DOClever使用手册

作者：神圣计划

移动时代首选接口管理平台

目　录

快速开始

DOClever-移动时代首选接口管理平台！

DOClever是一个可视化接免费开源的口管理工具,可以分析接口结构，校验接口正确性， 围绕接口定义文档，通过一系列自动化工具提升我们的协作效率。DOClever前后端全部采用了javascript来作为我们的开发语言，前端用的是vue+element UI，后端是express+mongodb，这样的框架集成了高并发，迭代快的特点，保证系统的稳定可靠。

主要特性：

• 可以对接口信息进行编辑管理，支持 get,post,put,delete,patch 五种方法，支持 https 和 https 协议，并且支持 query，body，json，raw，rest，formdata 的参数可视化编辑。同时对 json 可以进行无限层次可视化编辑。并且，状态码，代码注入，markdown 文档等附加功能应有尽有。

• 接口调试运行，可以对参数进行加密，从 md5 到 aes 一应俱全，返回参数与模型实时分析对比，给出不一致的地方，找出接口可能出现的问题。如果你不想手写文档，那么试试接口的数据生成功能，可以对接口运行的数据一键生成文档信息。

• mock 的无缝整合，DOClever 自己就是一个 mock 服务器，当你把接口的开发状态设置成已完成，本地 mock 便会自动请求真实接口数据，否则返回事先定义好的 mock 数据。

• 支持 postman，rap，swagger 的导入，方便你做无缝迁移，同时也支持 html 文件的导出，方便你离线浏览！

• 项目版本和接口快照功能并行，你可以为一个项目定义 1.0，1.1，1.2 版本，并且可以自由的在不同版本间切换回滚，再也不怕接口信息的遗失，同时接口也有快照功能，当你接口开发到一半或者接口需求变更的时候，可以随时查看之前编辑的接口信息。

• 自动化测试功能，目前市面上类似平台的接口自动化测试大部分都是伪自动化，对于一个复杂的场景，比如获取验证码，登陆，获取订单列表，获取某个特定订单详情这样一个上下文关联的一系列操作无能为力。而 DOClever 独创的自动化测试功能，只需要你编写极少量的 javascript 代码便可以在网页里完成这样一系列操作，同时，DOClever 还提供了后台定时批量执行测试用例并把结果发送到团队成员邮箱的功能，你可以及时获取接口的运行状态。

• 团队协作功能，很多类似的平台这样的功能是收费的，但是 DOClever 觉得好东西需要共享出来，你可

以新建一个团队，并且把团队内的成员都拉进来，给他们分组，给他们分配相关的项目以及权限，发布团队公告等等。

序言

DOClever-移动时代首选接口管理平台！

DOClever是一个可视化接免费开源的口管理工具,可以分析接口结构，校验接口正确性， 围绕接口定义文档，通过一系列自动化工具提升我们的协作效率。DOClever前后端全部采用了javascript来作为我们的开发语言，前端用的是vue+element UI，后端是express+mongodb，这样的框架集成了高并发，迭代快的特点，保证系统的稳定可靠。

主要特性：

• 可以对接口信息进行编辑管理，支持 get,post,put,delete,patch 五种方法，支持 https 和 https 协议，并且支持 query，body，json，raw，rest，formdata 的参数可视化编辑。同时对 json 可以进行无限层次可视化编辑。并且，状态码，代码注入，markdown 文档等附加功能应有尽有。

• 接口调试运行，可以对参数进行加密，从 md5 到 aes 一应俱全，返回参数与模型实时分析对比，给出不一致的地方，找出接口可能出现的问题。如果你不想手写文档，那么试试接口的数据生成功能，可以对接口运行的数据一键生成文档信息。

• mock 的无缝整合，DOClever 自己就是一个 mock 服务器，当你把接口的开发状态设置成已完成，本地 mock 便会自动请求真实接口数据，否则返回事先定义好的 mock 数据。

• 支持 postman，rap，swagger 的导入，方便你做无缝迁移，同时也支持 html 文件的导出，方便你离线浏览！

• 项目版本和接口快照功能并行，你可以为一个项目定义 1.0，1.1，1.2 版本，并且可以自由的在不同版本间切换回滚，再也不怕接口信息的遗失，同时接口也有快照功能，当你接口开发到一半或者接口需求变更的时候，可以随时查看之前编辑的接口信息。

• 自动化测试功能，目前市面上类似平台的接口自动化测试大部分都是伪自动化，对于一个复杂的场景，比如获取验证码，登陆，获取订单列表，获取某个特定订单详情这样一个上下文关联的一系列操作无能为力。而 DOClever 独创的自动化测试功能，只需要你编写极少量的 javascript 代码便可以在网页里完成这样一系列操作，同时，DOClever 还提供了后台定时批量执行测试用例并把结果发送到团队成员邮箱的功能，你可以及时获取接口的运行状态。

• 团队协作功能，很多类似的平台这样的功能是收费的，但是 DOClever 觉得好东西需要共享出来，你可

以新建一个团队，并且把团队内的成员都拉进来，给他们分组，给他们分配相关的项目以及权限，发布团队公告等等。

在线帮助：

产品官网：http://doclever.cn
视频帮助：http://doclever.cn/help/help.html
官网Q群：611940610

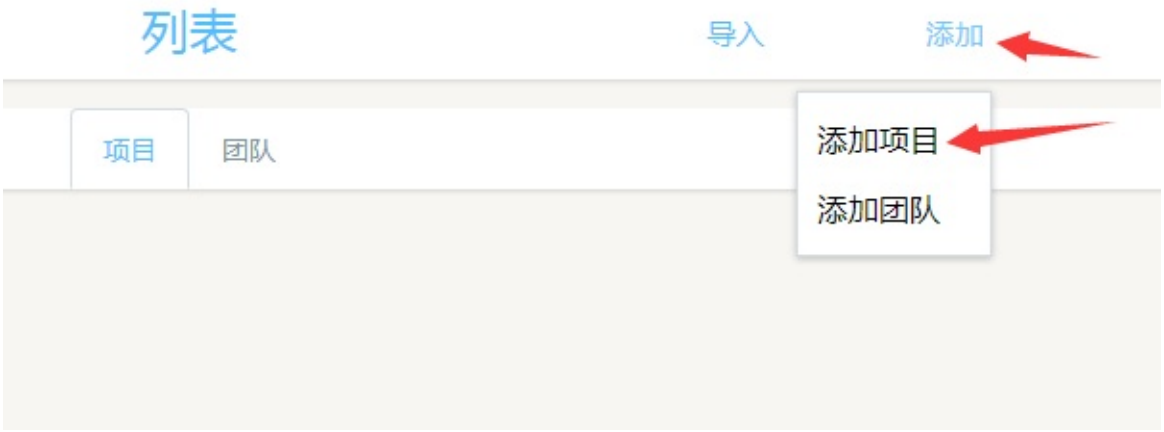
开源地址：

Github: https://github.com/sx1989827/DOClever
码云: https://git.oschina.net/sx1989827/SBDoc

快速创建项目

现在我们要开始创建第一个项目了，准备好了吗？

> 1、登录后台后点击右上角“添加-->添加项目”按钮



> 2、点击后会弹出如下界面：

> 名称：该项目的名称
> 描述：该项目的说明，如该项目的简单描述

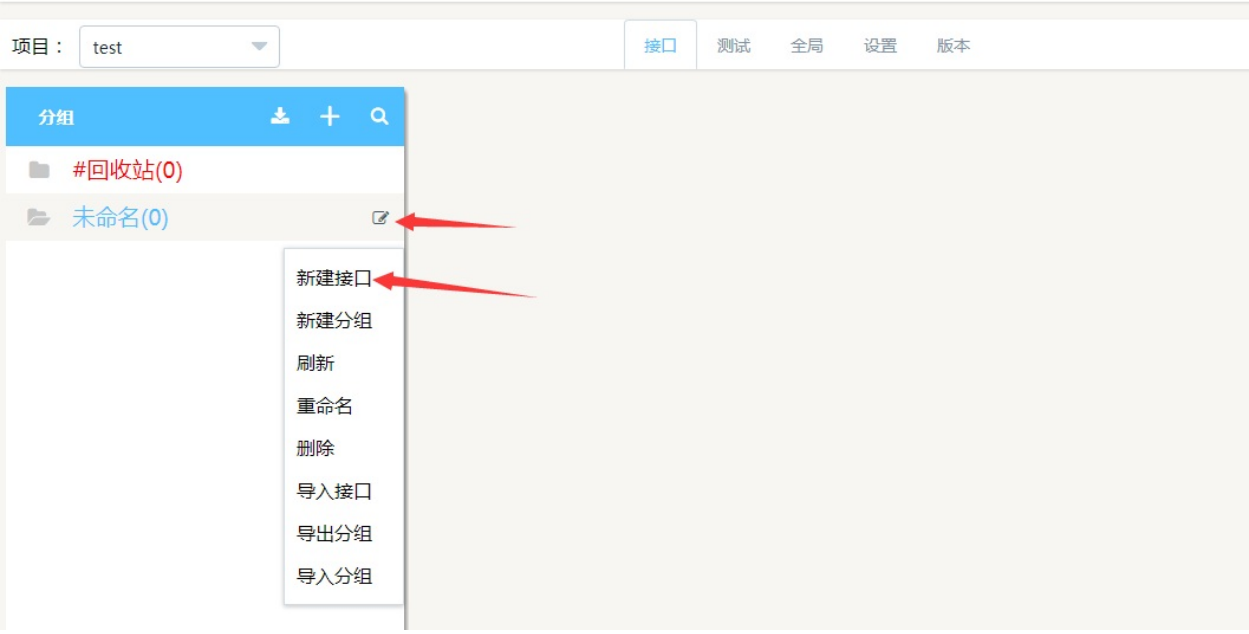
> 至此，您已经知道怎么创建一个项目了吧？

快速创建接口

上个章节，我们已经学会该如何简单的创建一个项目了；
这节来带大家快速创建一个接口！

> 创建完项目后，会出现，我创建的：

> 点击项目后会出现如下界面：

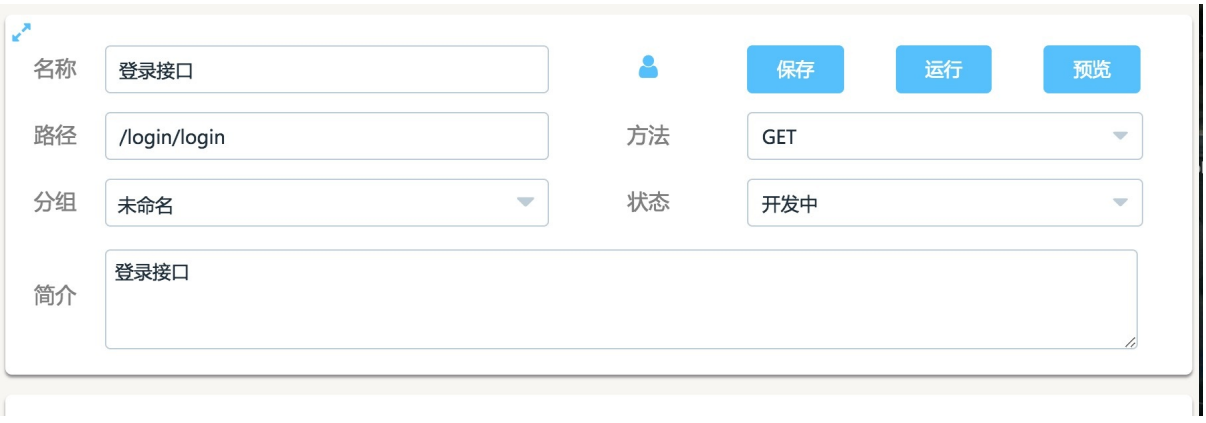


> DOClever默认会创建两个分组：回收站 和 未命名
> 我们点击未命名分组，会弹出一个下拉框，我们点击新建接口；

> 点击新建接口后，右边会出现如下界面

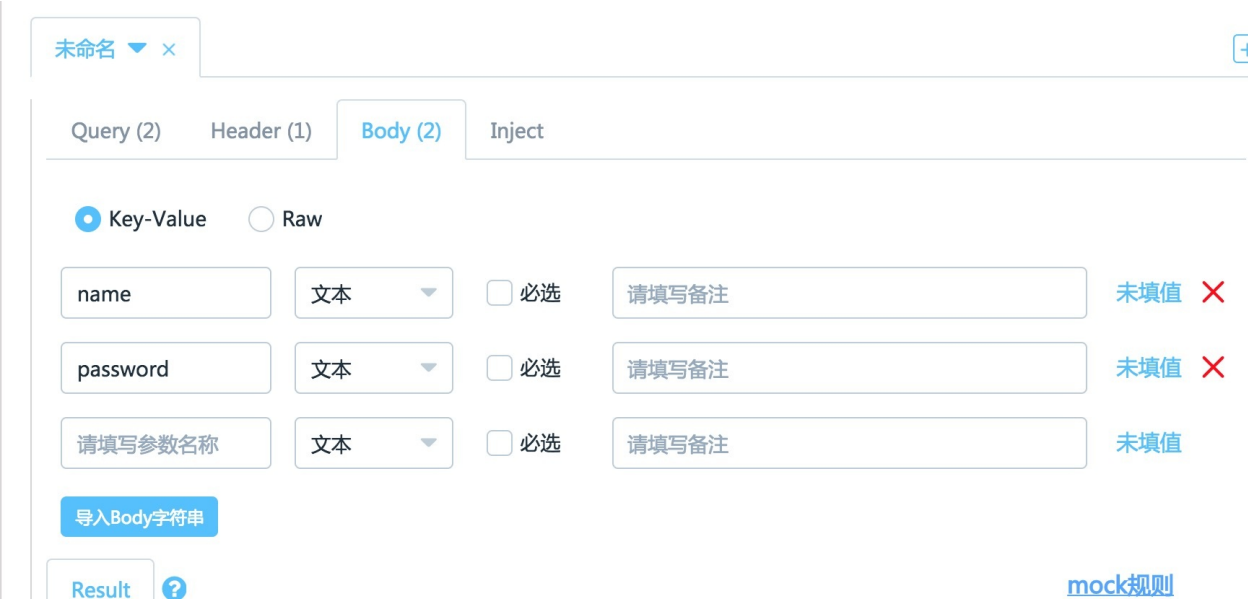
> 我们根据上图图示，填写您的接口名称等信息，就可以快速创建一个接口了,举个例子 创建登录接口

> 1.填写接口的信息:名称，路径，接口方法，分组，状态以及简介

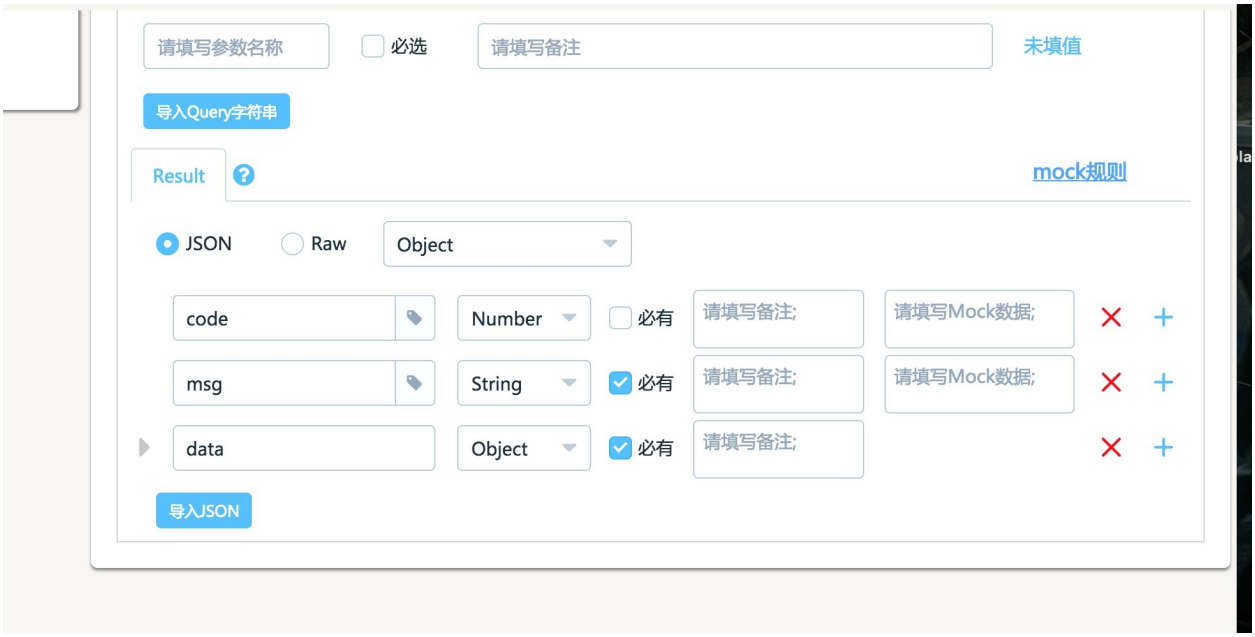


> 2.接口的方法是GET 入参的填写

3.接口的方法是POST 入参的填写



4.接下来是出参



接下来就可以愉快的使用doclever接口管理平台了

详细使用

基础

注册登录

第一步：

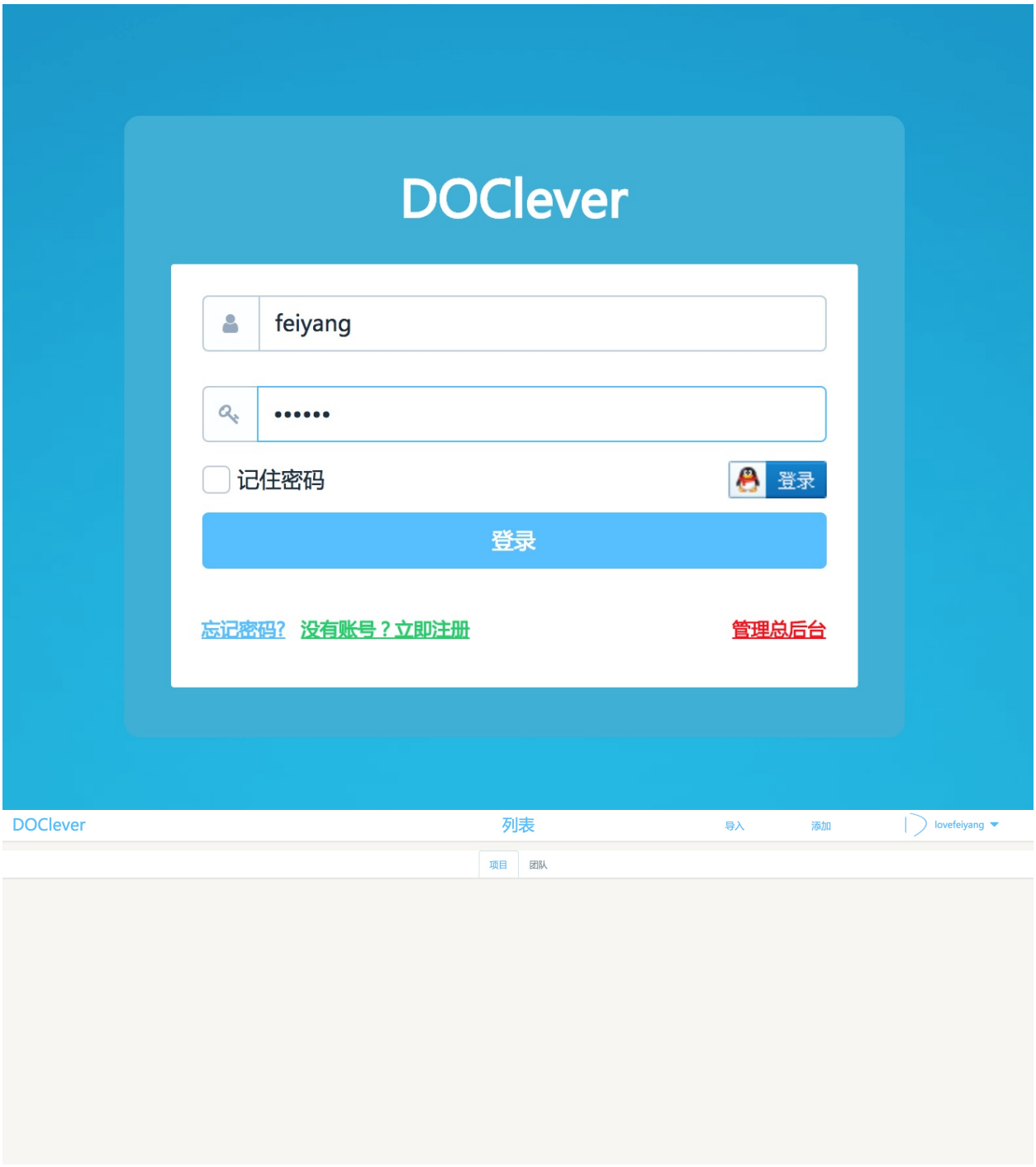
当你进入DOClever首页的时候，请点击注册按钮。

第二步：

在注册页面填写相关注册信息，点击注册按钮完成注册。

第三步：

注册成功后会跳转到登陆界面，填写用户名和密码，进入项目列表页。


DOClever
feiyang
记住密码
登录
登录
忘记密码?
没有账号？立即注册
管理总后台
DOClever
列表
导入
添加
lovefeiyang
项目
团队

创建工程

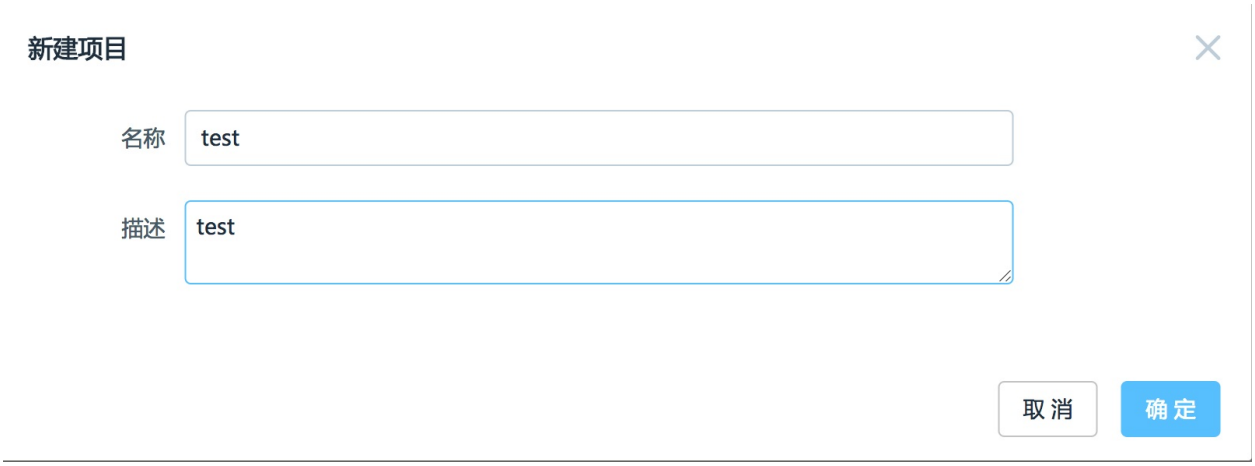
普通创建

第一步：

在项目列表页的右上方有创建按钮，我们在下拉菜单选择第一个创建项目

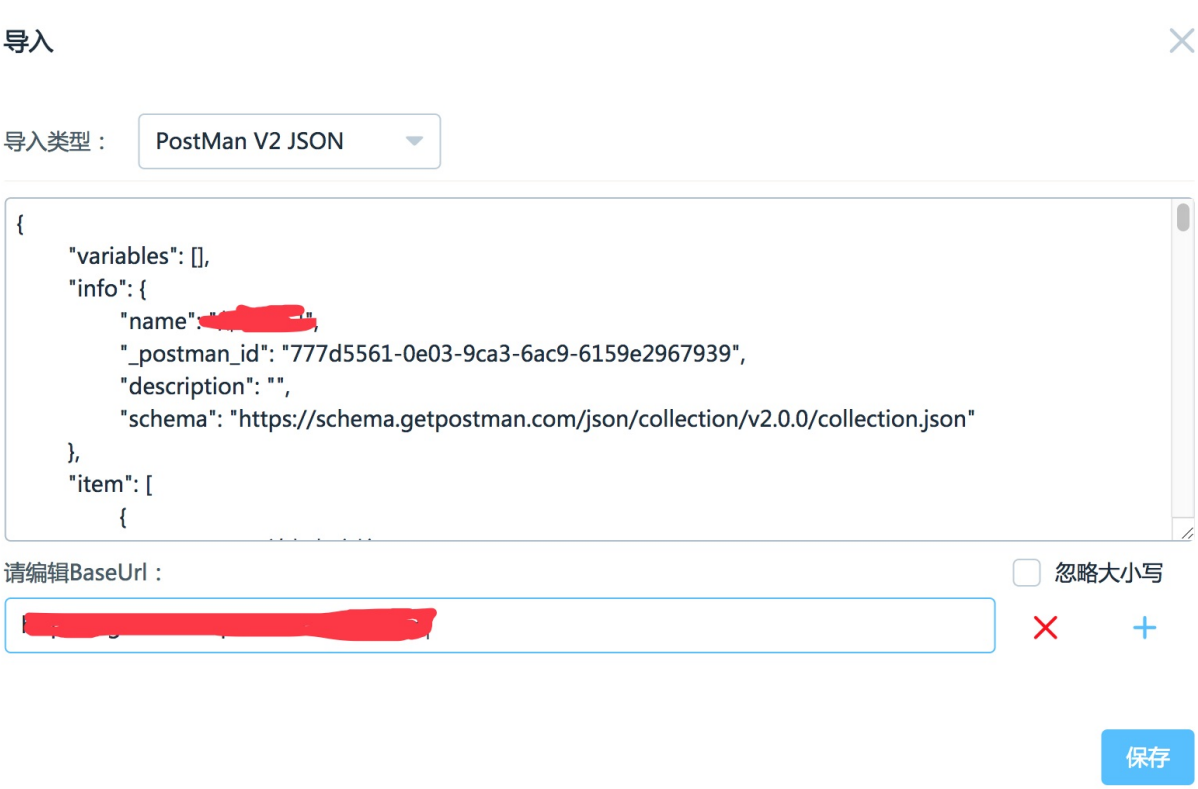
第二步

完成项目名称和描述的填写，点击确定，在页面上面便会出现我们刚才创建的项目了。



导入
DOClever目前支持4中内外部JSON格式的导入：postman v2导出JSON，DOClever导出JSON，RAP导出JSON，Swagger导出JSON

