API管理平台操作手册

1. **引言**
   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. 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. 参考资料

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

1. **软件概述**
   1. 目标

能让前后端的小伙伴们能更愉快的一起写代码。

* 1. 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。

1. 校验后端同学返回的结果是否正确。
2. **运行环境**
   1. 硬件

● 不详

* 1. 软件

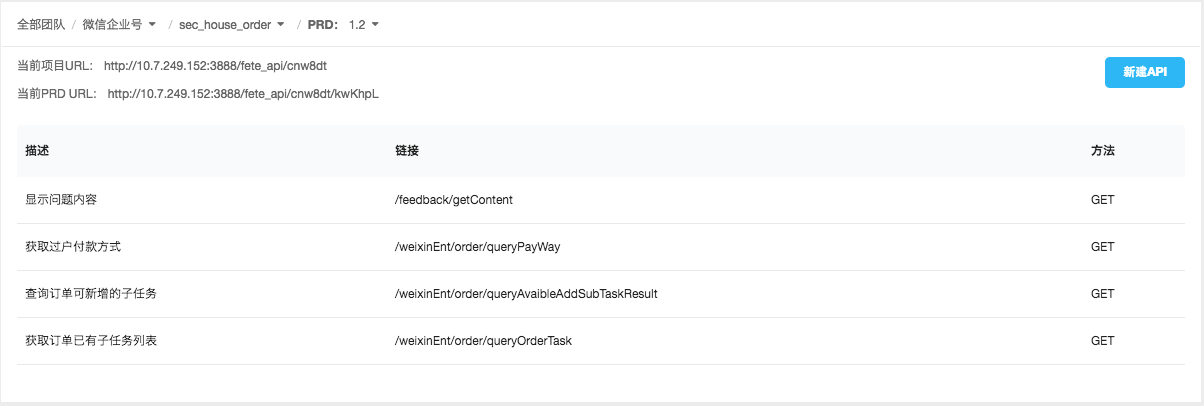
● linux   
● nodejs   
● mongodb   
● mock.js

1. **使用说明**
   1. 用户管理
   2. 团队管理
   3. 项目管理
   4. PRD管理
   5. API管理
      1. **进入API管理平台**

