: }}，写成：|1:true

➤ 修改说明：每次改动时，必须要填写修改说明。

🔍 查询订单可新增的子任务

标题

URL (首字符请输入)

method

查询订单可新增的子任务

/weixinEnt/order/queryAvailbleAddSubTaskResult

get

输入数据格式

1 {
2 " id": 1
3 }

返回数据格式

1 {
2 "status": 1,
3 "msg": "返回结果说明",
4 "type": [
5 1
6],
7 "data": [
8 {
9 "taskId": "1",
10 "taskName": "受理"
11 },
12 {
13 "taskId": "5",
14 "taskName": "查限购"
15 }
16],
17 "user": {
18 " id": "1",
19 "name": "jade"
20 }
21 }

mock->
1 {
2 "user": {
3 "name": "孟勇",
4 " id": "wbnjttjlloyjtnewdujvmswqkifgntxpy"
5 },
6 "data": [
7 {
8 "taskName": "出押",
9 "taskId": "4"
10 }
11],
12 "type": [
13 "lcz",
14 "ry",
15 "cy"
16],
17 "msg": "成功",
18 "status": 0
19 }

属性	数据类型	含义	mock规则
status 数字	Number	状态 (0: 成功, 1:失败)	1:[0,1]
msg 字符串	String	返回结果说明	1:['成功','失败']
type 数组值	Array	类型	1-2:['lcz','ry','cy']
- data 对象集合	Array	返回数据	1-3:
taskId	String	taskId	+1:['4','5','6']
taskName	String	task名称	1:['受理','查限购','出押']
- user 对象	Object	用户对象	:
id	String	用户ID	:@word(32)
name	String	用户名	:@cname

修改说明

接口修改说明

上一条

确定

删除

取消

下一条

点击 确定 按钮保存。

4. 5. 3 修改 API 接口

点击列表数据行，左侧弹出 API 详情，修改信息后，点击 确定 按钮保存。

4. 5. 4 删除 API 接口

点击列表数据行，左侧弹出 API 详情，点击 删除 删除。

4.5.5 查看提醒

点击右上角 **消息 (6)**，进入消息列表：

首页 团队 消息 (7) lancui

时间	用户名	平台	操作	描述	状态 全部已读
2016-07-12T10:38:44.089Z	123456	team	invited	123456邀请你加入"PC"	接受 拒绝
2016-07-12T09:31:24.688Z	jesse	api	add	add	未读
2016-07-12T07:48:35.473Z	123456	api	delete	删除API接口:ghjkgjh/asdsa/sdaf	未读
2016-07-12T07:47:49.789Z	123456	api	add	new	未读
2016-07-12T07:47:31.486Z	123456	api	add	new	未读
2016-07-12T07:45:42.929Z	123456	api	update	new	未读
2016-07-12T07:42:30.548Z	123456	api	add	new	未读
2016-07-12T06:55:52.574Z	葛袁军	team	invited	葛袁军邀请你加入"test12"	已读
2016-07-12T03:10:39.926Z	test	api	add	new	已读
2016-07-12T03:03:22.075Z	test	api	delete	删除API接口:123/asd/asdasd	已读
2016-07-12T03:02:57.500Z	jade1	team	invited	jade1邀请你加入"test12"	已读
2016-07-12T02:59:25.060Z	jade	team	invited	jade邀请你加入"全新团队"	已读

点击 接受、拒绝、未读 按钮，标记为已读。

4.6 校验请求参数和返回参数正确性。

1、用法（两种方法选一种）

```
// 1: gulp api --pjid=123456 --mock --task=vue --dev
// 2: 在 package.json 里的 script 里面新建项，把上面的命令当做值；然后使用 npm
start [name] 来启动
```

2、以下代码加入 gulpfile.js 中：

```
gulp.task('api', function() {
  // 必要参数的检查
  if (!argv.pjid) {
    console.error('必须加 --pjid=[对应的 projectId] ')
    return false
  }

  // 如果项目目录下没有 .tmp 文件夹，下面到代码会新建一个 .tmp 文件夹。
  // 注意：需要自己手动把它添加到 .gitignore 里面
  var fs = require('fs')
  var tmpDir = fs.statSync('./.tmp')
```

```

    if (!tmpDir.isDirectory()){
        fs.mkdirSync('./.tmp')
    } else {
        fse.emptydirSync('./.tmp')
    }

    // 发 http 请求去动态下载 mock_check.js 文件
    var http = require('http')
    var file = fs.createWriteStream('./tmp/mock_check.js', { flags: 'w' })
    var request = http.get('http://fe.superjia.com:8888/api/mock_check.js?useMockData=${argv.mock ? true
false}&projectId=${argv.pjid}', function(response) {
        var stream = response.pipe(file)
        stream.on('finish', function() {
            // js 文件下载保存完成之后，判断是否有 --task 参数，有到话启动相应的任务
            if (argv.task) {
                gulp.start(argv.task)
            }
        })
    })
})

// 这里是用 --dev 参数来决定是否把 mock_check.js 文件一起打包
// 这段代码需要在 webpack 使用 config 之前
if (argv.dev) {
    config.entry.common.push('./tmp/mock_check.js');
}

```

```

// 这里是用 --dev 参数来决定是否把 mock_check.js 文件一起打包
// 这段代码需要在 webpack 使用 config 之前
if (argv.dev) {
    config.entry.common.push('./tmp/mock_check.js');
}

webpack(config, function (err, stats) {
    console.log(stats.toString());
});

```

在这个点上面加那个 if 语句