1.postman导入
将postman json的内容复制到文本框中，然后将改工程的baseurl（接口的根路径）填入下方输入框，可以填入多个，如果选中忽略大小写，这样在解析json的过程中就对baseurl的大小写不敏感了！点击保存，导入成功！

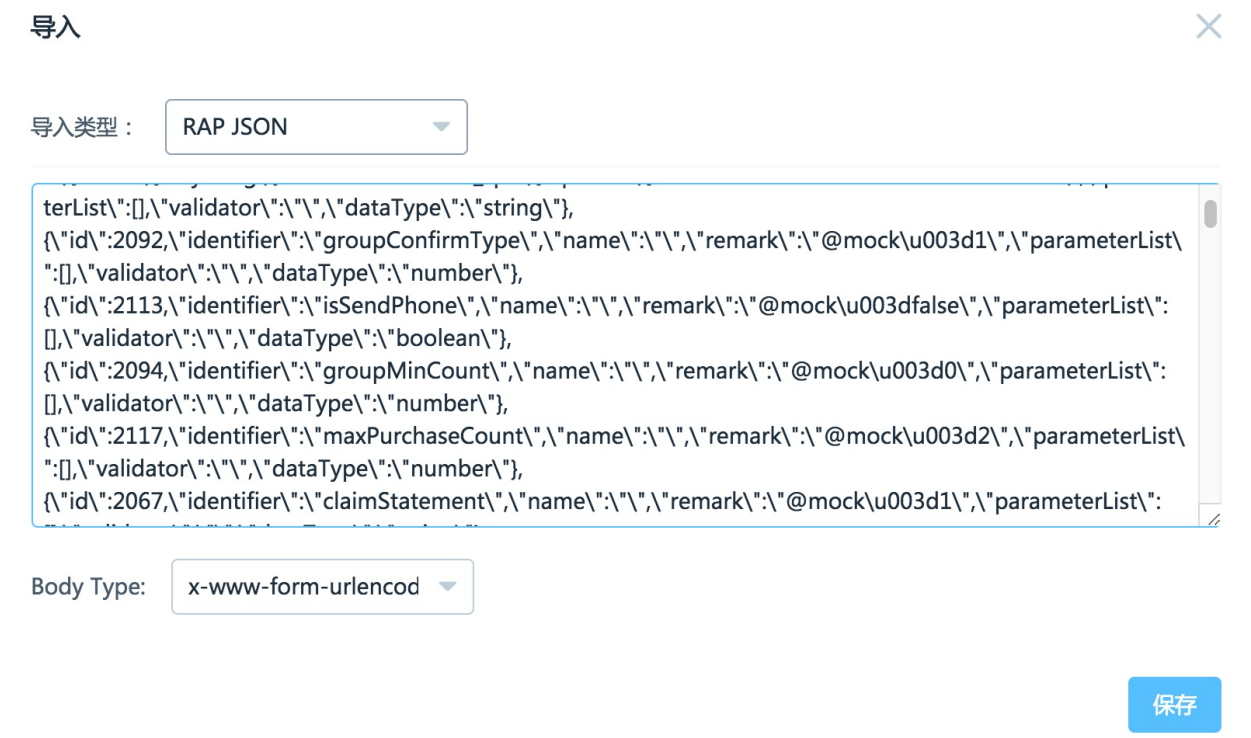


2.DOClever 导出JSON导入

可以将DOClever工程导出JSON导入为新的工程，常用于DOClever项目的迁移和备份。

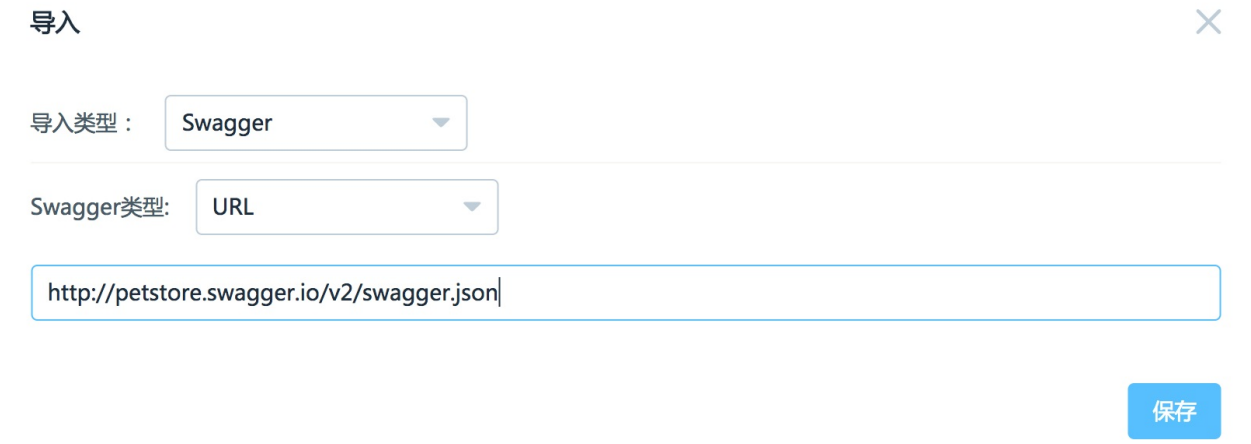
3.RAP导入

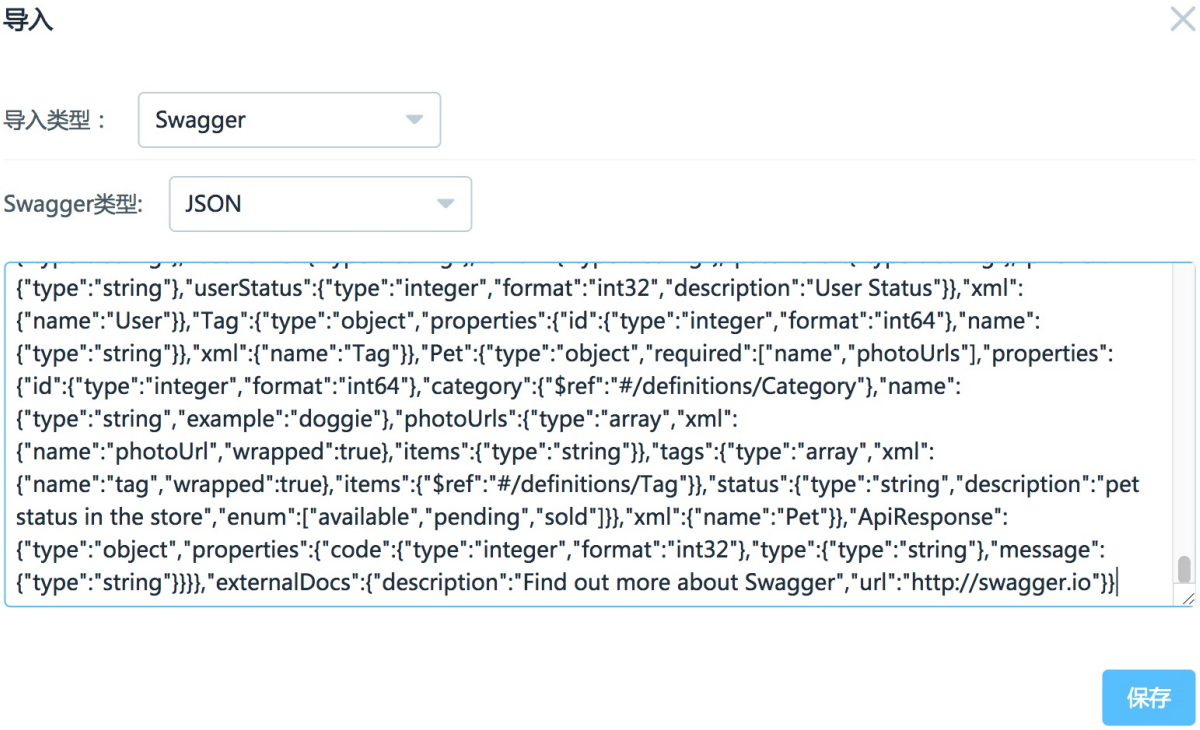
这里要注意的一点，对于post这样的需要传body的接口，在导入的底部有一个body type，选择我们需要的body type，DOClever会将工程解析为相应的格式。



4.Swagger导入

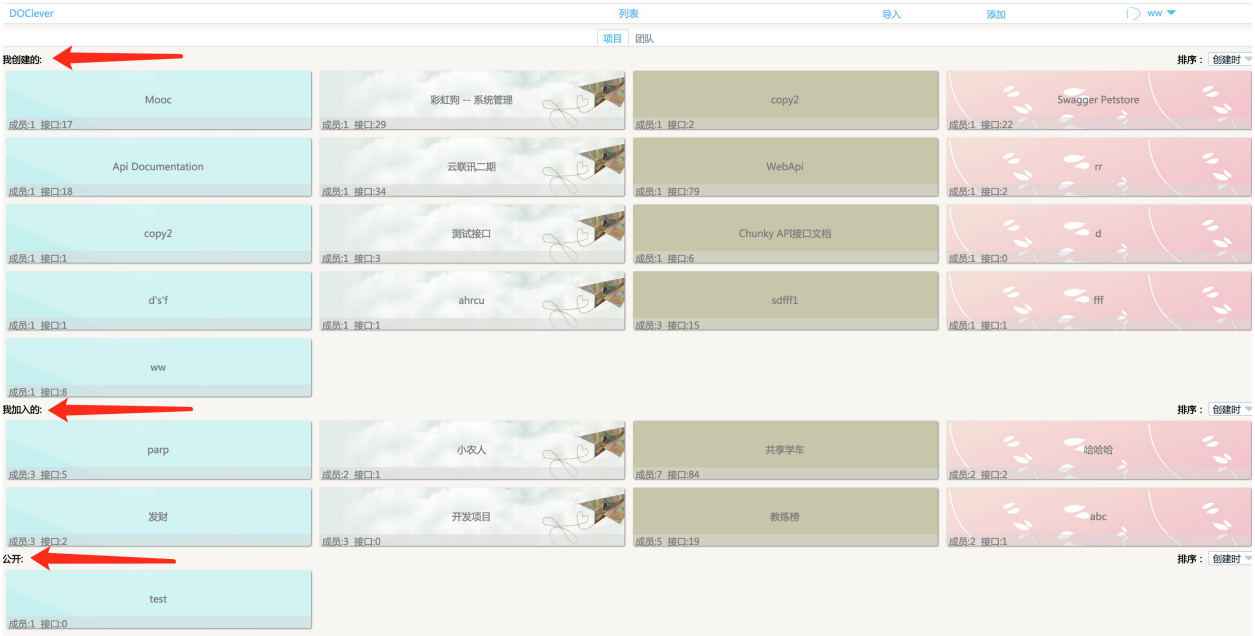
Swagger导入分为两种，一种是通过swagger的json url来导入，还有一种是通过json来进行导入，在目前的版本中，swagger url导入后的工程会托管其url来更新工程，这样当swagger有更新的时候就可以将其DOClever工程保持同步。




导入
导入类型：
Swagger
Swagger类型:
JSON
{"type":"string"},"userStatus":{"type":"integer","format":"int32","description":"User Status"}},"xml":
{"name":"User"}},"Tag":{"type":"object","properties":{"id":{"type":"integer","format":"int64"},"name":
{"type":"string"}},"xml":{"name":"Tag"}},"Pet":{"type":"object","required":["name","photoUrls"],"properties":
{"id":{"type":"integer","format":"int64"},"category":{"$ref":"#/definitions/Category"},"name":
{"type":"string","example":"doggie"},"photoUrls":{"type":"array","xml":
{"name":"photoUrl","wrapped":true},"items":{"type":"string"}},"tags":{"type":"array","xml":
{"name":"tag","wrapped":true},"items":{"$ref":"#/definitions/Tag"}},"status":{"type":"string","description":"pet
status in the store","enum":["available","pending","sold"]}},"xml":{"name":"Pet"}},"ApiResponse":
{"type":"object","properties":{"code":{"type":"integer","format":"int32"},"type":{"type":"string"},"message":
{"type":"string"}}}},"externalDocs":{"description":"Find out more about Swagger","url":"http://swagger.io"}}
保存

项目分类

在项目列表中，可以将项目分为三种类型：我创建的，我加入的，公开。



我创建的：

这个分类里面都是有用户本人建立的项目，用户对改项目拥有绝对的权限。

我加入的

这个分类里面是由别个用户建立的项目，用户参与到该项目中的。

公开

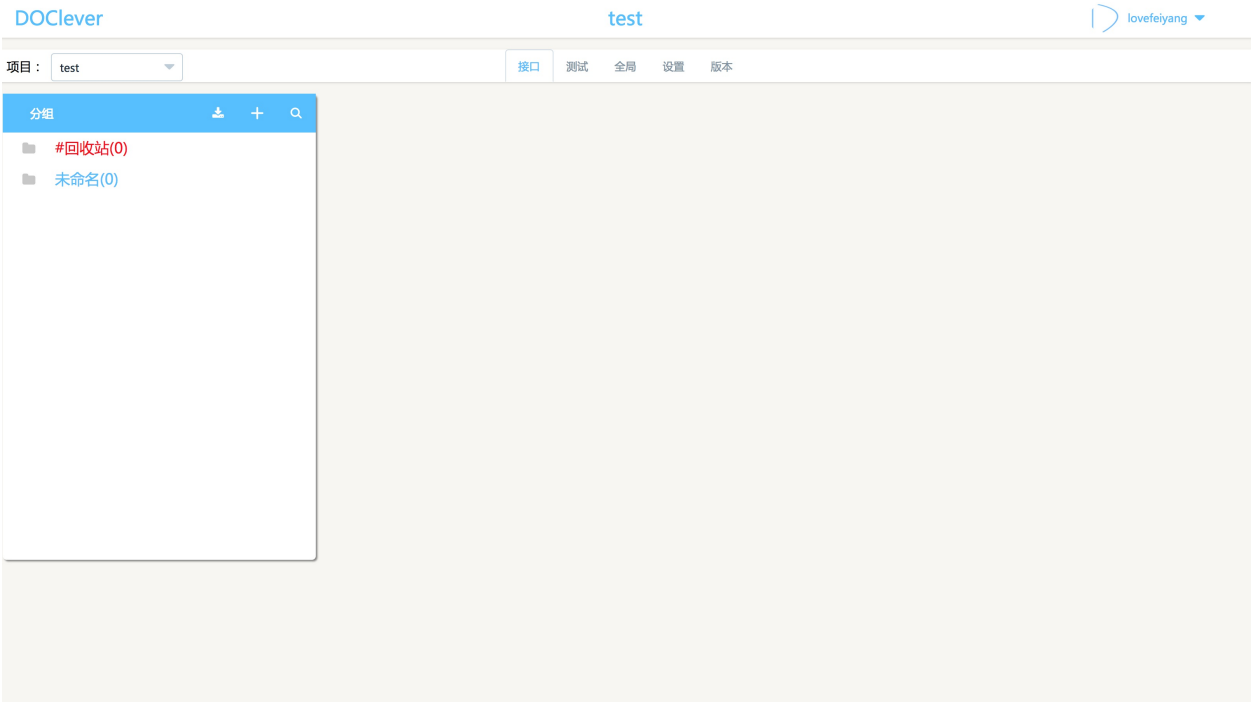
这个是所有的注册用户都可以看到的项目，但是对该项目只有浏览的权限。

接口管理

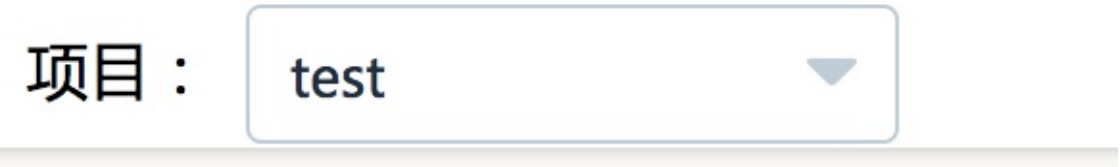
创建接口

第一步:

在项目列表页点击进入我们刚才创建的test项目，会来到项目主页

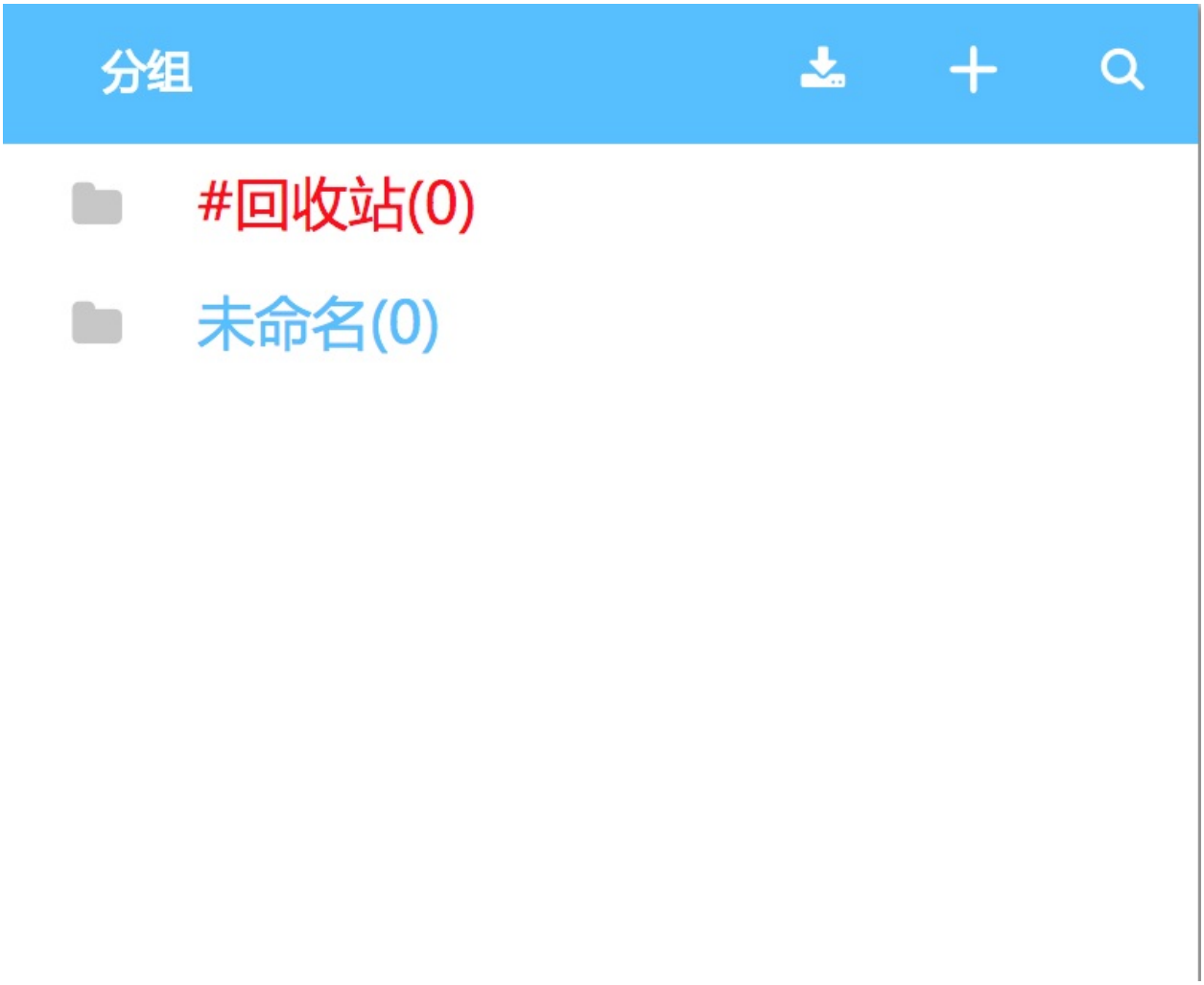


这里来详细说明下



左上角的这里是项目快速切换下拉框，在这里你可以快速切换你的项目。

中间位置是项目的几个功能的tab，接口就是接口的处理页面，测试就是自动化测试页面，全局里面可以设置项目的各种全局变量，设置可以对项目的各项功能参数进行设置，版本可以对项目进行版本处理。



这个区域是接口列表，右边有三个按钮，第一个是导入DOClever的分组JSON（这个JSON是有已有的分组导出的），第二个是新建分组，第三个是搜索接口。
然后看列表，每个新建的项目（导入的除外）都有回收站和未命名两个分组，最上面红色的是接口的回收站，这个和电脑的回收站类似，都是待删除的接口会移动到这里面来，而下面的未命名，你可以根据需要修改该分组的名称，在其分组下可以新建接口！