**1、在我的PRD页面里，点击****，**

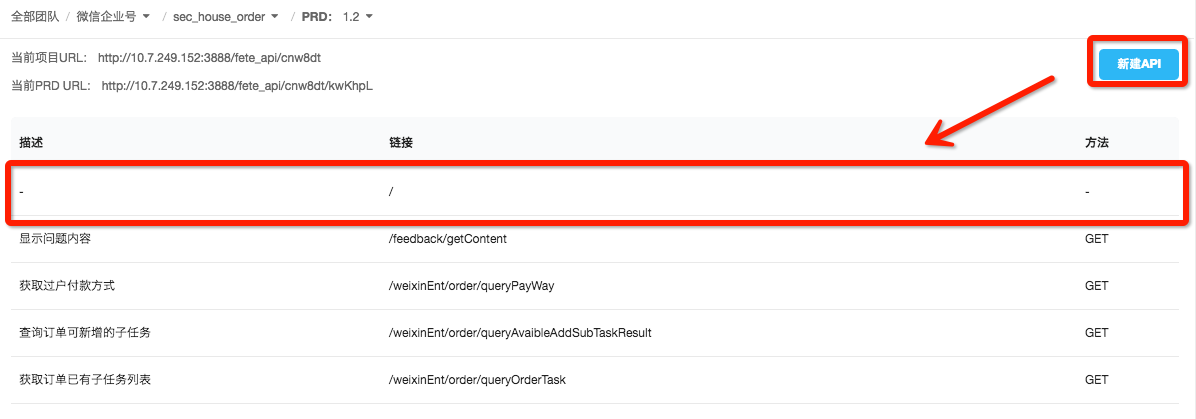


**2、进入api管理平台页面。**

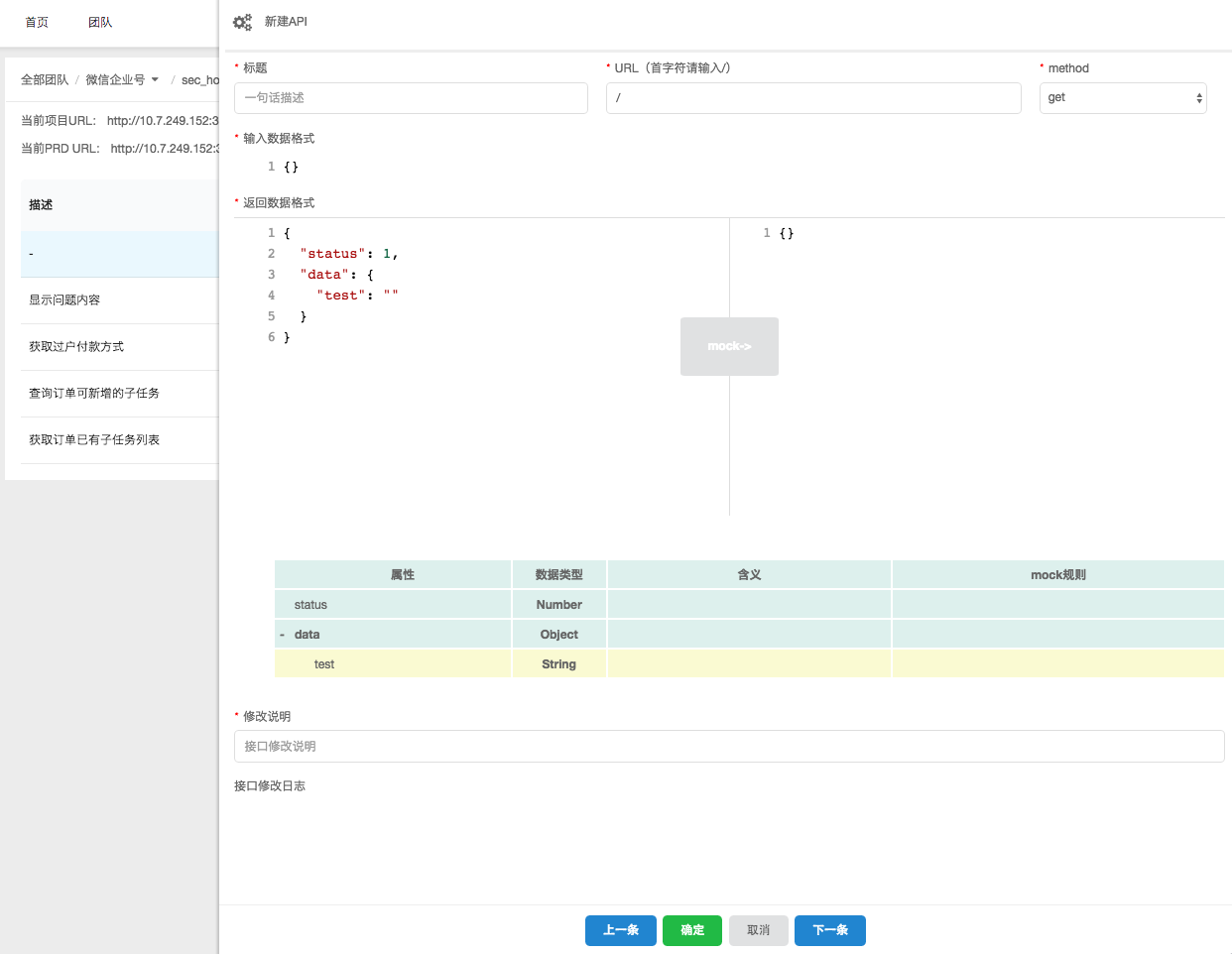
****

* + 1. **新增API接口**

**1、点击按钮，新增一条空的API：**

****

**2、点击新增行，左侧弹出编辑页面编辑**

****

**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://mockjs.com/>**,输入mock规则。**
* **写入方法：从“｜”开始写到最后，**

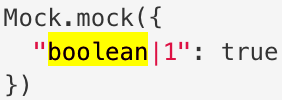
**1、字符串： 写成：｜1-10:”\*”,省略掉冒号左边的引号！**

**2、集合对象：写成：data: |1-10: ,(最后的“：”也要保留)**

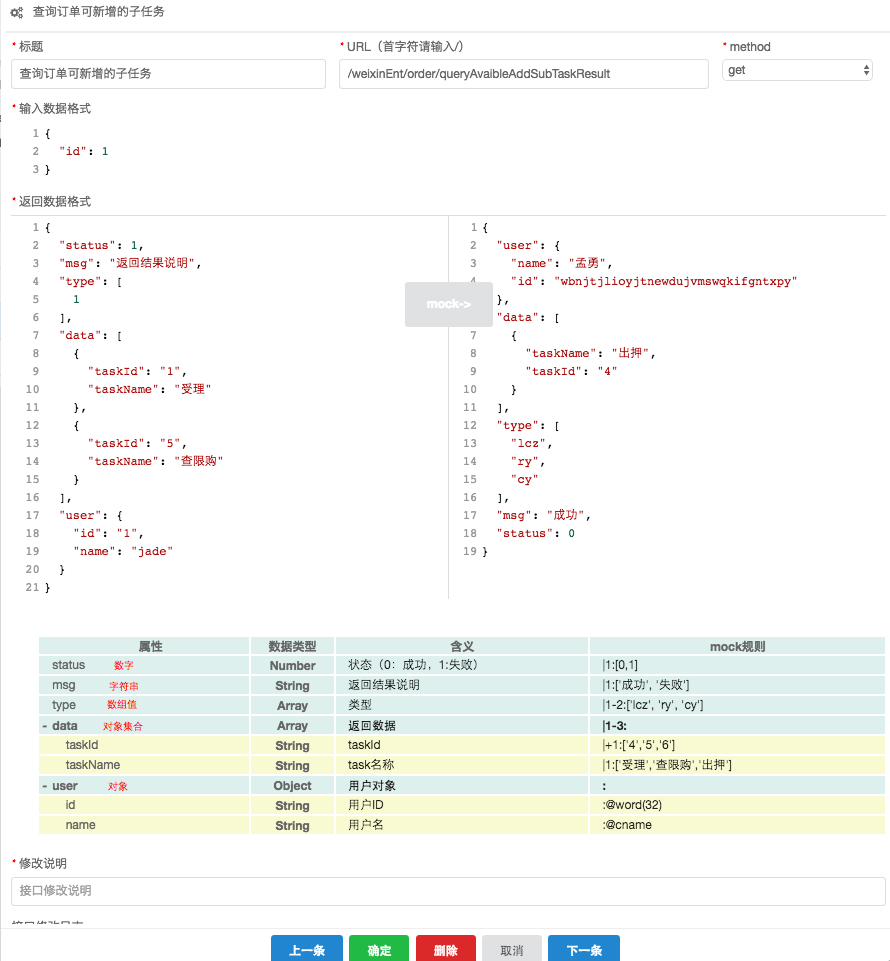