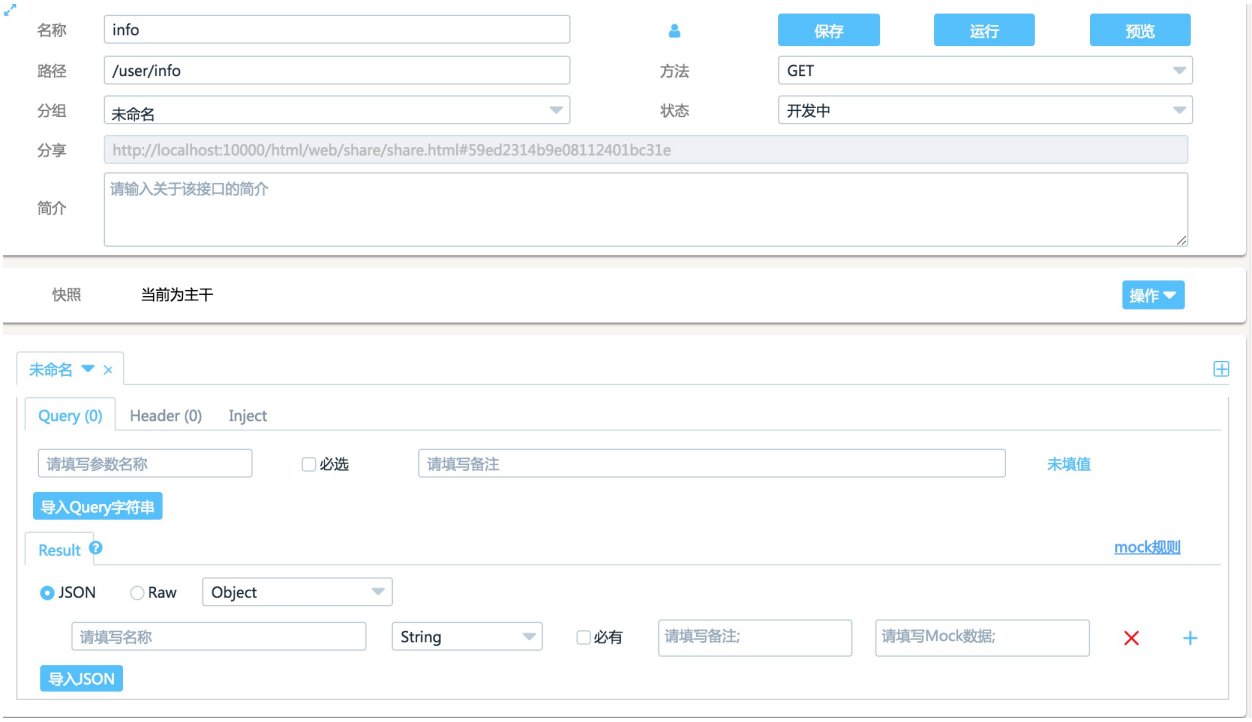
我们将鼠标移动到未命名分组上，其右边会出现编辑的图标，移动上去，选择新建接口

这时在屏幕页面的右侧会弹出接口页面，等待我们输入保存。

我们输入接口的基本信息，点击保存，在接口列表页便会出现我们新创建的接口

接口编辑

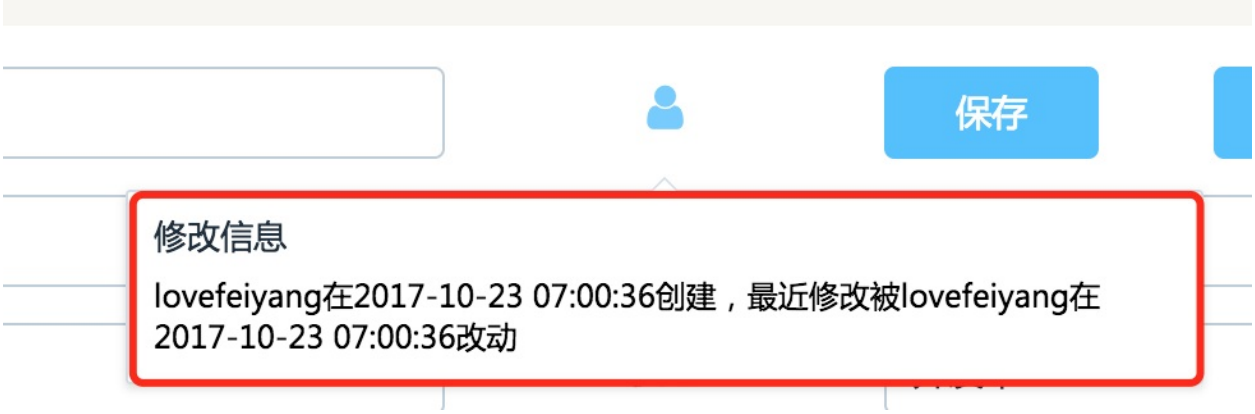
我们再次点开info这个接口，来详细的给大家说明下如何去编辑一个接口



上图中接口编辑页面可以分为三个部分：接口的基本信息，接口快照，接口的参数信息。

基本信息：

名称： 接口的名称



这里表示接口的创建者创建时间，最后操作者和操作时间

保存： 点击保存，该接口的数据会被保存下来

运行： 点击运行，可以调试该接口的数据

预览： 点击预览，会切换到文字版本的展示页面

路径： 这里可以填入接口的路径地址，路径不包含baseUrl，baseUrl请前往左边全局标签页里面设置。例如http://abc.com/login, http://abc.com 是baseUrl,这里输入/login即可,支持restful url形式，例如：/info/{name} 支持路径参数的粘贴，系统会自动识别路径和query参数

方法： 支持GET，POST，PUT，DELETE，PATCH五种http方法

分组： 接口所在的分组

状态： 分为开发中，开发完成，已废弃，具体使用会在Mock数据里面做详细解释。

分享： 这个链接可以分享给没有注册DOClever的用户，他们通过这个链接打开的就是这个接口的预览界面。

简介： 就是关于这个接口的描述

快照

快照我们会在版本管理-接口快照里做详解！

参数信息

对于很多接口，不同的入参会返回不同的出参，所以在DOClever里面，可以有多个入参出参的配对，每个配对可以称为一个参数实例。

上图中未命名就是一个参数实例，我们可以点击右边的加号来添加多个实例

Param: 当我们在路径里面填写含有restful形式的url时，比如：/user/info/{id},DOClever会自动解析路径，并且提供对param参数的支持，如下图

Query: 这里代表的含义类似于http://aaa.com/user?name=sx&pass=111 这个地址中的name=sx&pass=111这样的参数，我们需要填入的是参数名称，参数是否必选，备注，参考值。点击未填值，弹出如下图所示：

填入该参数的参考值和备注，点击保存

Header: 这里的含义代表我们发送该请求的自定义http头部

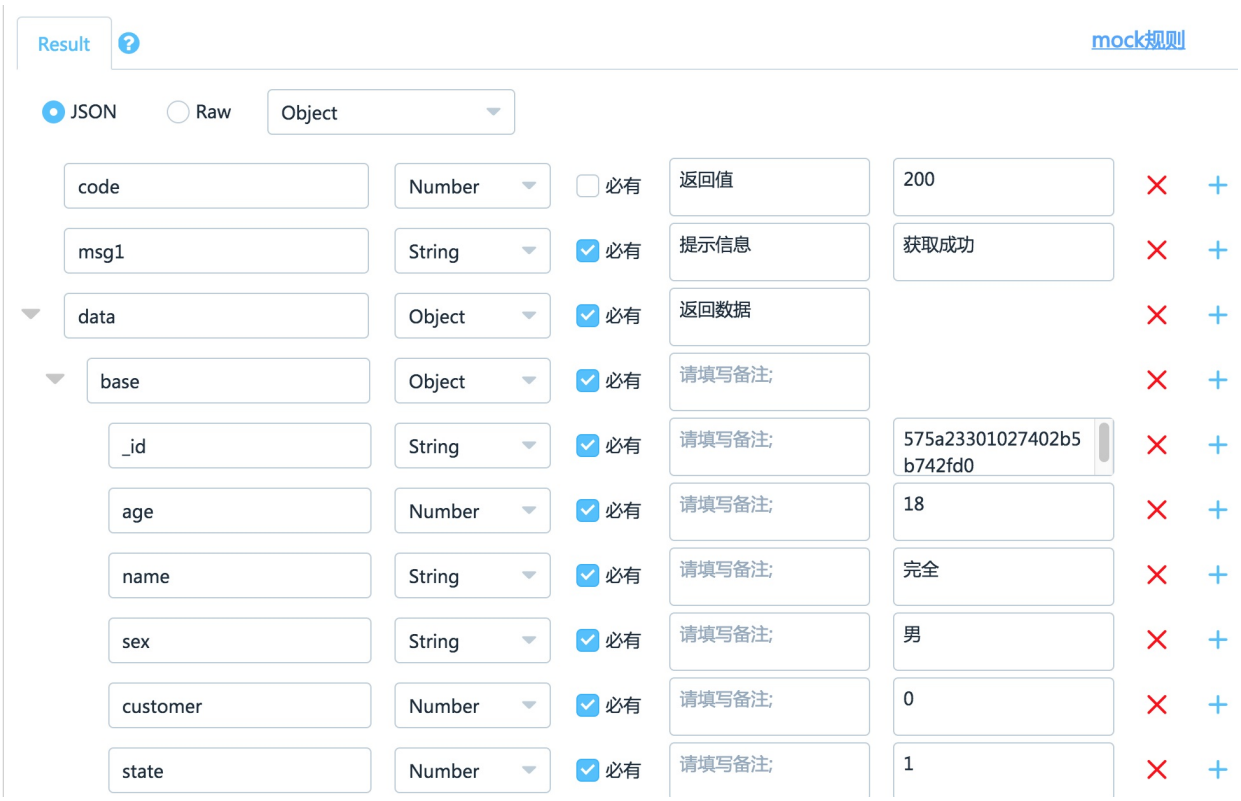
Body: 当我们切换http method到post，put或者patch的时候，会出现Body这个标签页，我们切换到这个标签页，如下图：

这里分为两种模式，一种是以键值对形式（key-value)，还有一种是把数据直接传给http body(raw)，当Header里面的content-type为空或者为application/x-www-form-urlencoded，multipart/form-data的时候，DOClever会切换为key-value的形式，其他情况会切换为raw。
key-value：值得注意的是参数类型，可以有文本和文件两种，当选择文件的时候，DOClever会切换到multipart/form-data的方式发送数据。
raw：我们可以现在不同的数据类型，选择不同的类型header里面的content-type也会发生相应的变化，当现在JSON的时候，会出现如下图所示的界面：

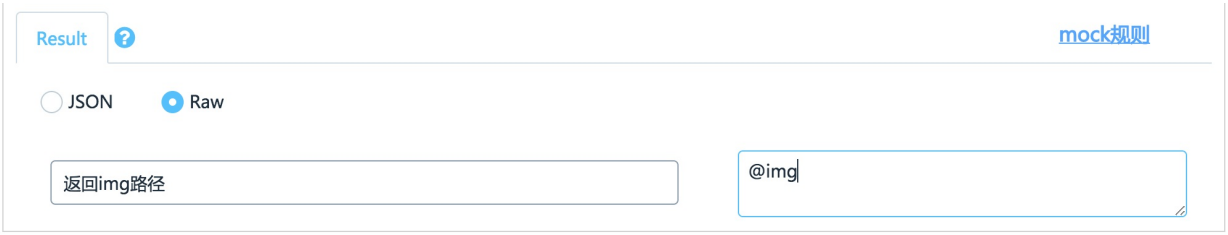
JSON的具体操作会在出参里面做具体讲解

Inject: 这一部分在注入里面做详细讲解

Result: 这里代表的是出参的数据模型，出参也分为两种，JSON，RAW。
JSON：这里又分为Object和Array两种类型，Object代表返回的是一个json对象，Array代表返回的是一个json数组，参数字段从左到右依次为名称，类型，是否必有，备注，Mock或者参考值，删除按钮，新增按钮，这里值得注意的是类型，在DOCLever里面，json的类型有六种：String，Number，Boolean，Array，Object，Mixed，前五种都好理解，最后一种Mixed代表的是任意类型，当我们的参数类型不定的时候，就可以用Mixed来代替。当参数的类型为Array或者Object的时候，我们可以为该字段添加子字段，拖动该字段，我们还可以改变字段的顺序，或者父子关系。



RAW:如果返回的是其他类型的数据，就可以在RAW标签页里面填入相关的信息。



接口分组

DOClever的接口分组是可以无限层级的往下分的，我们可以导入分组，导出分组，移动分组等。

新建分组

点击加号，出现新建分组弹出框，填入我们的信息，点击确定，完成新建分组

鼠标移动到任意分组上，在右边的编辑图标会出现下拉菜单，选择新建分组，就可以在一个分组内建立子

分组了。

移动分组

拖动接口或者分组，移动到其他分组上，当颜色出现绿色的时候，代表可以移动到这个分组下，如果是拖动到回收站，则代表删除这个分组或者接口，注意：分组不可以拖动到自己的子分组下。

导出分组

我们可以备份某个分组的数据，在刚才的下拉菜单下选择导出分组就可以把这个分组的数据导出为JSON格式了。

导入分组

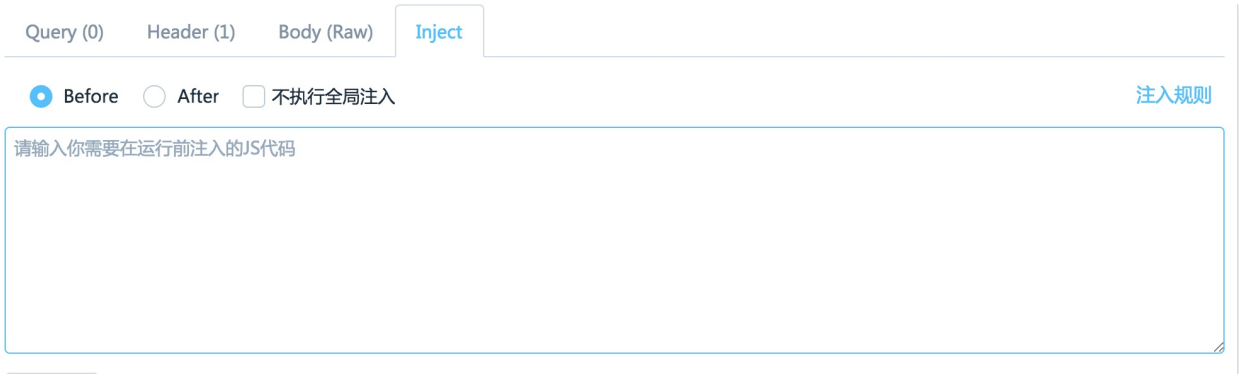
当我们导出分组后，就可以在其他的分组或者项目下面将这个分组导入，依然是在上面的下拉菜单下面选择导入分组即可。

接口搜索

点击最右边的搜索按钮，即可搜索接口，可以按照接口的名称和路径来搜索。

注入

我们可以在运行一个接口开始前和结束后进行js代码的注入操作，这样我们可以自由的修改很多功能，比如我们需要对接口的字段进行自定义的加密处理，那么这种情况只能注入代码了。DOClever提供了很多内置变量供用户操作！



Before:
接口在运行前执行的代码，内置变量：
url:BaseUrl字符串
path:path路径字符串
method:HTTP方法字符串
param:Object对象，里面存放了Param的数据
query: Object对象，里面存放了Query的数据
header:Object对象，里面存放了Header的数据
body:Object对象，里面存放了Body的数据
Base64,MD5,SHA1,SHA256,SHA512,SHA3,RIPEMD160:这些加密函数只有一个参数，为加密的字符串
AES,TripleDES,DES,Rabbit,RC4,RC4Drop:这些加密函数有两个参数，第一个参数是加密的字符串，第二个参数是salt，值得一提的是在使用这些加密的时候，最好在后面加上一个toString()方法，这样会确保编码没有问题，比如MD5("aaa").toString()

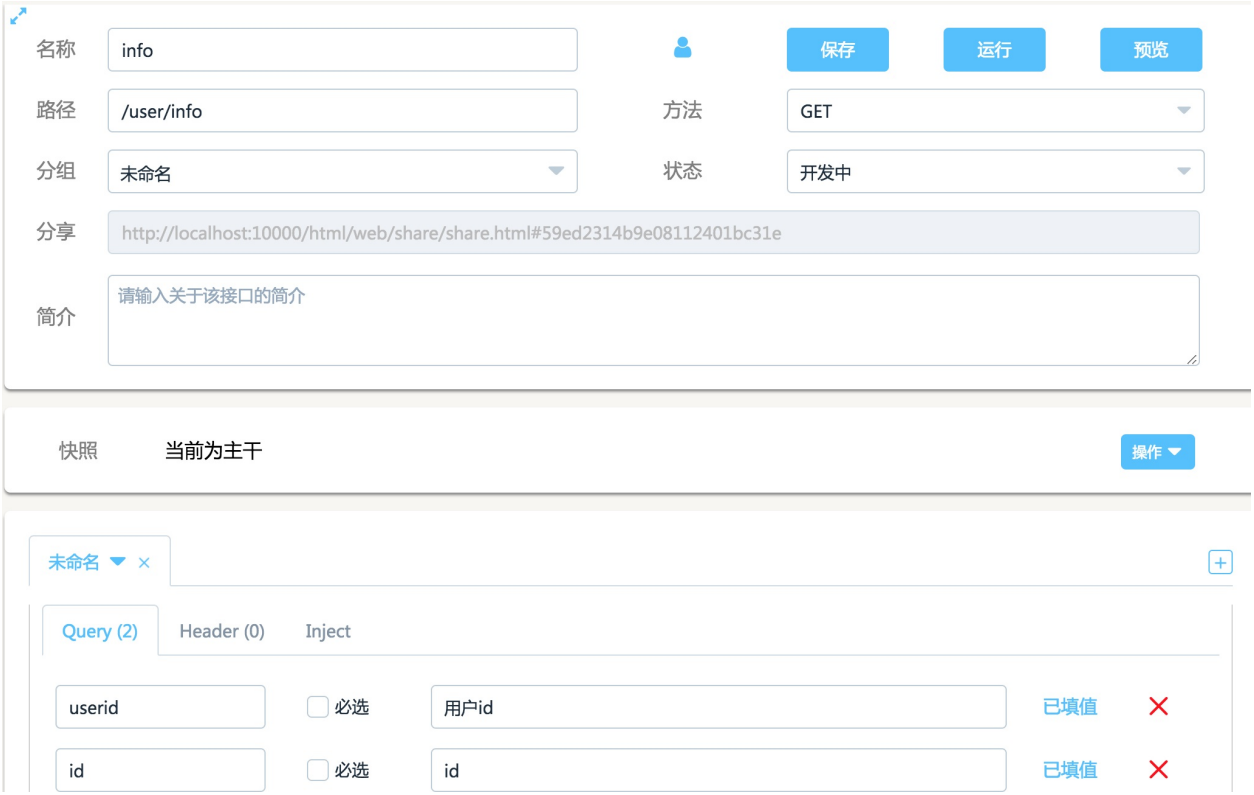
After:
接口在运行后执行的代码，内置变量：
status:接口返回的状态字符串
header:Object对象，接口返回的HTTP头部
data:不确定，可能是JSON对象，可能是String，可能是Blob，根据具体接口来

这里有一个选项：不执行全局注入，如果我们选中了，则不会去执行全局注入，而只会取执行针对于当前接口的注入。

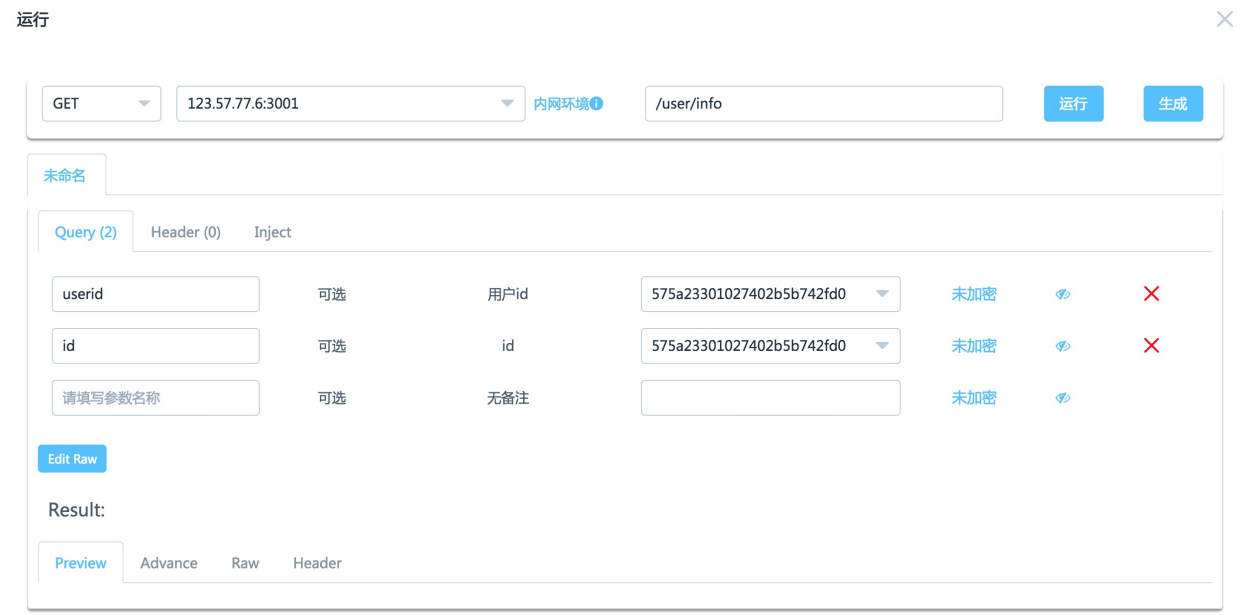
接口调试

接口测试

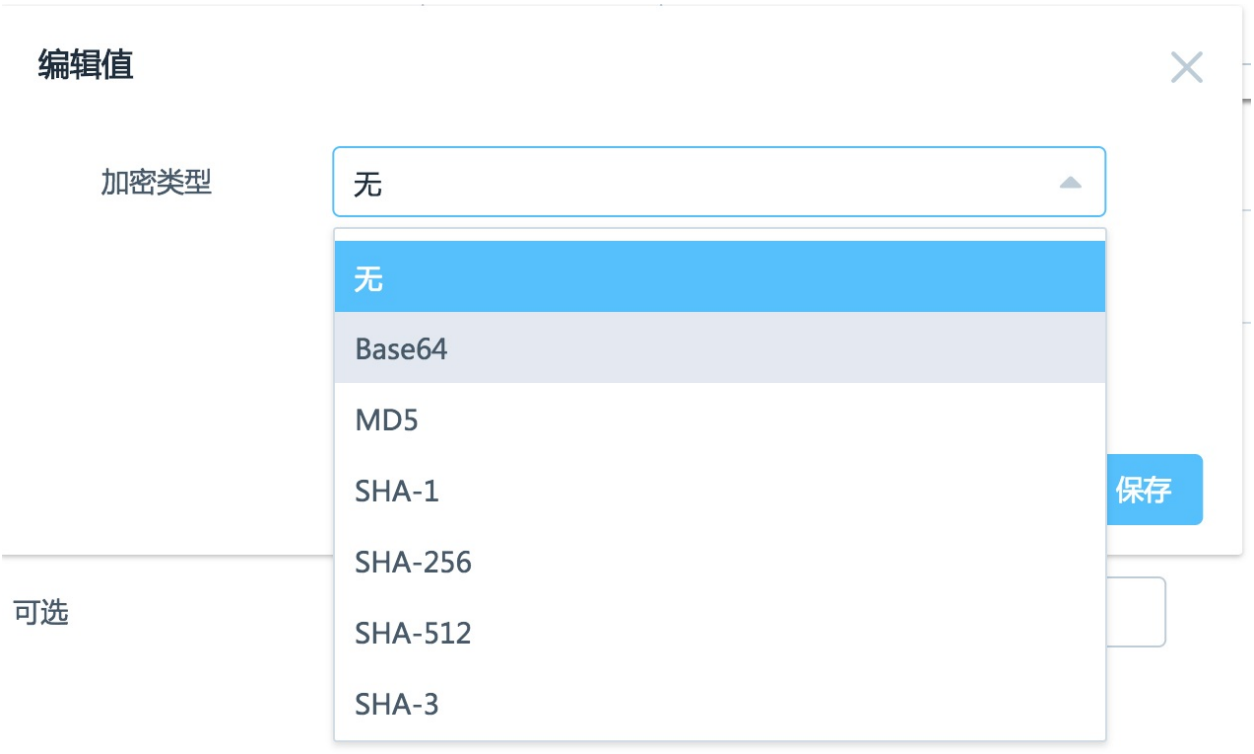
当我们编辑好接口后，我们肯定需要运行这个接口看数据正不正确，DOClever提供了强大的接口测试功能，可以很好验证数据的正确性。



对于如上的接口，我们点击运行按钮，弹出运行界面

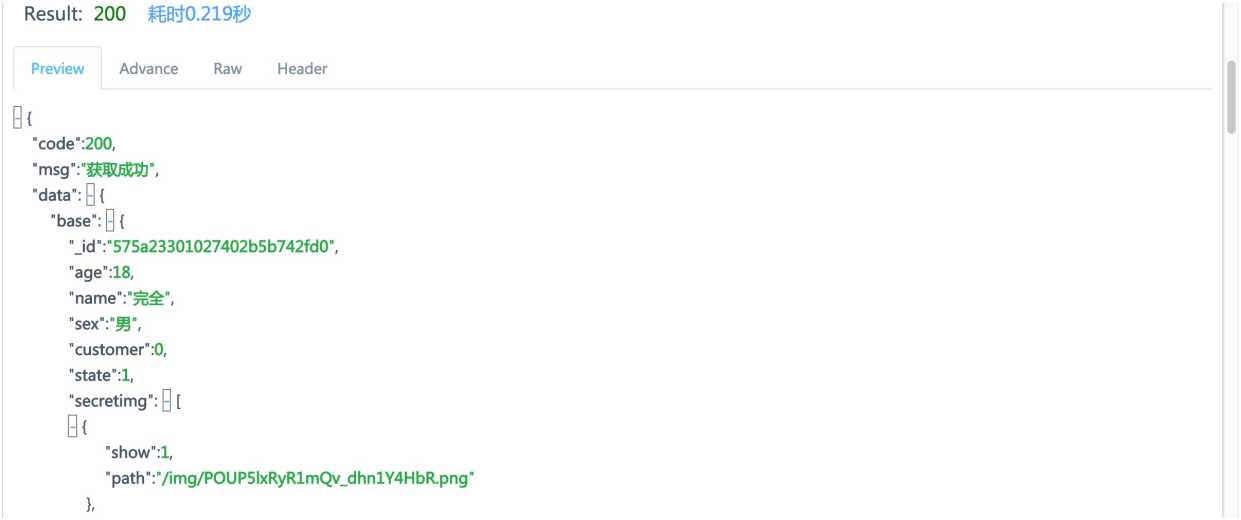


在运行界面里，我们输入或者选择BaseUrl，入参如果之前是已填值的，那么这里会默认选中第一个值，如果你有需要，还可以对参数进行加密



当然，你也可以现在某个参数不发送

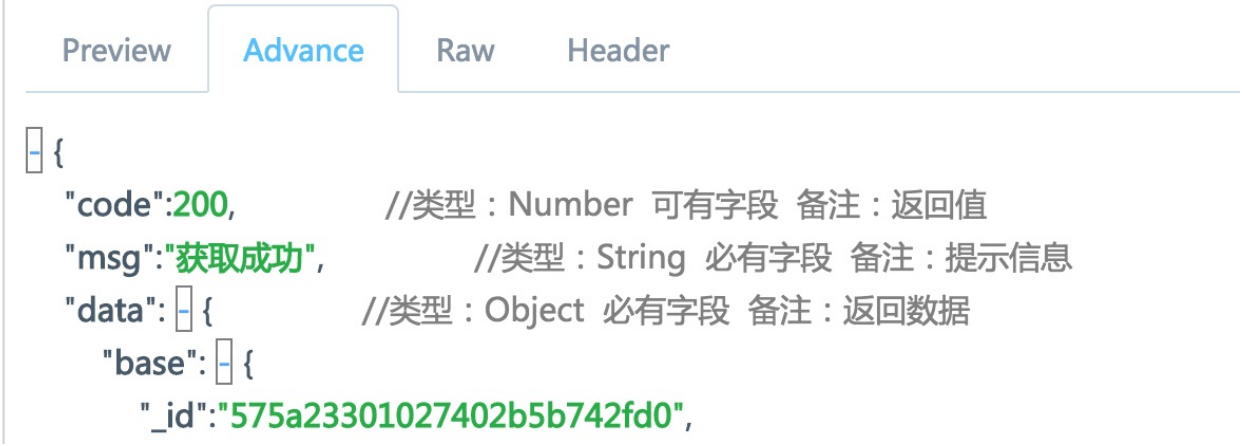
点击运行按钮，在底部会出现如下信息：



这里有四个tab：Preview，Advance，Raw，Header

Preview： 这里是格式化后的json数据

Advance： 这里显示每个字段的详细信息，如下图



Raw： 这里显示返回的最原始数据，如下图

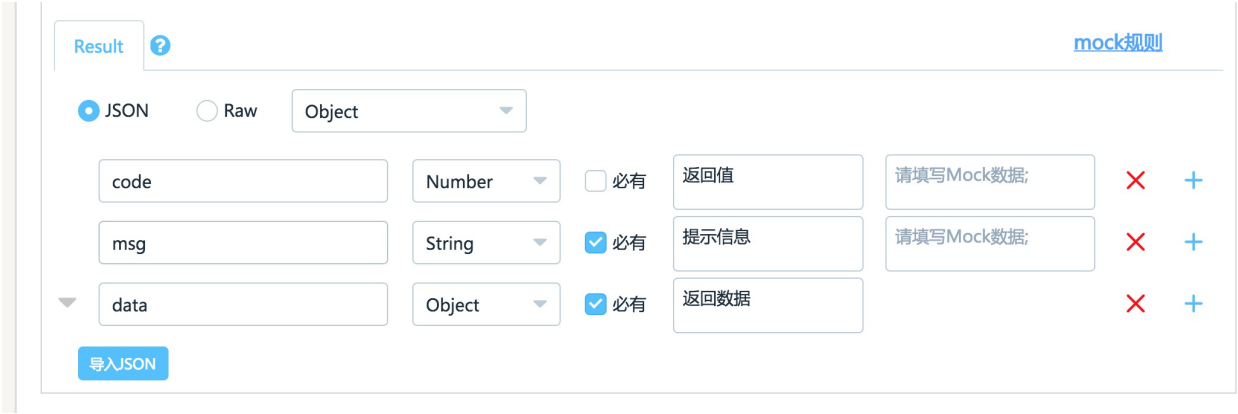
Header: 这里显示返回的http头部


Preview
Advance
Raw
Header
date
Tue, 24 Oct 2017 07:50:21 GMT
connection
close
x-powered-by
Express
content-length
1469
content-type
application/json; charset=utf-8

一键生成

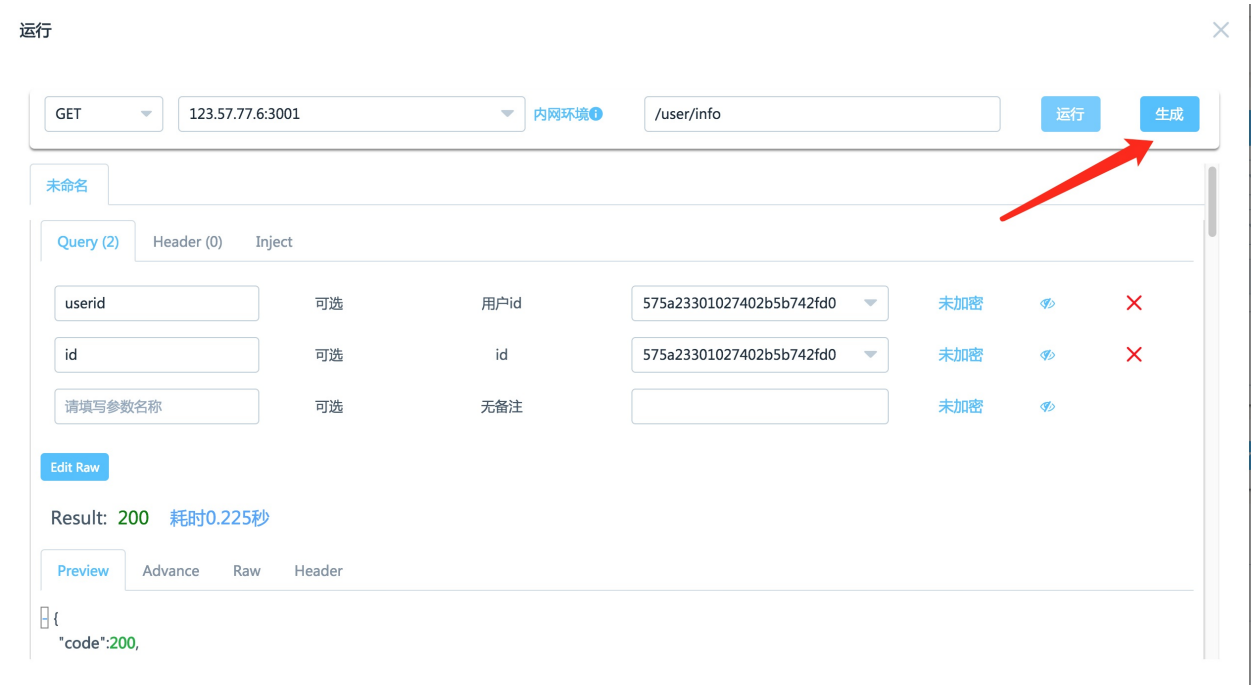
通常，接口的返回数据都是比较复杂的，而且很多时候接口的返回数据都是边开发边修改的，所以我们不可能每次都是手动的一个个去定义返回数据，这样太繁琐，效率太低了，DOClever提供了接口数据的一键生成功能，可以很快捷的根据接口的真实入参出参生成对应的数据模型！

举个例子，还是刚才的接口，我们来看下出参：

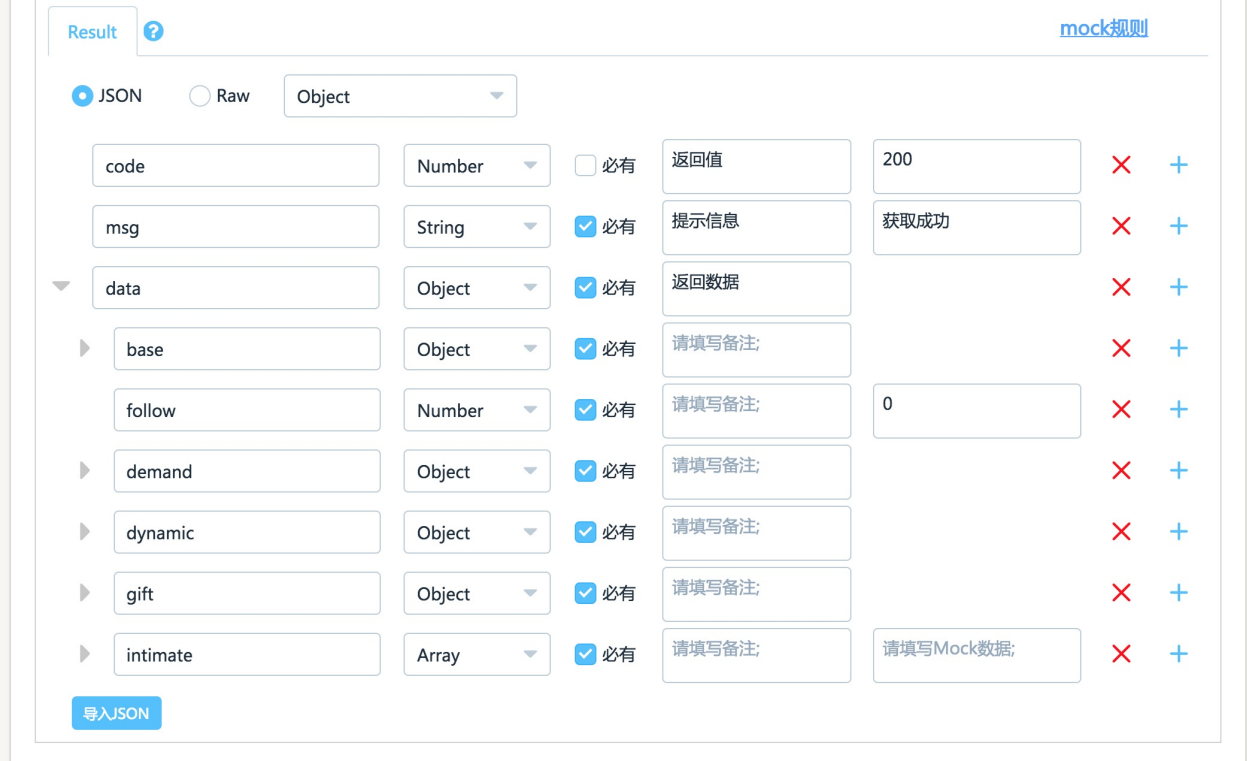


这里定义了三个参数，code，msg和data，但是data里面还是有子结构的，而且数据比较复杂，我们怎么根据接口的返回数据自动生成这个结构呢。

切换到接口的运行界面



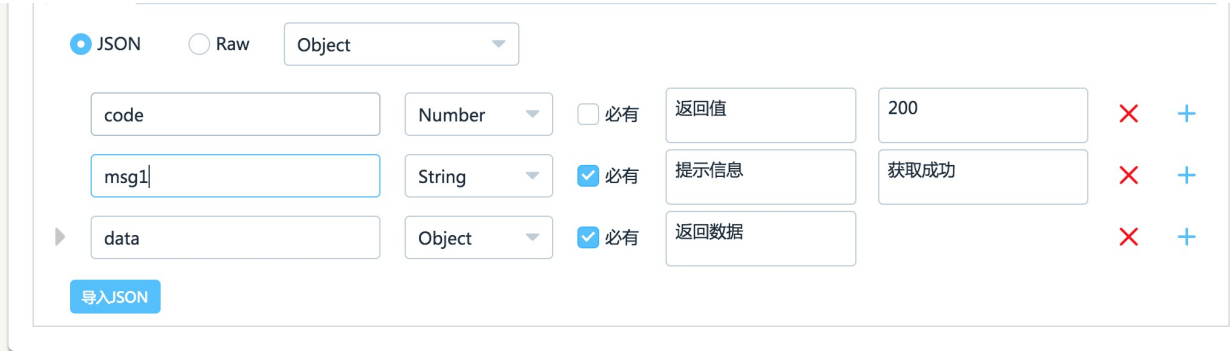
我们在运行完接口后点击图中所示的生成按钮，如果你当前输入的BaseUrl不在项目的BaseUrl的列表里，系统会提示你是否添加，然后我们查看接口编辑页面的Result，发现data的子结构已经被填上了。



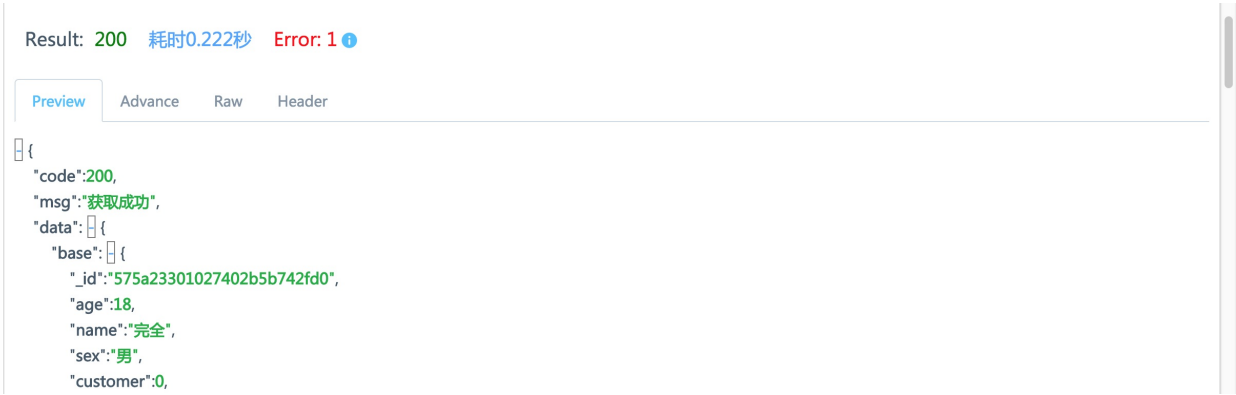
不光如此，对于入参，也是一样，你可以在新建一个接口的时候填写完接口的基本信息后就运行接口，在运行页面填入入参和出参，运行接口后然后点击生成，这个接口的参数数据模型就会自动生成，点击保存即可保存这个接口的数据了！

数据效验

当我们定义完接口的返回数据模型后，我们肯定需要知道是否和我们的真实返回数据一致，DOClever提供了数据效验的功能，可以验证我们数据的一致性！
比如在接口编辑页面有这样的数据模型：

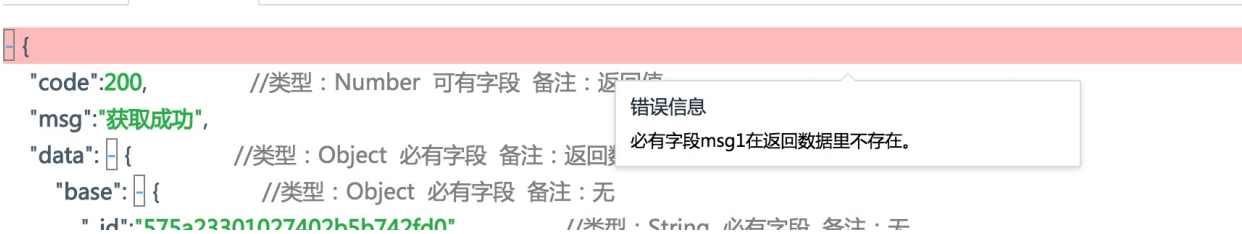


然后我们点击运行，如下图：



我们看到了有Error的提示，我们鼠标移动到上面，出现提示：

我们切换到Advance标签页



可以看见真实的返回数据和我们的定义的数据模型一对比，发现msg1这个字段在返回数据里面是不存在的，这样可以很多的帮助我们排查接口出现的问题！

内网调试

内网调试分为两种情况：
线上环境：如果你的接口服务是在本地或者局域网的，外网环境访问不到的，那么这个时候就需要内网调试了。
线下环境：你的DOClever服务端和接口服务不是在同一个内网里的或者类似于线上环境的那个情况。

如何进行内网测试：
1.在你本地的机子上首先需要安装node环境，推荐node最新的lts版本（windows，linux（mac））
2.将net.js保存到本地，用node运行，如下图

```
sunxindeMacBook-Pro:resource sunxin$ node net.js
正在运行中，请用DOClever的接口测试页面进行内网测试！
```

3.对于线下用户来说，还需要确保右上角个人头像下拉菜单里的Proxy处于开启状态

4.现在就可以用运行页面测试接口数据啦！

Mock数据

对于我们产品开发来说，正常流程是设计原型，设计UI，后端设计数据库和接口，前端开发页面调用接口，对于前端来说，很多时间都是在等待接口的完成，这样会耗费很多时间，如何更好的利用这部分时间进行前后端并行开发呢，于是便有了Mock数据的诞生。
简单来说，Mock数据就是借助一些后端来产生假数据返回给前端，让前端的业务流程可以跑下去。目前市面上也有很多工具专门做mock数据的，其实mockjs就是其中最有名的一款，DOClever同样对其进行了支持。

基础

下面罗列了DOClever自身的一些mock规则：
@date 针对String类型，生成当前日期
@num(min,max) 针对Number类型，生成min到max范围之间的数字，比如@num(1,100)
@in(val1,val2,...) 针对String，Number类型，生成括号内的某一个值，比如@in(get,post,put,delete)
@img(width,height) 针对String类型，生成一个图片地址,如果填写@img，那么图片默认就是600x400
@null 针对String，Number，Boolean，Mixed类型，生成null值
@arr 针对Mixed类型，生成一个数组，比如@arr([123,45])
@obj 针对Mixed类型，生成一个对象,比如@obj({"name":"aa"})
@count(min,max) 针对Array类型，确定生成数组的大小,比如@count(1,10)
@code 针对String，Number，Boolean，Mixed类型，会执行自定义的代码并返回结果,比如@code(body["aa"])这个就会返回body参数里key为aa的值，有以下几个内置对象：param,query,header,body,global（global里面还有以下几个成员：name,baseurl,path,method)
如果不是以@字符开头的话 就直接生成你输入的那个值,当类型不匹配的时候会尝试去做类型转换
@mj 针对String，Number，Boolean，Mixed类型，提供对mockjs的支持，有两种写法，一种是@mj(DPD),例如@mj(@email),表示生成一个随机的email，还有一种是@mj({@DTD:@DPD}),例如：@mj({"3","a"})，表示生成"aaa"，具体的语法规则可以参考mockjs的官网

详解

对于接口的状态，目前分为三种：开发中，开发完成，已废弃。对于DOCLever，提供无缝mock的功能，所谓的无缝mock，就是在接口状态为开发中或者已废弃的时候返回定义好的mock数据，在接口状态为开发完成的时候返回真实接口返回的数据，那么如何去做呢，还需要用到net.js文件，本地需要最新的lts版本的node环境（windows，linux（mac）），在本地用node运行net.js ,加上mock server地址和你需要请求的真实地址的根地址，比如：node net.js http://localhost:10000/mock/59ec8b73b9e08112401bc31b http://localhost:8081 当您的接口的状态为开发完成的时候，net.js不会去请求mock server地址而去请求真实地址,然后将你本地开发工程下的根地址替换为localhost:36742，这个时候DOClever的无缝mock就默默的守护你的开发工程啦！

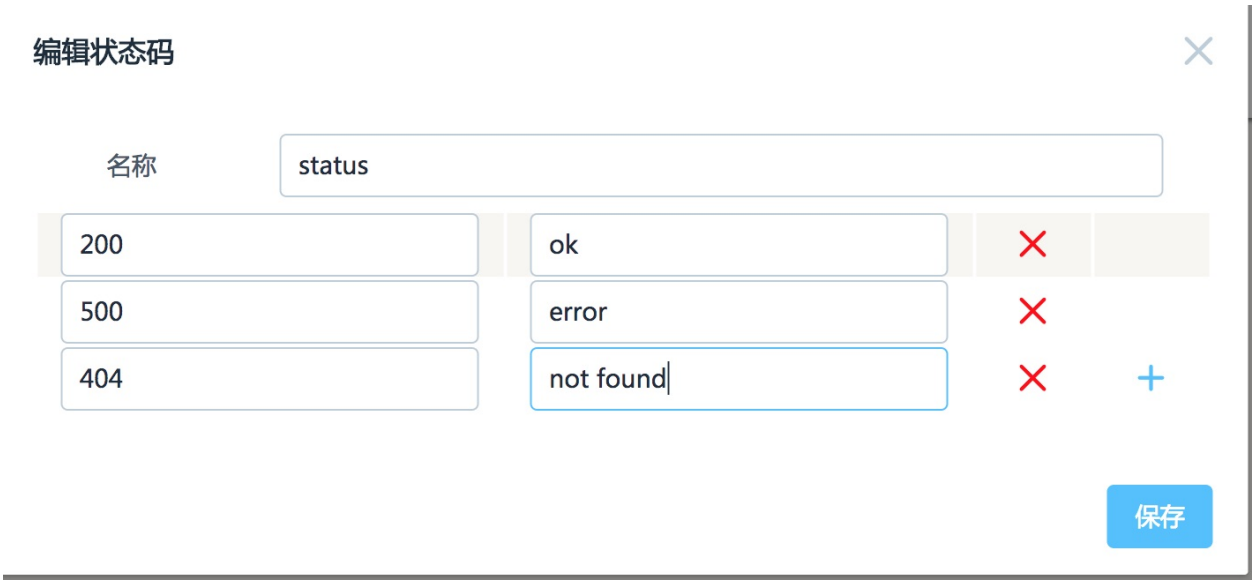
项目设置

状态码

状态码可以看成是一个数据模型，举个例子，比如我们接口返回的response里面有code这个字段，code为200的时候是正确，为500的时候是错误，，为404的时候是找不到资源，我们就可以将这些状态整理成状态码。