**name: |+1:[“Hello”,”Mock.js”,”!”]**

**3、对象：Object类型，只需要写“：”**

**4、数组值：Array类型：如，写成：｜1-10:[“mock.js”]**

**5、Boolean：，写成：|1:true**

* **修改说明：每次改动时，必须要填写修改说明。**

****

**点击按钮保存。**

* + 1. **修改API接口**

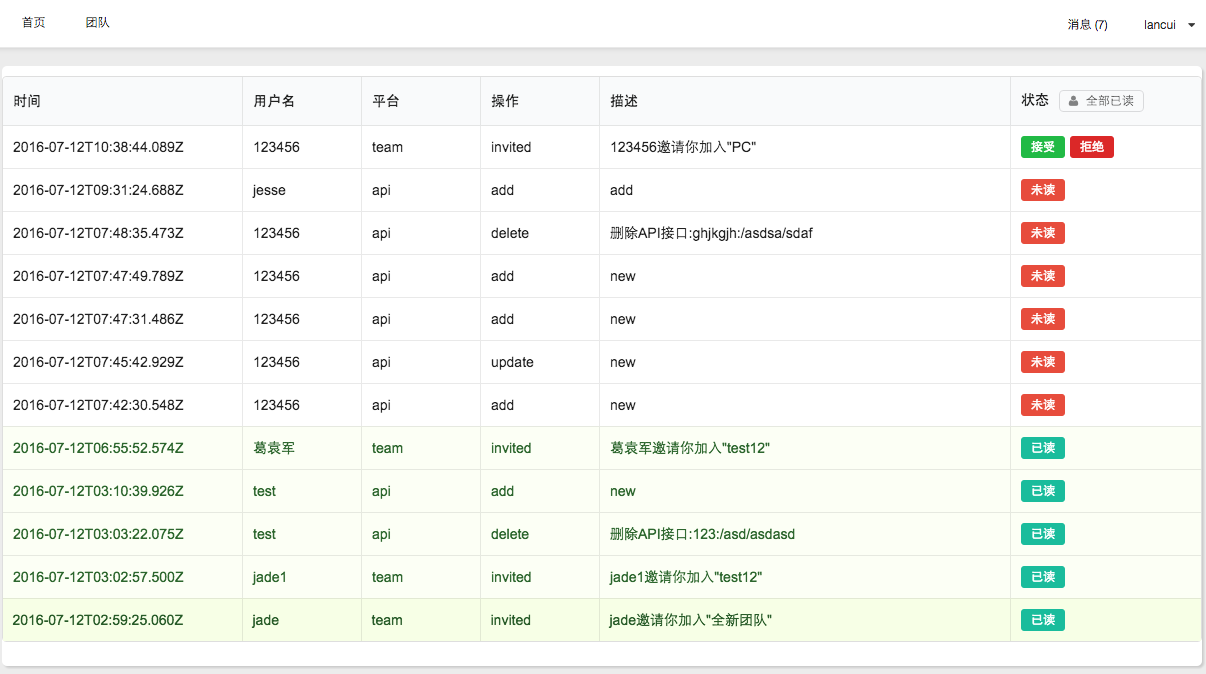
**点击列表数据行，左侧弹出API详情，修改信息后，点击按钮保存。**

* + 1. **删除API接口**

**点击列表数据行，左侧弹出API详情，点击删除。**

* + 1. 查看提醒

点击右上角，进入消息列表：



点击、按钮，标记为已读。

* 1. 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 |

1. 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');  } |