新建

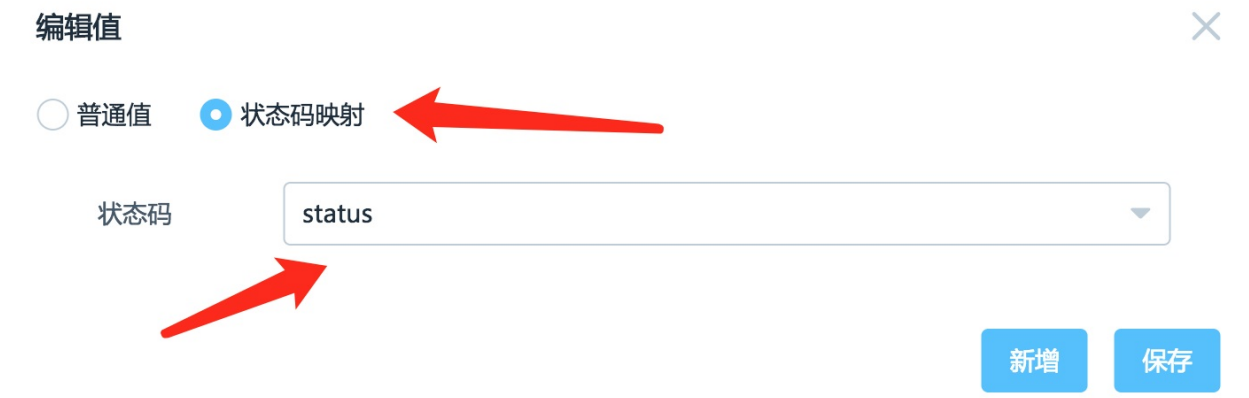
如上图所示，点击新建状态码，然后会弹出一个对话框，填入你的状态码和备注，保存即可



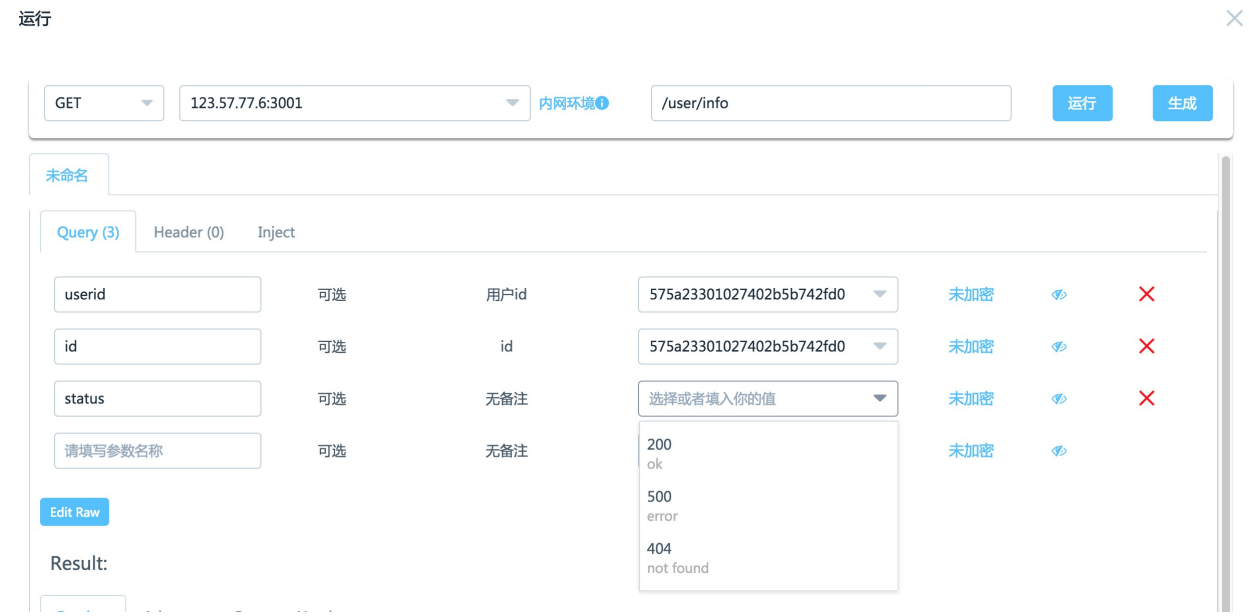
使用

入参：

对于status这个参数，我们点击未填值



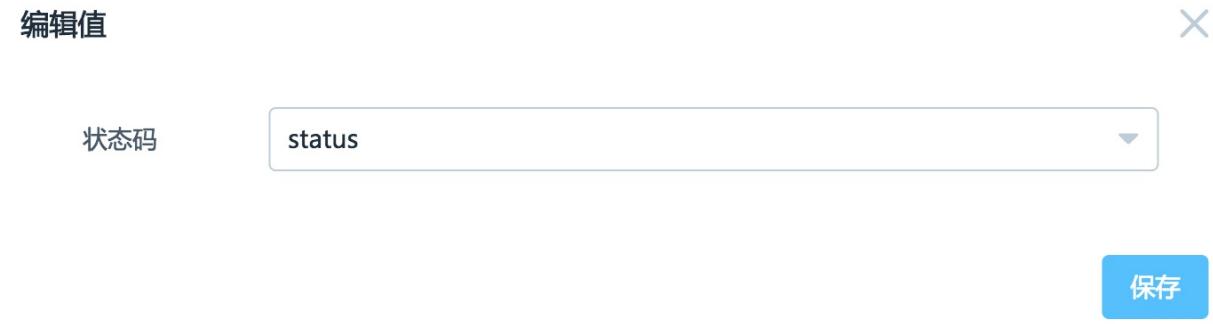
然后切换到状态码映射下，状态码选择status



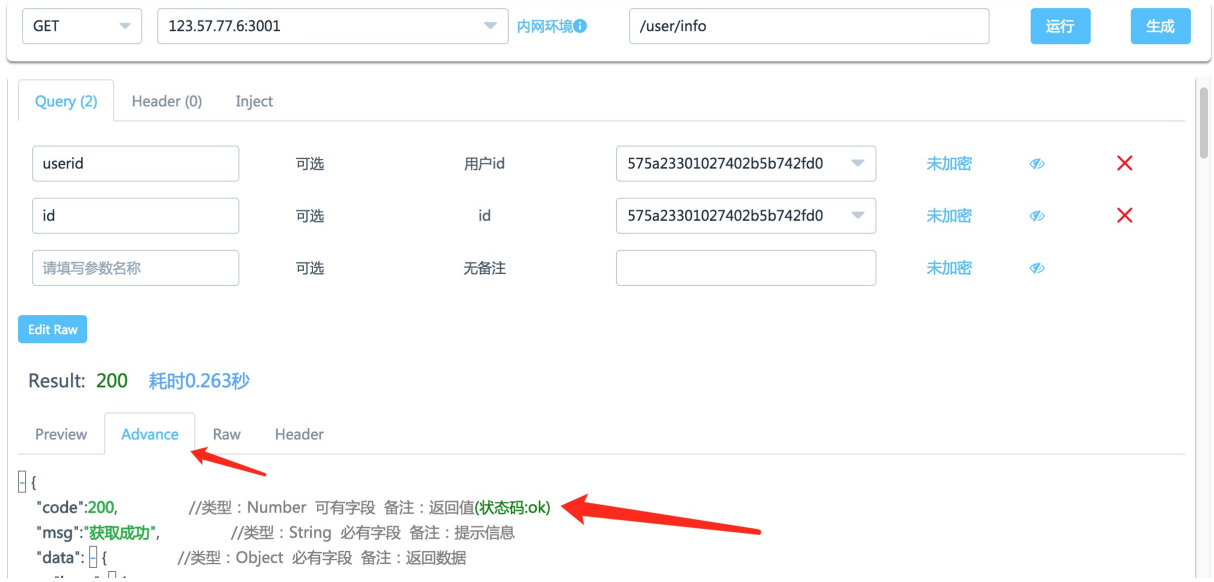
在接口运行页面，我们可以看到status这个状态码里面的值自动的填入了status这个参数下，供用户选择！

出参：

对于code参数，我们点击上图所指的那个图标



在弹出的选择框里我们现在status状态码



在接口运行页面，我运行接口，在Result的Advance标签页下，我们可以看到状态码的含义被正确的解析出来了

全局环境注入

全局的环境注入规则和接口的注入规则一样，只不过全局注入会多项目内的每一个接口都有效，且顺序是先执行全局注入，再执行每个接口的注入。

文档

文档目前的功能很简单，支持markdown语法，可以实时预览！

BaseUrl

每一个项目都有其项目运行所对应的环境，比如开发环境，测试环境，生产环境，而每一个环境往往对应着一个不同的BaseUrl，我们可以在全局的BaseUrl里编辑这些信息：

这样，在接口的运行页面你就可以选择BaseUrl了。

项目成员

对于一个项目，有两种项目角色：管理员和观察者。
管理员： 可以对该项目进行增删改查等操作，拥有最大的权限。
观察者： 管理员可以对观察者分配相应模块的权限，默认情况下对项目只有浏览测试的权限。

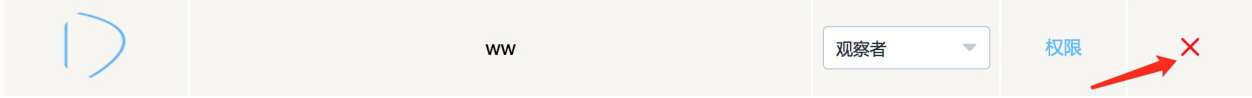
邀请成员

填入用户名，选择管理权限，点击邀请即可

修改权限

点击如上图所示，即可修改权限，当修改成观察者时，点击权限按钮，可以具体分配不同模块的权限

删除成员

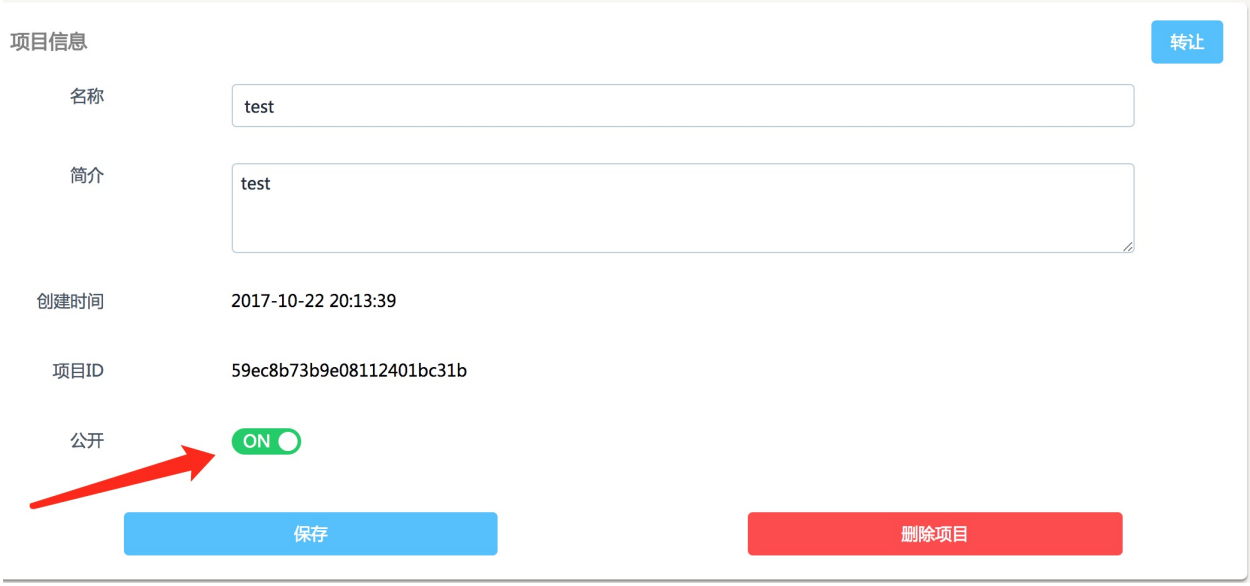


导入成员

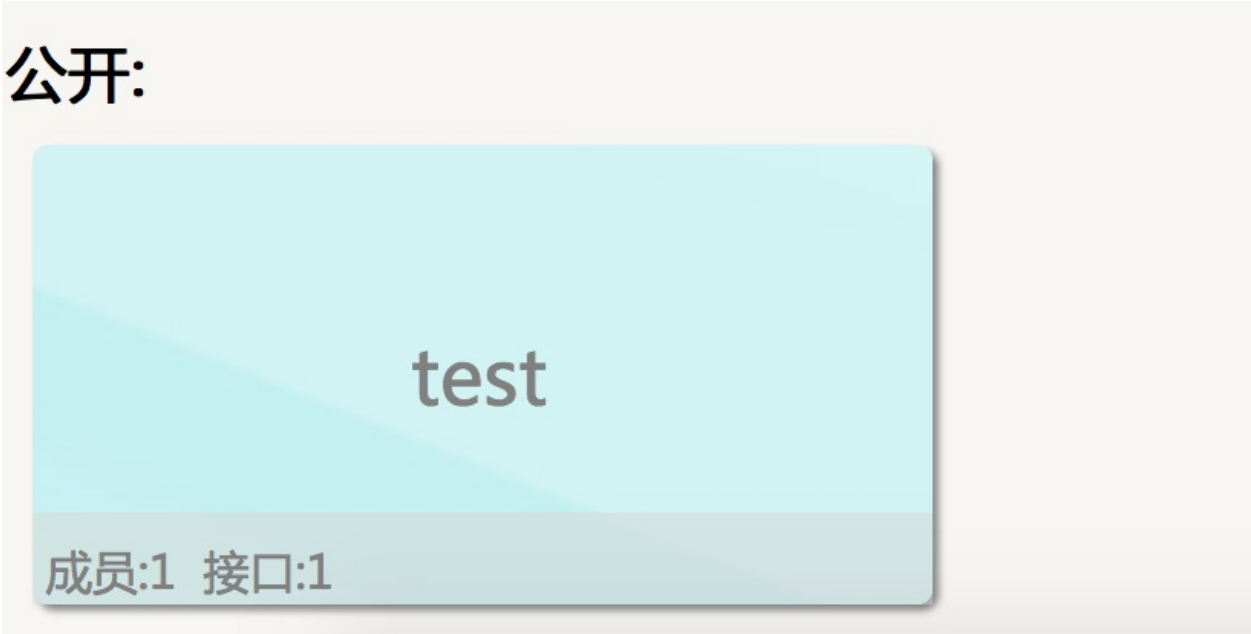
当之前在其他项目有其他成员有协作的时候，我们便可以把这些成员导入到这个项目里面来，点击导入按钮，选择成员和权限，点击确定即可。

公开化

在线下环境中，很多公司有一些公共的项目，里面的接口是大家都可以浏览运行的，如果把公司里面每一个人都加到这个项目中就会很繁琐，这个时候就可以将这个项目公开，那么每一个注册用户都可以在自己的项目列表里看到这个项目，不过对于这个项目的接口只有浏览测试的权限！



上图中把项目公开按钮打开，点击保存，那么这个项目就是一个公开的项目，在其他用户的项目列表里就会有如下图的显示



导出

可以将项目导入成DOClever自己的JSON格式和可以方便离线浏览的html格式

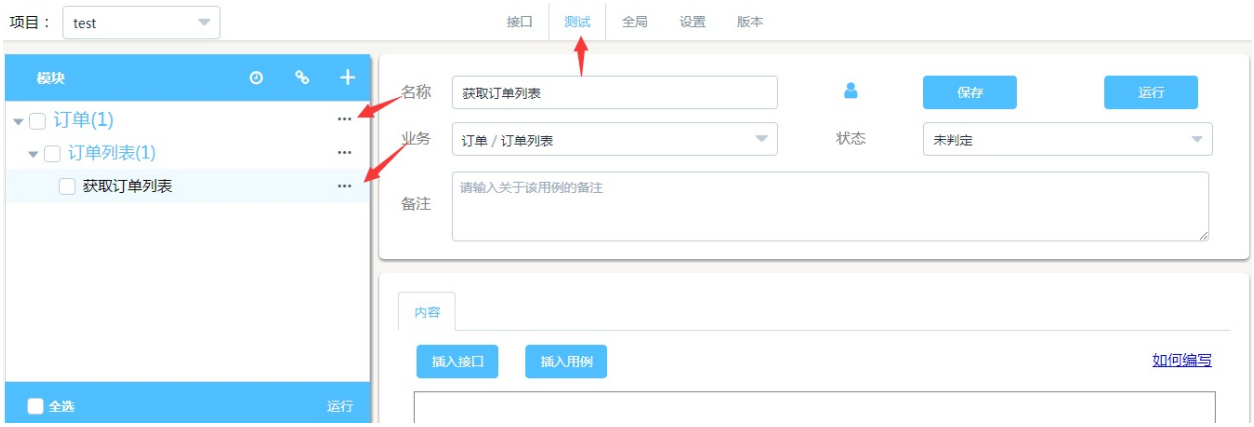
自动化测试

简单测试

简单测试

现在有这样一个业务场景，我需要先登录，然后获取报名订单列表，而如果要登陆的话又必须要获取图片验证码然后把验证码输入到登陆接口的入参里，如果这样一个业务场景由人工手动来做的话是比较繁琐的，接口之前的信息需要来回切换，那么用 DOClever 如何来做自动化测试呢？

> 我们在DOClever里切换到测试栏目下，新建订单模块，然后新建订单列表测试业务，在订单列表下新建一个测试用例，就叫获取订单列表，如下图所示：

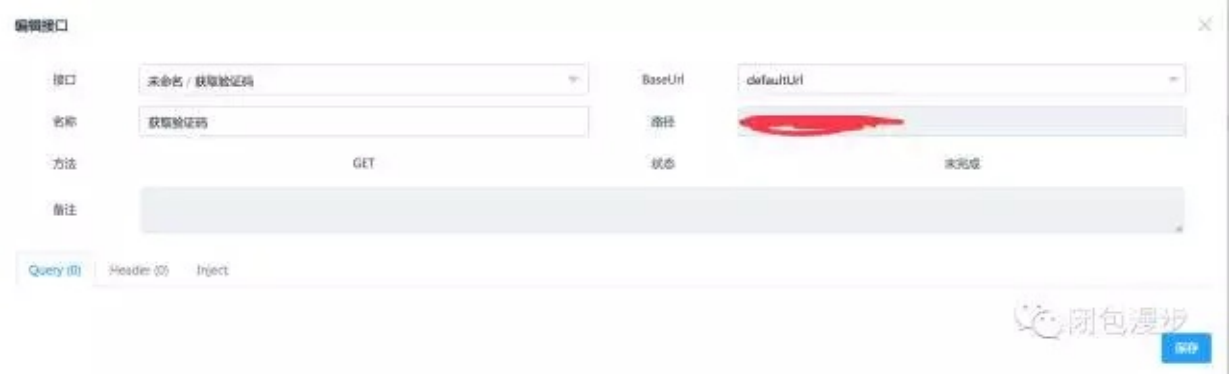


> 然后我们去编写这个测试用例，DOClever 的测试用例用 javascript 编写即可，js 本身还是比较简单，学习一下很快就能上手，如下图：

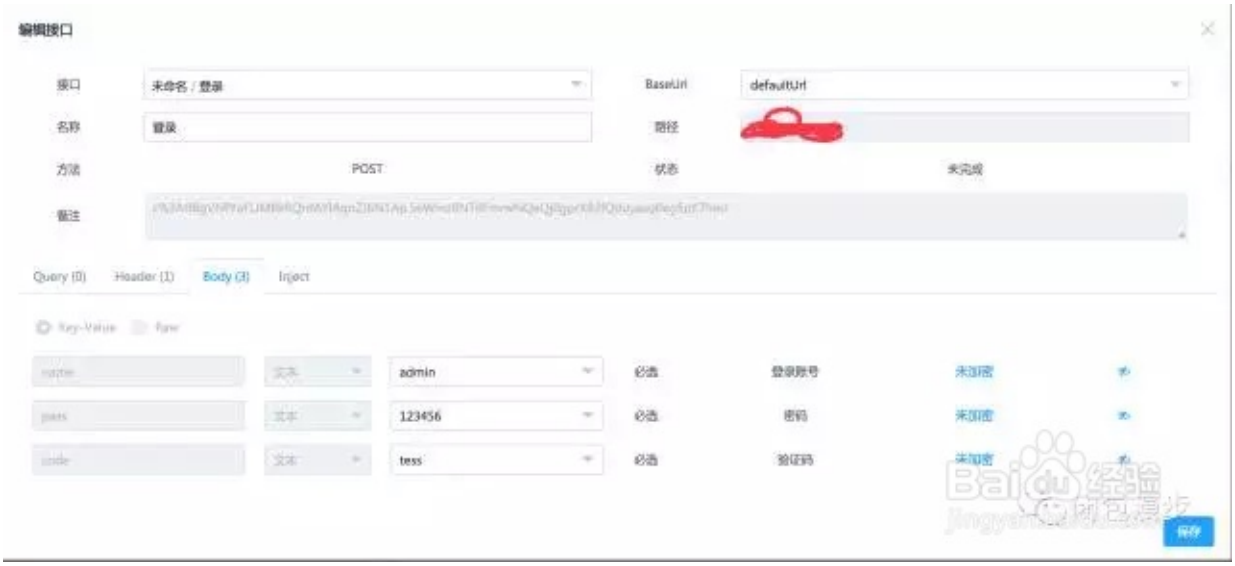
> 代码提取出来如下：

```
var a=获取验证码;
var res=await a();
var text=await input("aaa",res.data);
log(text); a=登录;
var res=await a({ body:{ "code":text } });
var c=报名订单列表;
res=await c();
log(res.data.code);
return true;
```

> 获取验证码，登陆，报名订单列表都是我们点击插入接口按钮插入的接口，插入页面如下：



> 上图中 baseurl 为 default；Url 代表使用默认的 baseurl 运行，否则可以选择我们想要的 baseUrl，query，header 里面如果有参数，我们可以设置参数的值，比如登陆接口的插入页面：




```
var res=await a();
```

> 这一行代表我们执行获取验证码的接口，因为这是一个异步接口，所以需要 es7 里面的新语法 await 来等待这个异步请求的数据返回。

```
var text=await input("aaa",res.data);
```

> res.data 的值就是验证码图形数据，input 函数是一个用户输入的函数，第一个参数是显示的 title，第二个参数是需要在输入框中显示给用户的内容，这里就是把返回的验证码图片展示给用户看，然后用户手动输入验证码，同时程序等待输入返回，返回值赋值给 text 变量。

```
log(text);
```

> log 是一个输出函数，只有一个参数，是需要输出的内容。

```
a=登录;
var res=await a({ body:{ "code":text } });
```

> 这段代码就是调用登陆接口，将 text 的内容作为登陆接口的 body 里面的 code 字段的入参，然后等待接口返回结果。

```
var c=报名订单列表;
res=await c();
log(res.data.code);
return true;
```

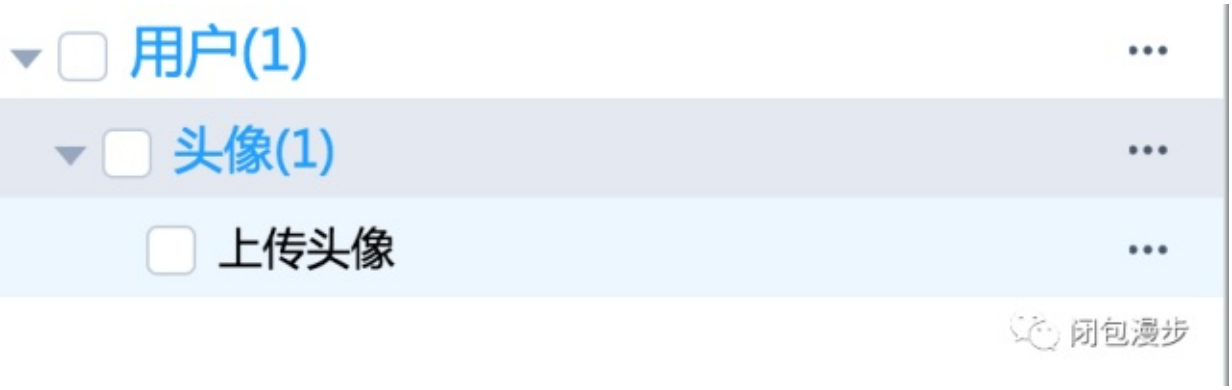
> 这一段代码就是调用报名订单列表，然后将返回数据中的 code 字段打印出来，return true 代表这个测试用例已通过，return false 代表未通过，return 或者没有 return 语句代表这个用例结果未判定。

> 最后，不要忘记在每个语句结束的地方以分号结尾，这个非常重要！ ok，我们写完后还有一件事情不要忘记了，就是设置 defaultUrl，它会作为每个接口的 baseUrl 来运行接口，我们点击这个按钮即可选择：

> 设置完成后，点击用例编辑页面的运行按钮，会出现下图所示：

> 这就是 input 弹出的输入框，图片展示的我们请求验证码接口返回的验证码图片，我们输入 pppx 验证码，然后点击确定，测试会自动按照流程走下去，当弹出运行完成的提示框时，我们看下输出标签页：

> 整个用例的详细输出信息都会完整的打印出来。 那么如果接口需要文件上传，DOClever 可不可以实现呢，答案完全是可以的，我们新建一个用例：



> 然后编辑测试用例的代码：

```
var a=上传头像;
var res=await a();
log(res.data.data);
return true;
```

> ok，点击运行，便会弹出一个文件选择页面，如下图：

> 我们选择需要上传的图片，点击确定，等弹出运行完成提示框，我们看下输出标签页的内容显示：

> 已经执行成功，同时也打印出了新上传图片的路径地址！

高级测试

> DOClever允许你用javascript代码来编写你的测试用例,举例

```
var a=获取图片信息;
var  res=await a({
    query:{
        a:123
    }
});
global["sx"]="ddd";
var gg=登录;
await gg();
var b=info;
res=await b({
    query:{
        sx:"ddd"
    }
});
var text=await input("请输入值",res.data)
log(res.status);
return true;
```

> 上述代码为一个简单的用例，注意每个语句都需要用分号来结尾,蓝色的内容代表你插入的接口，橘黄色的内容代表你插入的用例，执行一个接口或者用例需要用await来等待数据返回，并且我们还可以在执行接口的时候动态的传入接口参数数据:

param:restful中的param对象
query:query参数对象
header:header参数对象
body:body参数

> 这里可以是一个对象或者字符串，字符串代表接口会把这个字符串当做整个body发送出去，如果是对象的话，有两种方式：
> 1、key-value类型
> 2、json类型
> 当类型为json的时候，支持层级之间用"."来进行分隔，比如：

```
body:{ "a.b.c":"aaa"}
```

> 这个就代表body["a"]["b"]["c"]的数据为aaa，如果路径不存在，会自动添加返回值：这里return true

> 代码当前的用例执行成功，如果是false代表失败，如果直接return或者没有return代表当前的用例结果未判定。

内置变量：

> input:输入框
> 第一个参数是给用户的提示信息
> 第二个参数是展示给用户的数据

>global:全局对象，可用于在不同的用例之间传输数据

> log:输出函数，只有一个参数，为需要输出的数据

> Base64、MD5、SHA1、SHA256、SHA512、SHA3、RIPEMD160
> 这些加密函数只有一个参数，为加密的字符串

> AES、TripleDES、DES、Rabbit、RC4、RC4Drop
> >这些加密函数有两个参数：
> >第一个参数是加密的字符串
> 第二个参数是salt

组合测试

> 那么不同的测试用例之间能不能联调呢，是完全可以的，我们可以在一个用例里面插入另一个用例，然后用内部对象 global 来在用例之间传递数据，比如我们可以把简单测试中的获取订单列表这个用例改写下：

```
global["name"]="sx";
var c=上传头像;
res=await c();
return true;
```

> 把上传头像这个用例也改写下：

```
var a=上传头像;
var res=await a();
log(global["name"]);
return true;
```

> 然后运行获取订单列表这个用例，输入如下所示：

> 说明用例是可以嵌套运行的，并且可以传递数据。 DOClever 还可以批量运行测试用例，如下图：

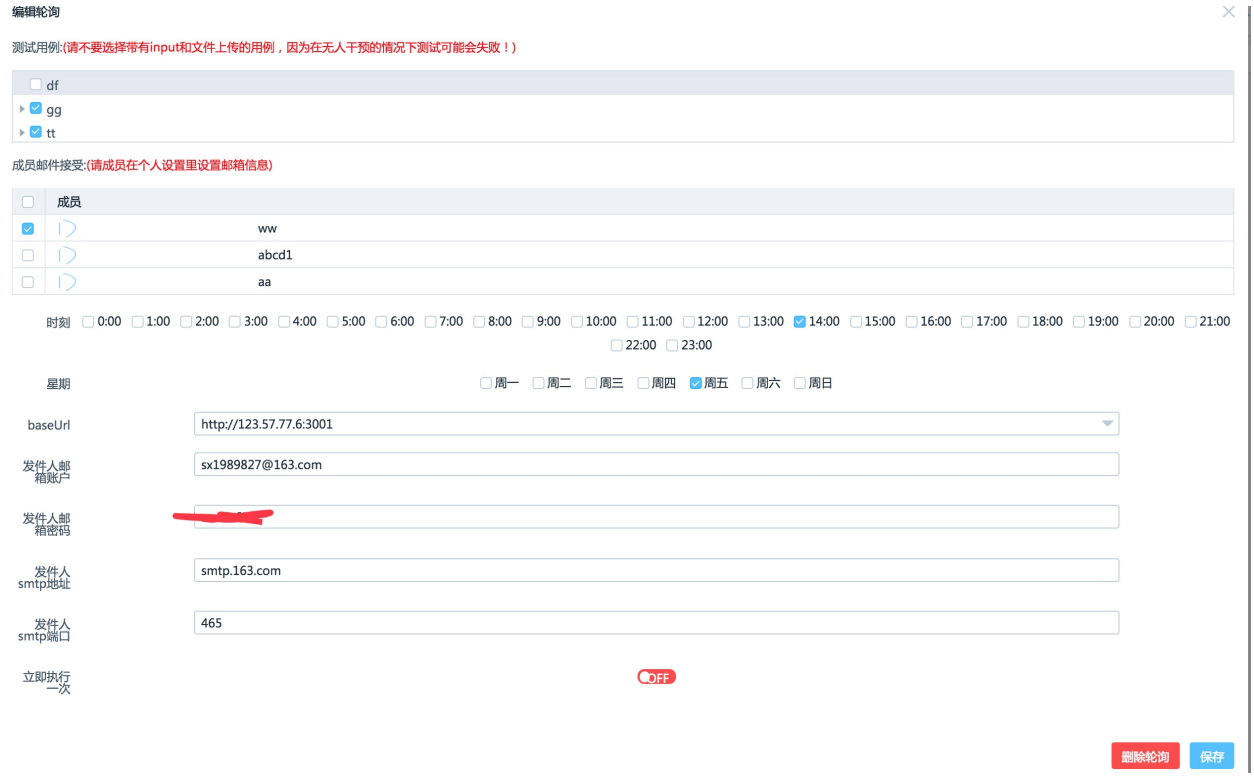
> 如上图所示，勾选你想要运行的测试用例，然后点击运行，便可以批量的运行测试用例，且测试用例的返回状态和输出都会实时的保存起来并显示，如下图：

这样无论是对于前后端开发，还是测试人员，都非常方便做回归测试啦！

后台轮询

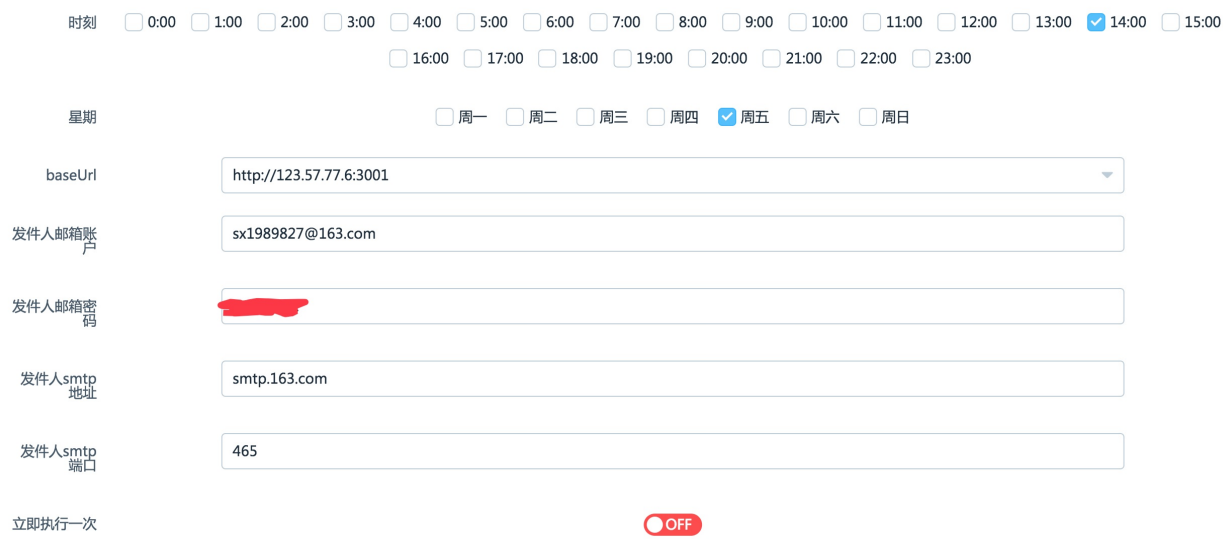
DOClever还提供了在后台定时自动运行测试用例的功能，并且可以把运行接口发送到指定成员的邮箱！

点击上图所示按钮，出现如下界面



我们现在需要运行测试用例，注意，这里的用例不能是带有input和图片上传的用例，因为没有外界的手动干预，这些用例是会失败的。

然后选择需要发送邮件的成员，这里的成员需要在个人设置里面设置好自己的邮箱



上图中有两点需要注意，第一是发件人的邮箱密码，很多邮箱做了权限加固，所有有时候密码不一定可以，需要邮箱的授权码，第二是立即执行一次，这里的意思是如果保存，那么这个轮询会立即执行一次，这样可以验证设置的内容是否正确！

版本管理

项目版本

项目创建版本
默认开发的版本为主板本,类似在git上打一个分支。打版本步骤见下图
1.建立一个版本

2.对当前版本进行编辑

首先是编辑，顾名思义 就是对当前的版本的信息进行编辑

3.切换

意思就是切换到当前的分支上

在当前的分支上做开发

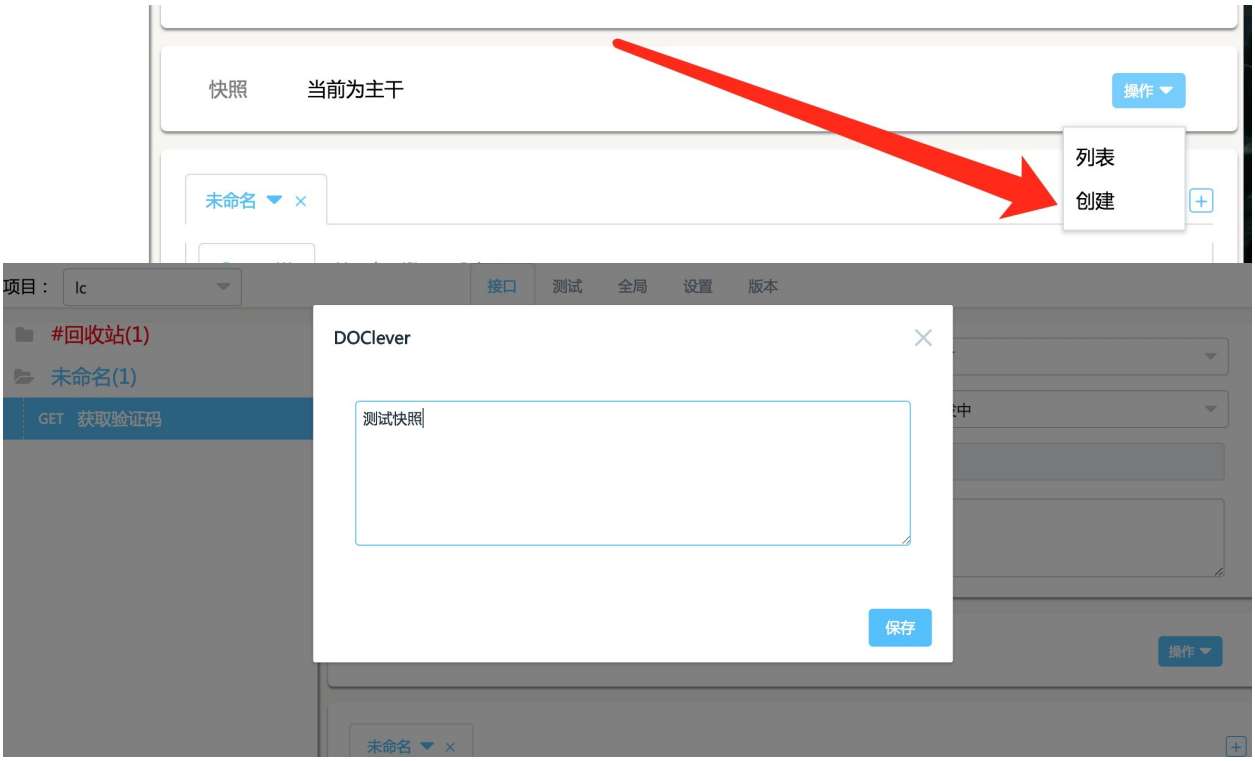
4.回滚

回滚的意思是当前分支回滚到最后一次保存这个版本的时候，而在最后一次保存这个版本的时间点到现在回滚时间段内开发的东西，会添加到主分支上。

5.删除就是删除当前版本。

接口快照

快照类似上面的项目版本
给当前接口打一个快照



查看当前接口的快照列表

某一个快照说明:

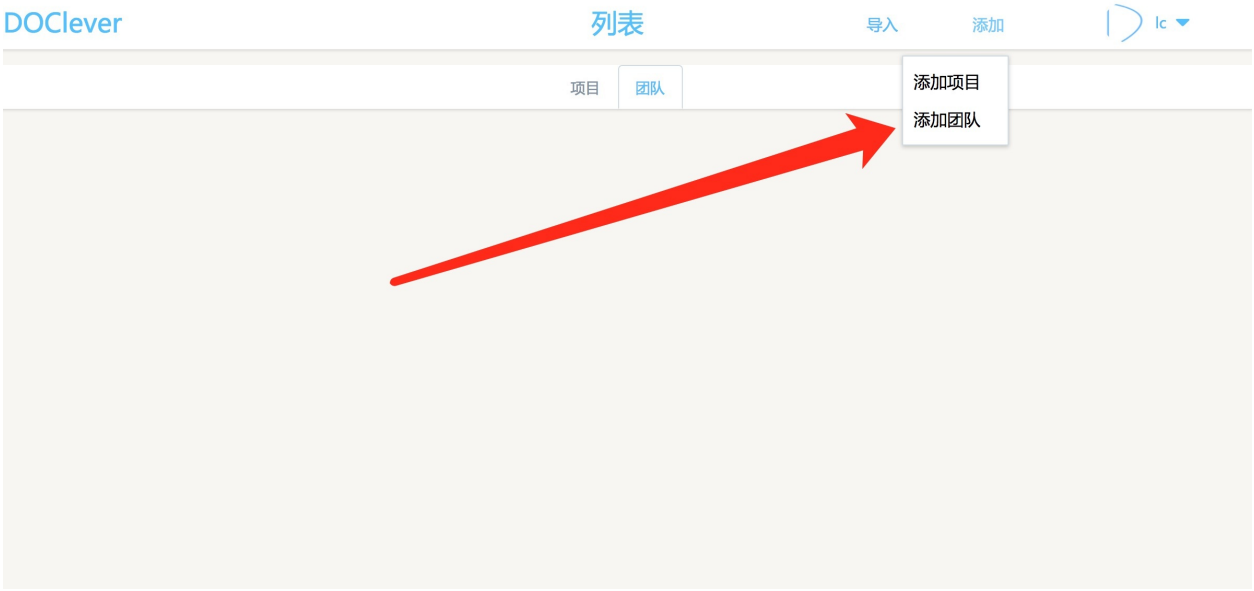
切换：就是将当前的接口版本切换到当前的快照版本。
回滚：就是在当前快照最后一次保存的时间点到现在回滚的时间段内接口的变动会添加到主快照上面（就是主分枝）
删除：就是删除当前的快照

团队协作

创建团队

如何创建自己的团队？

第一步：添加-->添加团队



第二步：填写团队

填写团队名称以及描述并确定

用户主动申请加入某个团队 团队id可以在团队概况中看到

分配项目

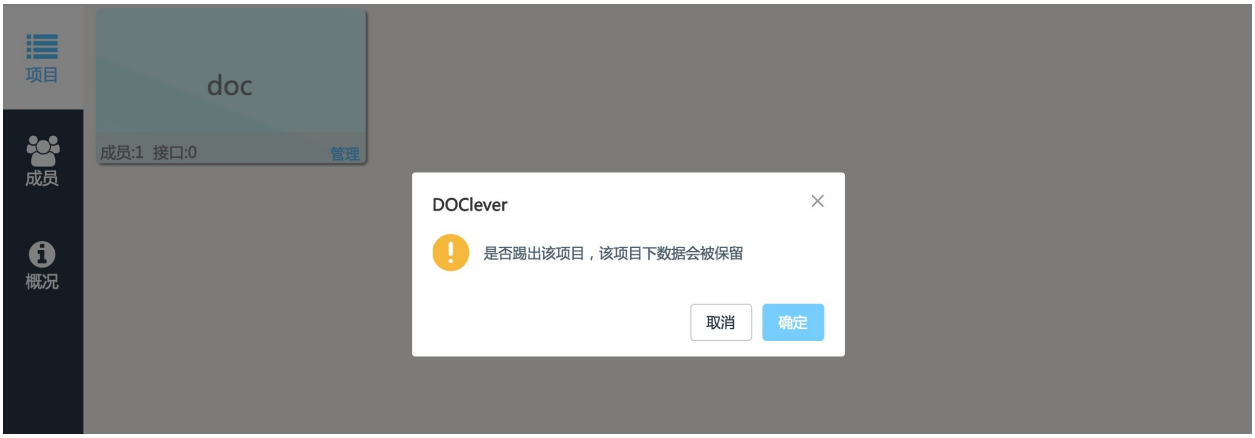
项目管理

1.项目管理有成员管理，踢出团队，删除项目以及指定所有者，成员管理可以指定项目内某个成员的权限，踢出团队就是将项目踢出这个团队，项目本身还是存在的。删除项目就是这个项目完全删除，包括数据。指定所有者就是指定这个项目的所有者。默认项目的创建者就是这个项目的所有者。

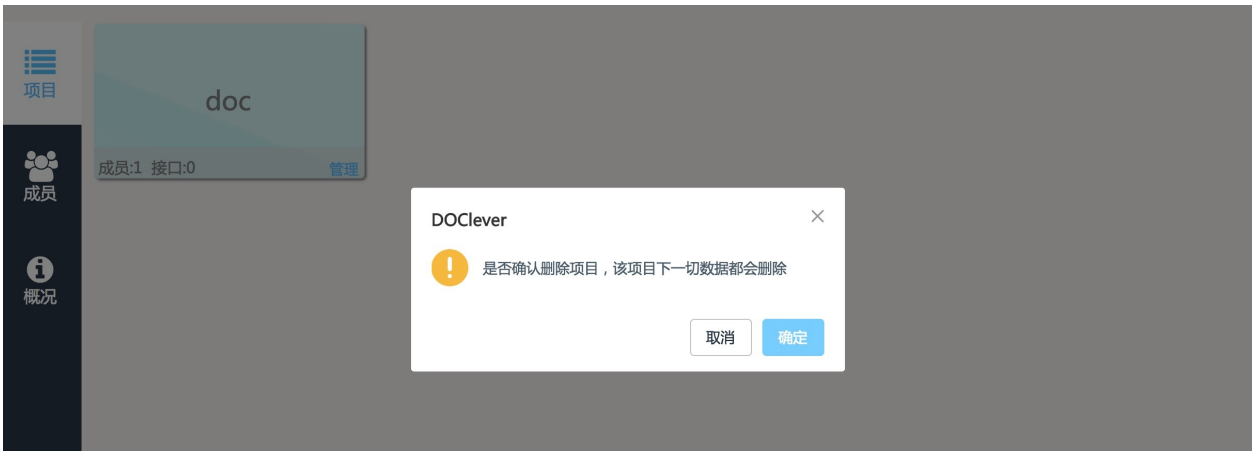
2.项目成员管理

点击权限可以分配某个账号的具体权限

3.项目踢出团队



4.删除项目



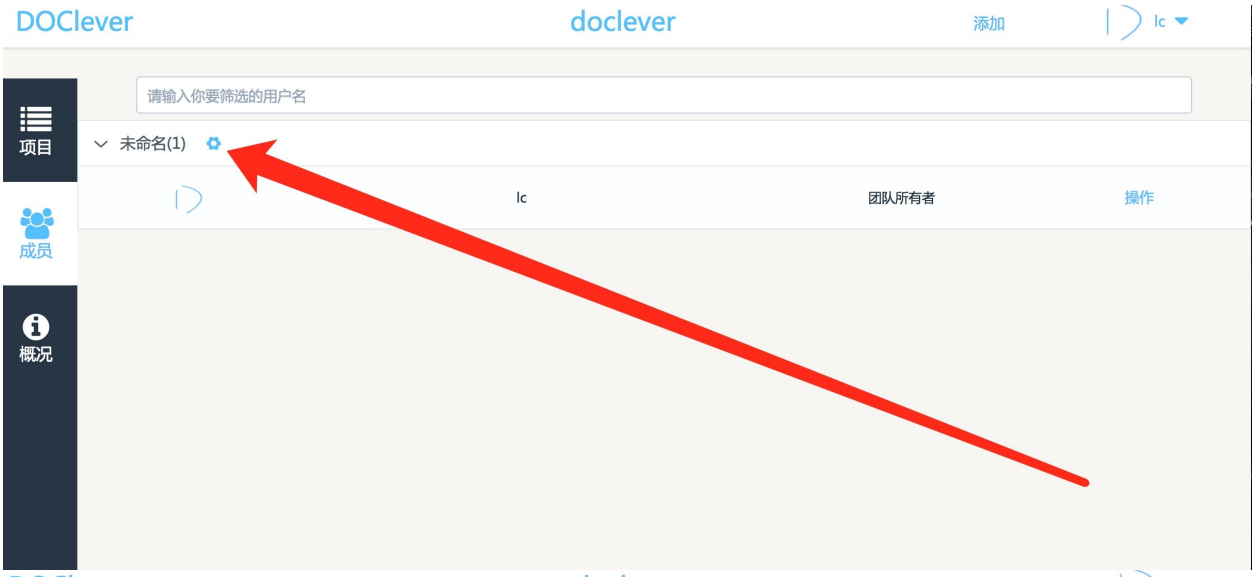
5.指定所有者

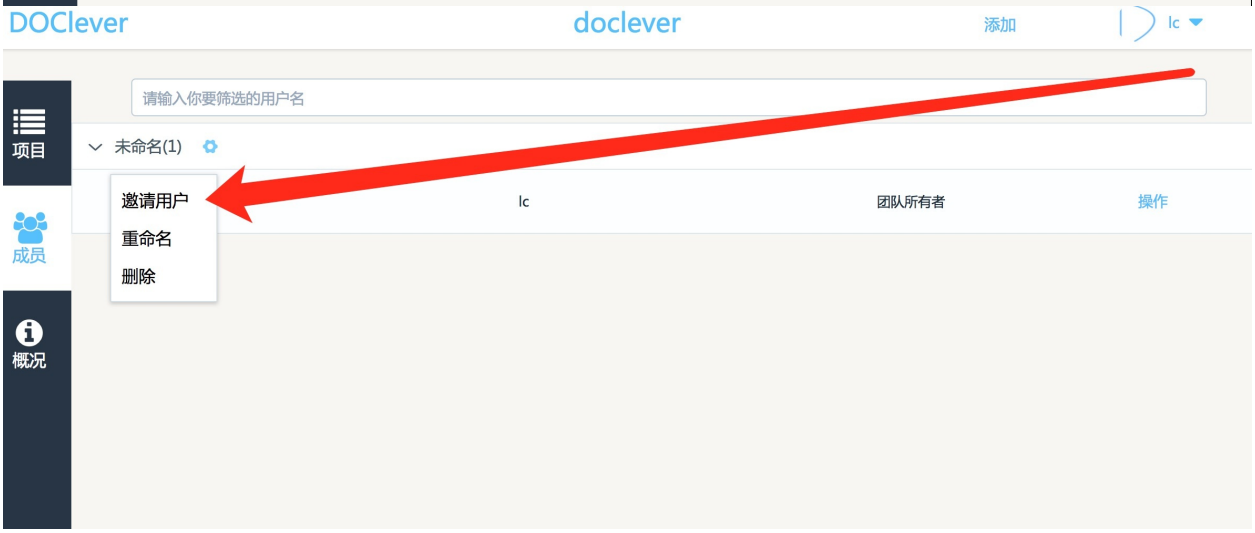
将项目指定团队内的某一个成员成为此项目的所有者

管理组员

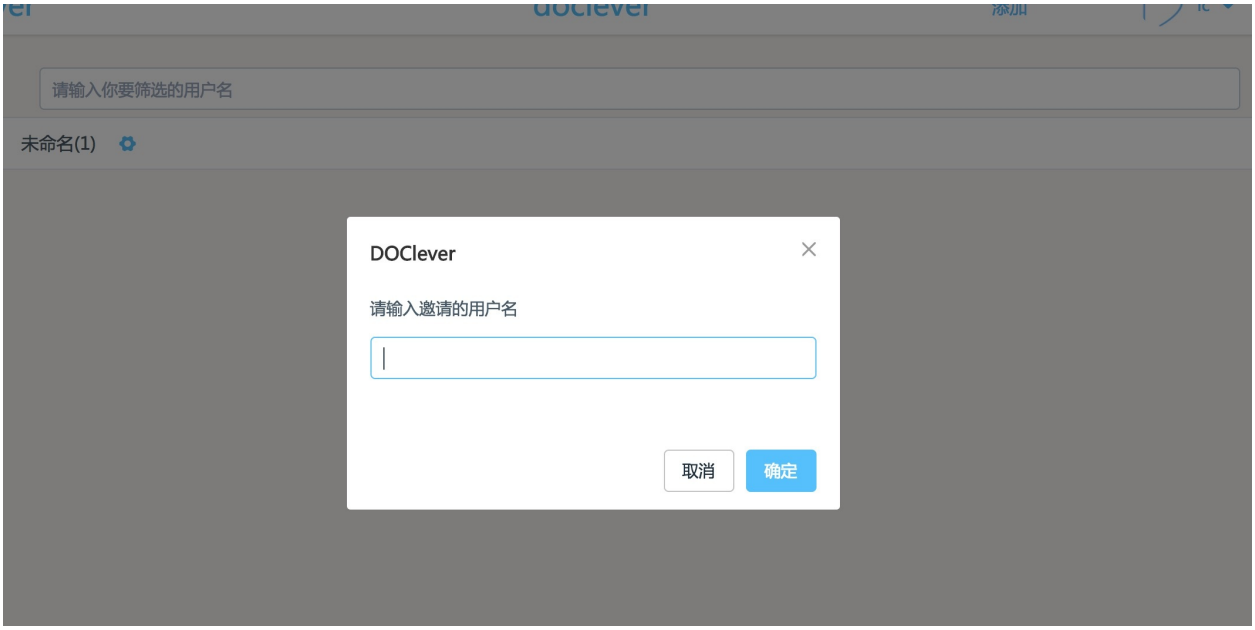
团队添加组员

第一步:邀请用户





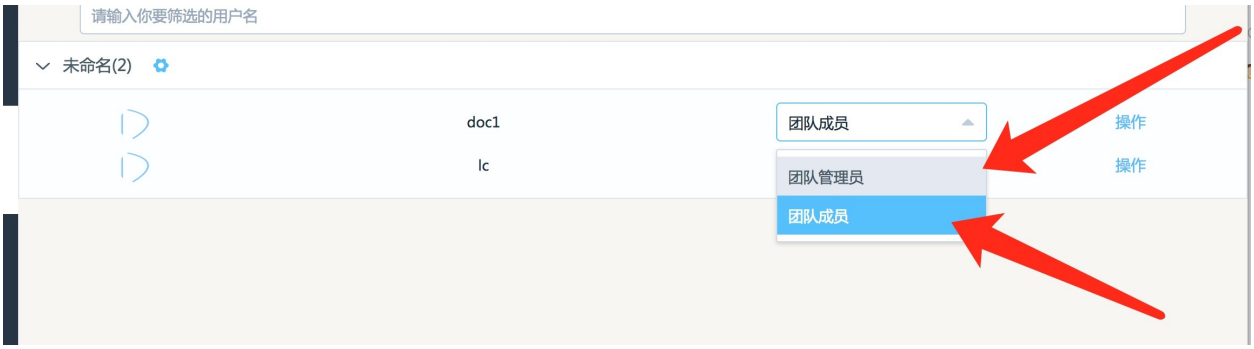
第二步:输入邀请用户的用户名



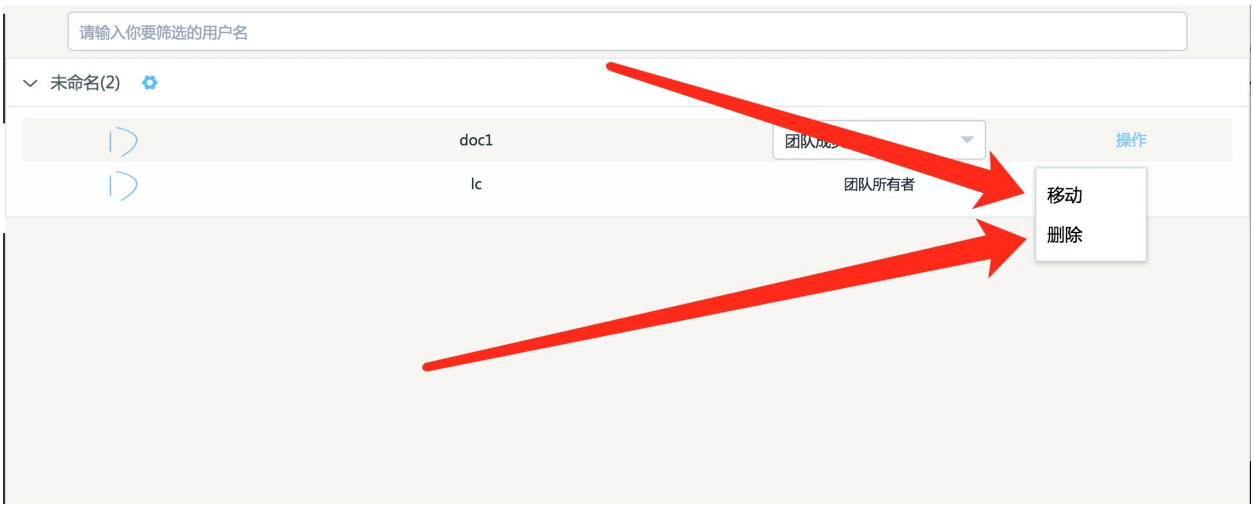
第三部:被邀请用户刷新界面或者重新登录会收到邀请信息

团队成员权限管理

1.被邀请的用户默认在团队里是普通成员，成员所有者可以提升某个用户的权限或者降级某个用户的权限。

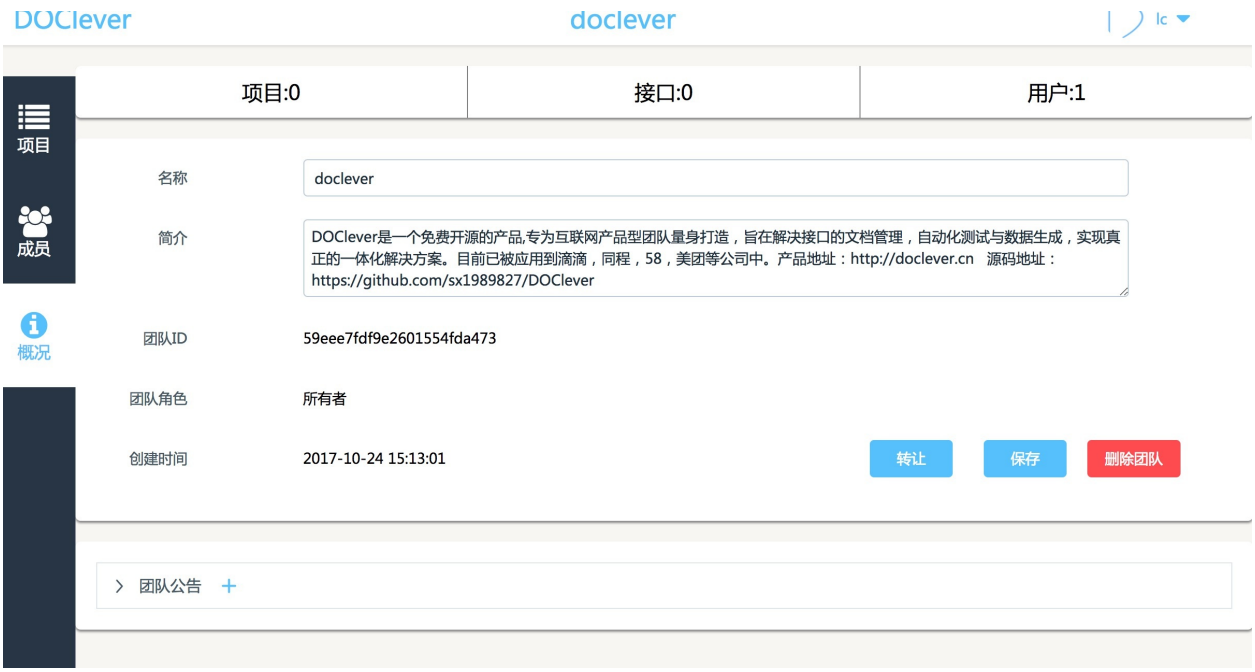


2.某一个用户被提升为管理员，这个用户既可以对普通用户做移除团队以及移动的权限。

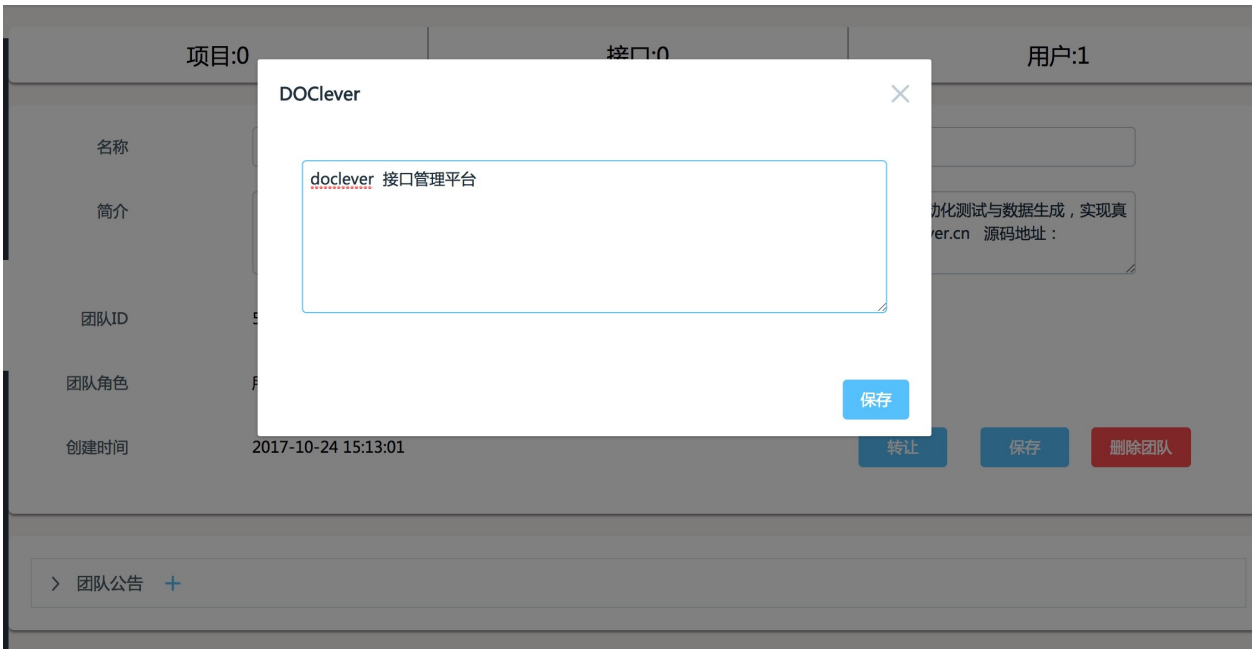


团队概况查看

1.概况可以看到这个团队目前的 项目数 接口数 以及用户数。项目的名称，简介，团队的ID,所有者以及创建的时间，同时可以修改团队的信息以及转让团队和删除团队。



2.添加团队公告 左下角的+号

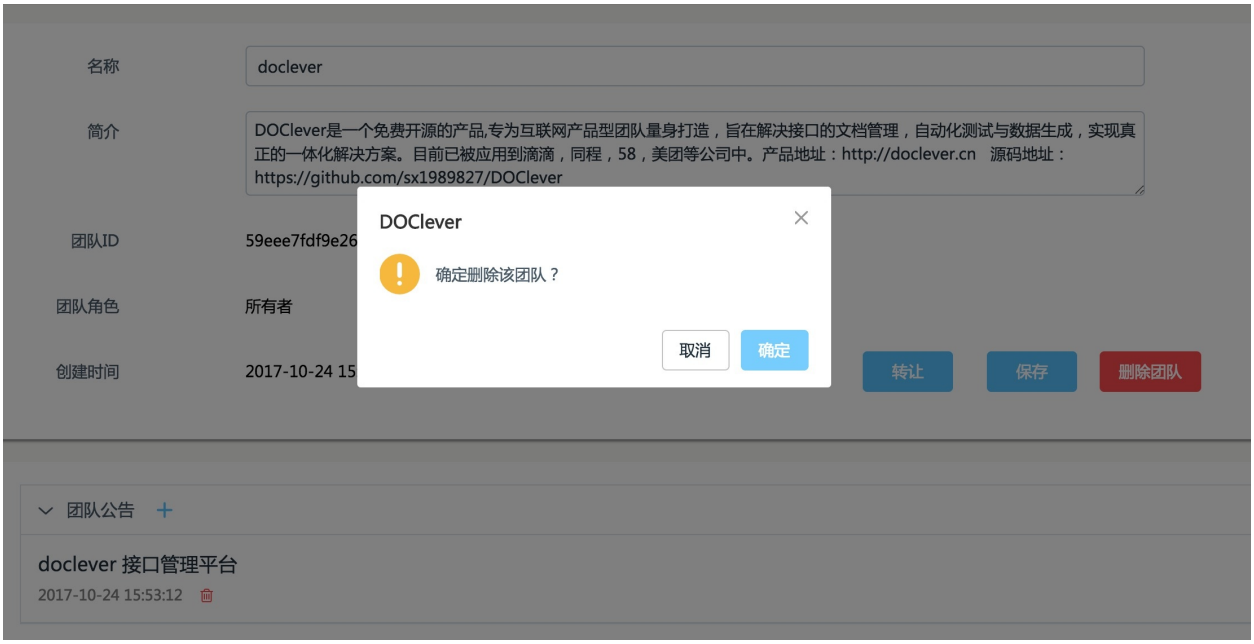


3.团队转让

团队转让只能转让给团队内的成员，不可以转让给团队以外的用户。

4.删除团队

删除团队就是将这个团队解散，团队的项目不会被删除，团队的项目还会在对应的归属者下面。



团队项目

在团队中创建项目

1.导入已有的项目

2.新建项目

总后台管理

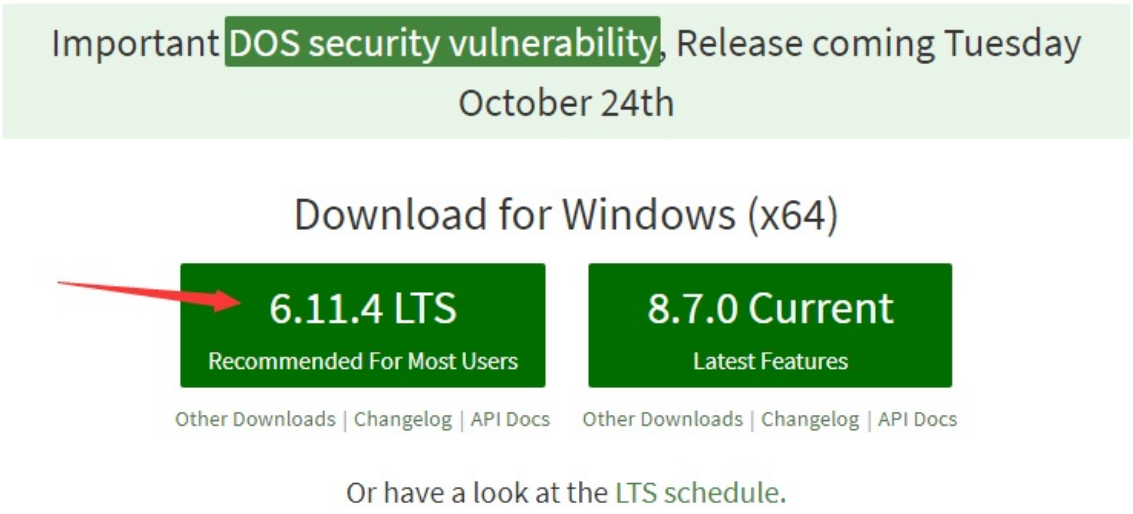
DOClever的线下部署拥有总后台管理功能，可以管理所有用户，项目和团队

默认账号密码都是DOClever

线下部署

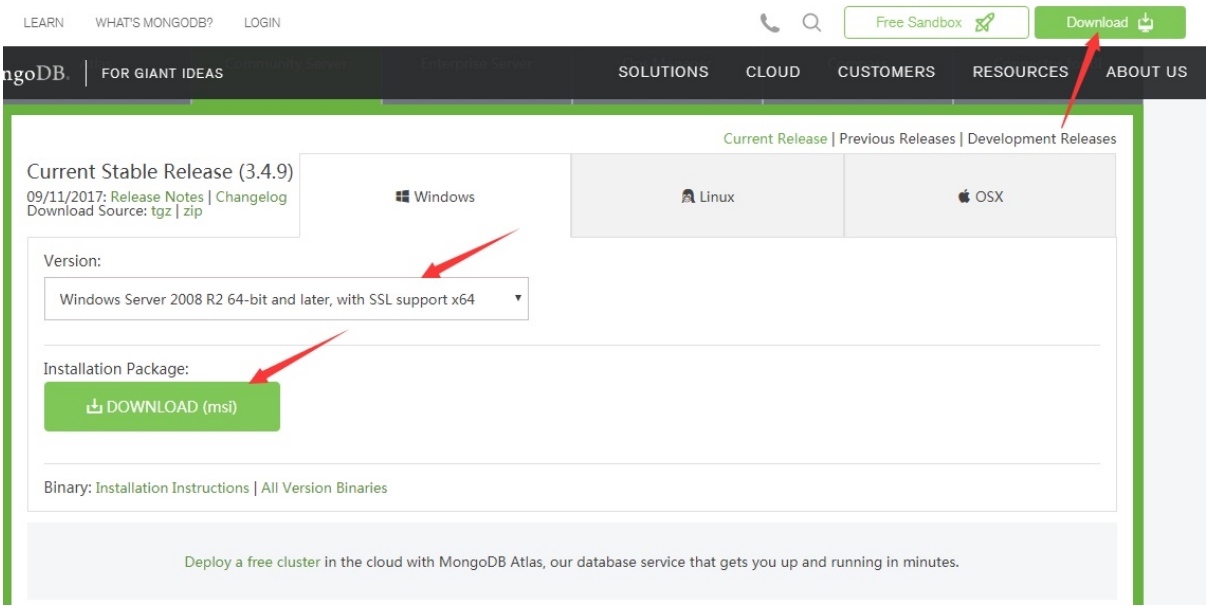
windows

> 首先本地要安装node环境，推荐6.10.0版本，下载页面

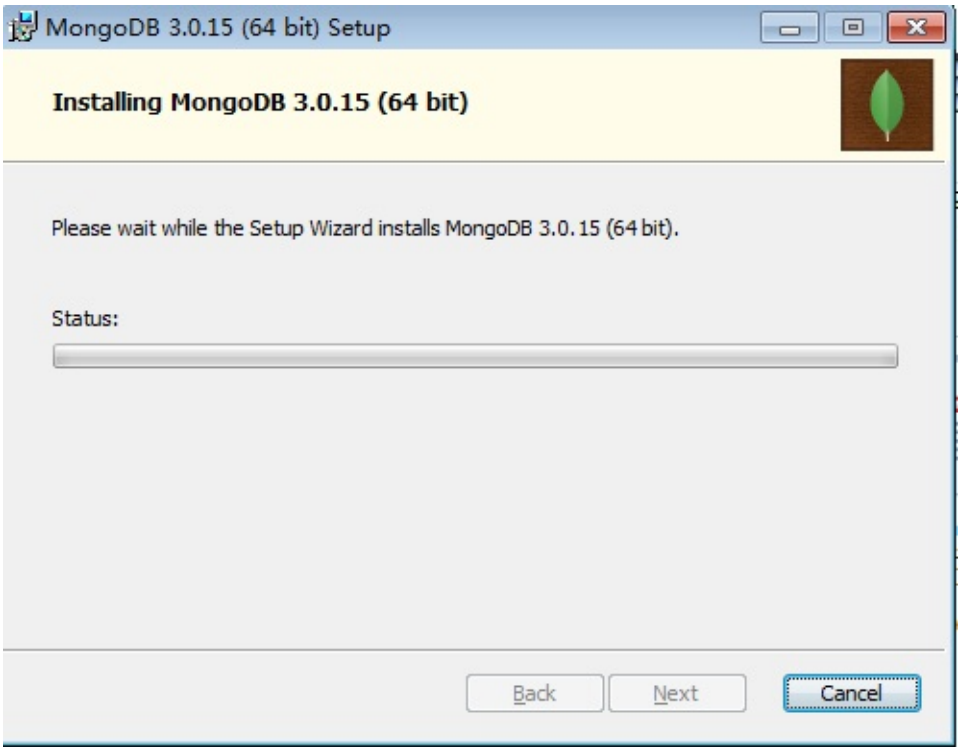


> 下载完成后双击进行安装，安装完成后此时Node环境就已经安装成功；

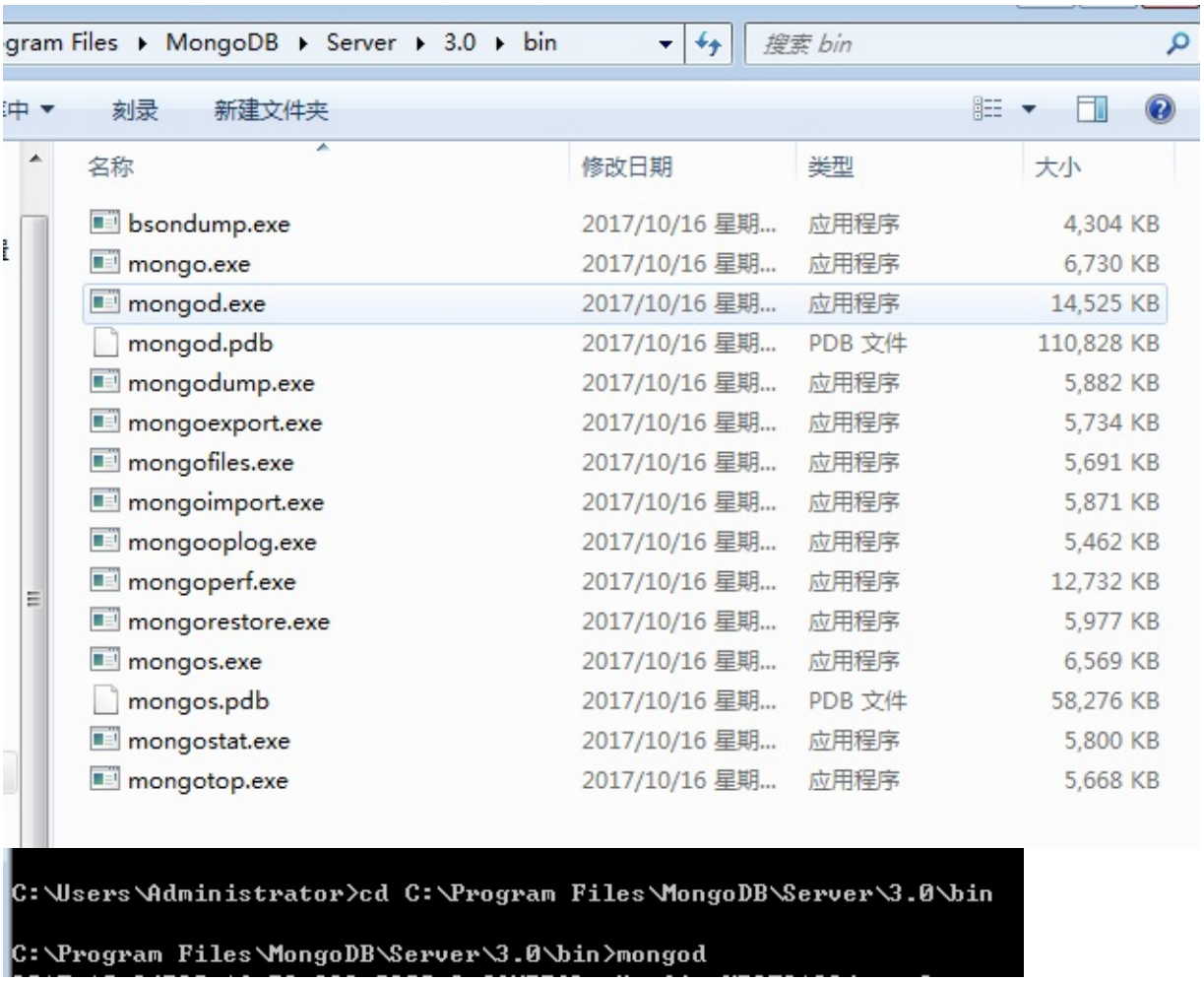
接下来我们开始安装mongodb(下载页面)；如下图所示：

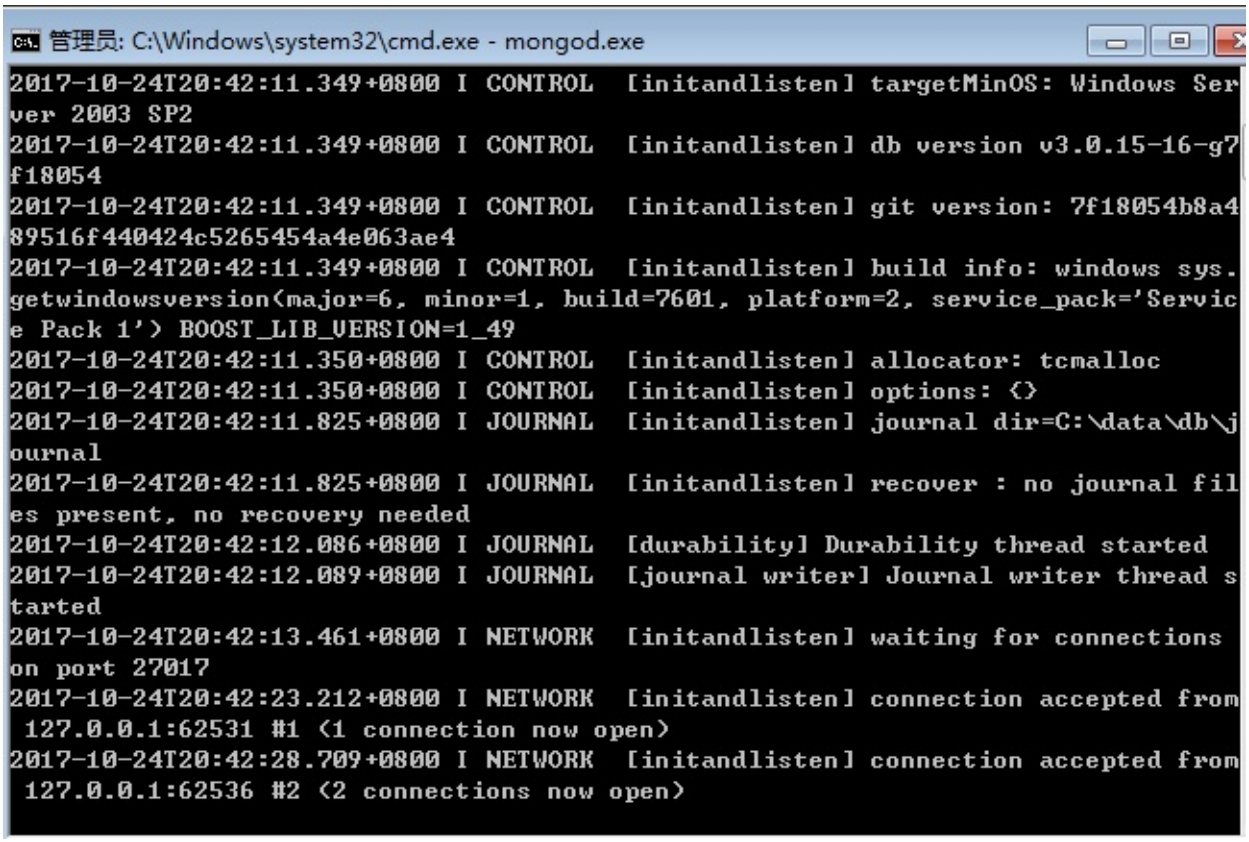


> 选择合适的windows版本进行下载

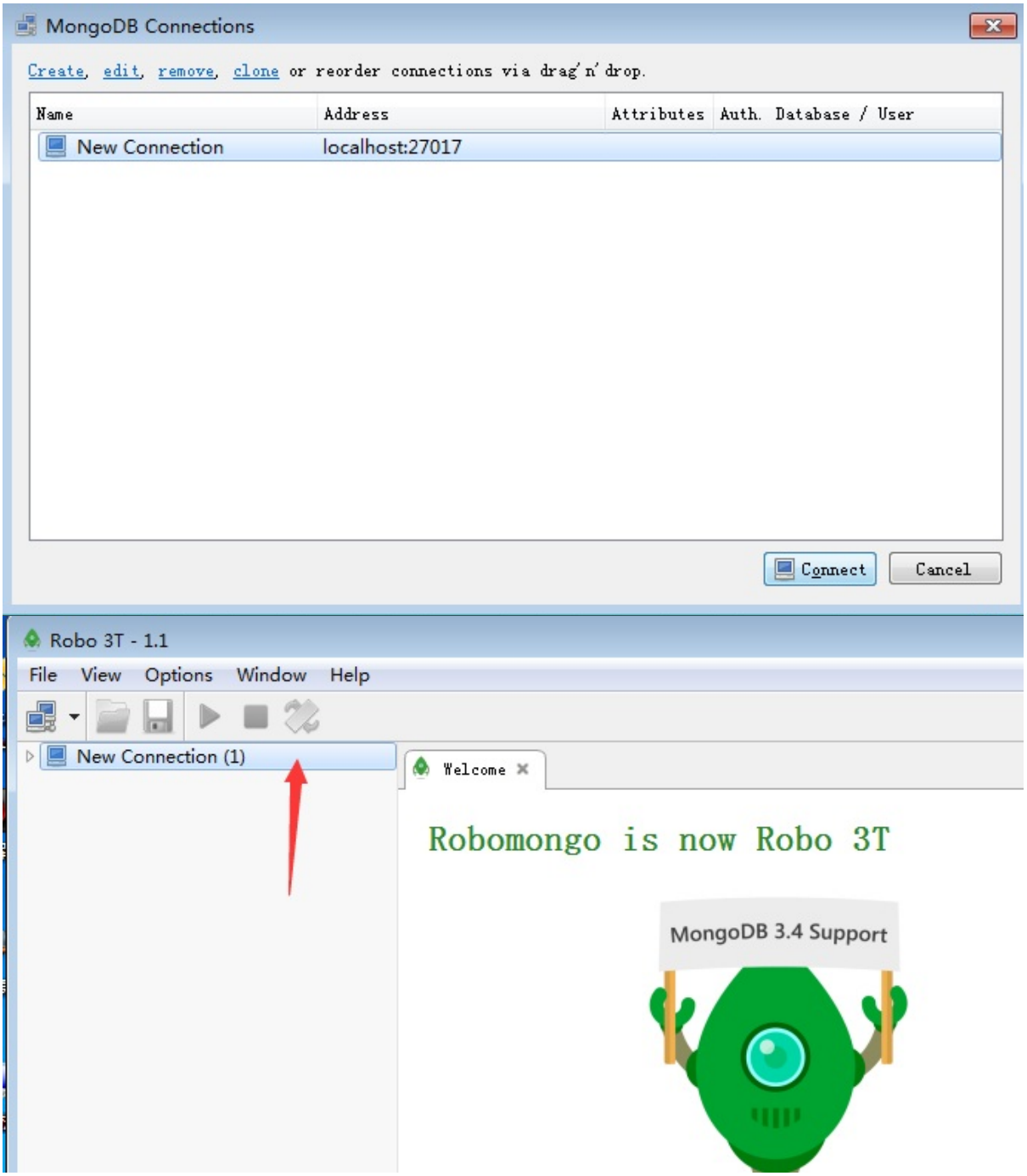


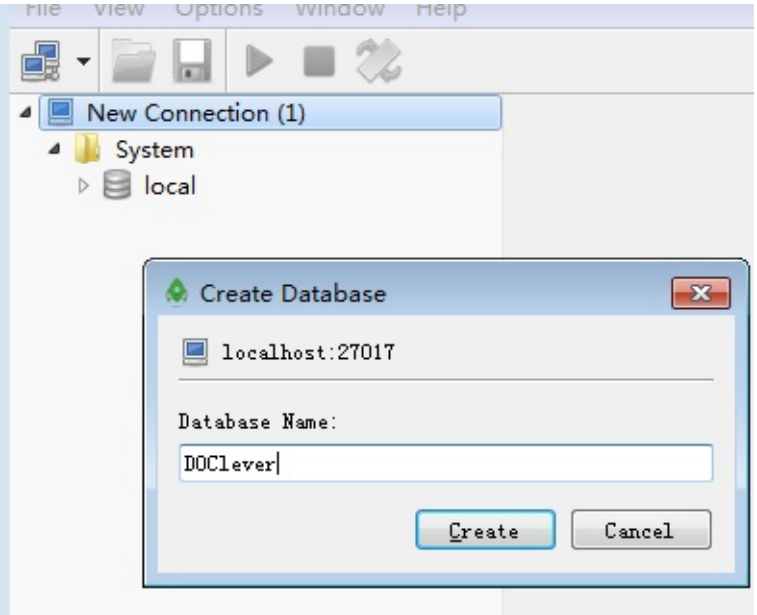
下载完成后我们一路 next进行安装即可，安装完成后，我们需要启动mongodb；


gram Files ▸ MongoDB ▸ Server ▸ 3.0 ▸ bin
搜索 bin
刻录
新建文件夹
名称
修改日期
类型
大小
bsondump.exe 2017/10/16 星期... 应用程序 4,304 KB
mongo.exe 2017/10/16 星期... 应用程序 6,730 KB
mongod.exe 2017/10/16 星期... 应用程序 14,525 KB
mongod.pdb 2017/10/16 星期... PDB 文件 110,828 KB
mongodump.exe 2017/10/16 星期... 应用程序 5,882 KB
mongoexport.exe 2017/10/16 星期... 应用程序 5,734 KB
mongofiles.exe 2017/10/16 星期... 应用程序 5,691 KB
mongoimport.exe 2017/10/16 星期... 应用程序 5,871 KB
mongooplog.exe 2017/10/16 星期... 应用程序 5,462 KB
mongoperf.exe 2017/10/16 星期... 应用程序 12,732 KB
mongorestore.exe 2017/10/16 星期... 应用程序 5,977 KB
mongos.exe 2017/10/16 星期... 应用程序 6,569 KB
mongos.pdb 2017/10/16 星期... PDB 文件 58,276 KB
mongostat.exe 2017/10/16 星期... 应用程序 5,800 KB
mongotop.exe 2017/10/16 星期... 应用程序 5,668 KB
C:\Users\Administrator>cd C:\Program Files\MongoDB\Server\3.0\bin
C:\Program Files\MongoDB\Server\3.0\bin>mongod



> 启动mongodb后我们可使用robomongo来作为mongodb的客户端工具(下载页面)
> >下载安装完成后我们用robomongo来连接，新建一个database作为DOClever的数据库（名称随意）


MongoDB Connections
Create, edit, remove, clone or reorder connections via drag'n'drop.
Name
Address
Attributes
Auth. Database / User
New Connection
localhost:27017
Connect
Cancel
Robo 3T - 1.1
File
View
Options
Window
Help
New Connection (1)
Welcome
Robomongo is now Robo 3T
MongoDB 3.4 Support



新建个数据库命名为“DOClever”

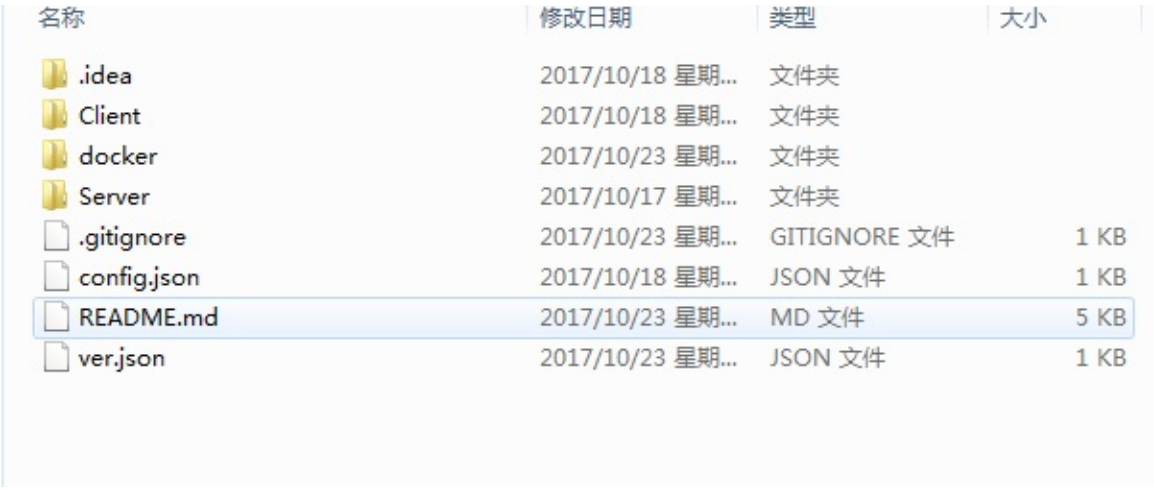
至此mongodb启动配置完成！

> 注意：前台启动mongodb的方式，不可关闭dos命令窗口界面
>
> 【以Windows Service的方式启动MongoDB】 点击此处查看

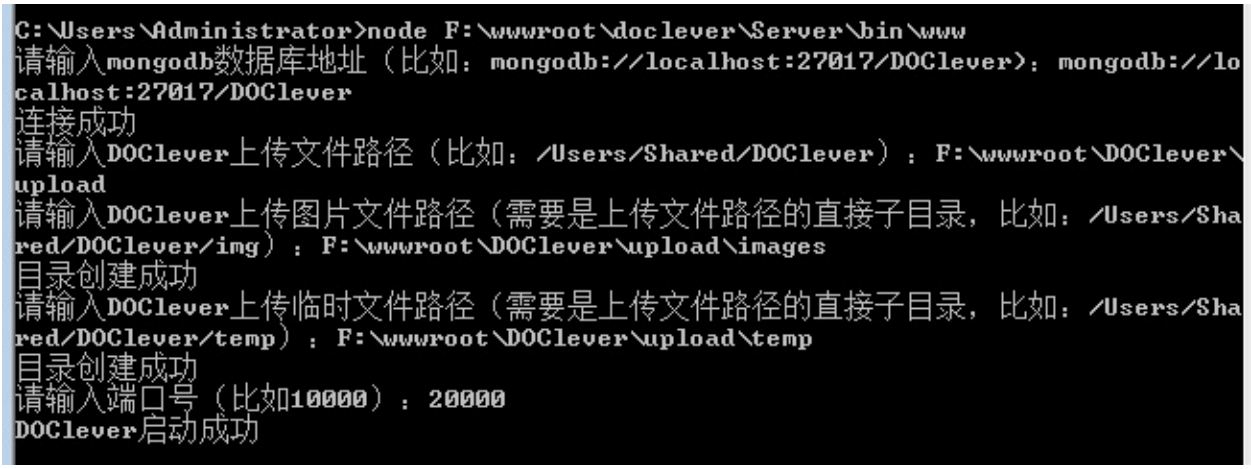
接下来我们开始启动DOClever:

> 下载DOClever源代码，这里以V4.1.3为例，下载地址：
> 码云下载
> GitHub下载

> 这里我把下载下载好的源文件解压到F:\wwwroot\doclever根目录下面；（记住这个目录，一会配置的时候要用到！）



在命令行下运行node DOClever的根目录:（上面我们提到的解压地址在F:\wwwroot\doclever目录下）\Server\bin\www（注意斜杠）



启动成功后，在浏览器输入http://localhost:20000/（这里20000改为你设置的端口号）

至此部署成功！！

linux(mac)

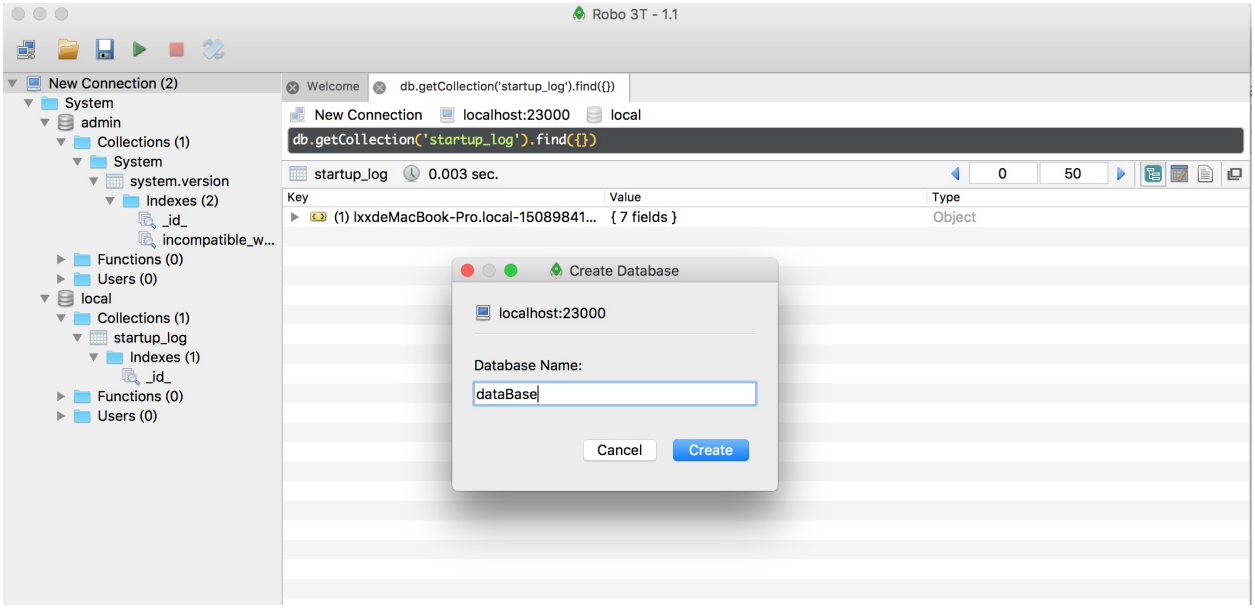
线下部署(以mac为例)

1.首先本地要安装node环境，推荐6.10.0版本下载页面

```
安装node环境
```

2.安装mongodb下载页面
3.可使用robomongo来作为mongodb的客户端工具下载页面

```
下载安装完成后我们用robomongo来连接，新建一个database作为DOClever的数据库（名称随意）
```



4.启动mongodb

```
启动Mongodb服务有两种方式，前台启动或者Daemon方式启动，前者启动会需要保持当前Session不能被关闭，后者可以作为系统的fork进程执行，下文中的path是mongodb部署的实际地址。

1. 最简单的启动方式，前台启动，仅指定数据目录，并且使用默认的27107端口，cli下可以直接使用./mongo连上本机的mongodb，一般只用于临时的开发测试。
```

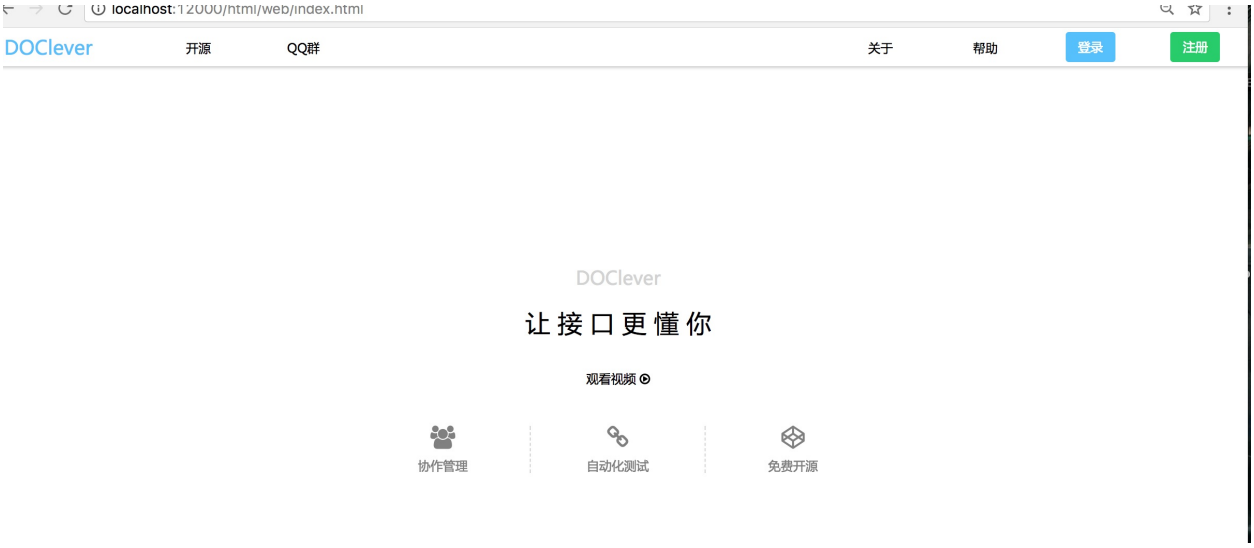
./mongod --dbpath=/path/mongodb

步骤见如下图

```
 以下是启动mongod
![1](images/45B31463-C20D-4B0C-B97D-C31BEB34F872.png)
![2](images/58055CA8-B337-4FFF-9A98-0E7FCEDBAF26.png)
以下是.将DOClever的源码down到本地，在命令行下运行node DOClever的根目录/Server/bin/www（如果是windows环境下，请修改目录分隔符），第一次启动，会出现命令行提示符，按照提示符输入即可完成相关的配置，等到DOClever启动成功后， 在浏览器里输入localhost:DOClever启动的端口号,出现首页表示部署成功。
```

最后在浏览器中输入之前的端口号

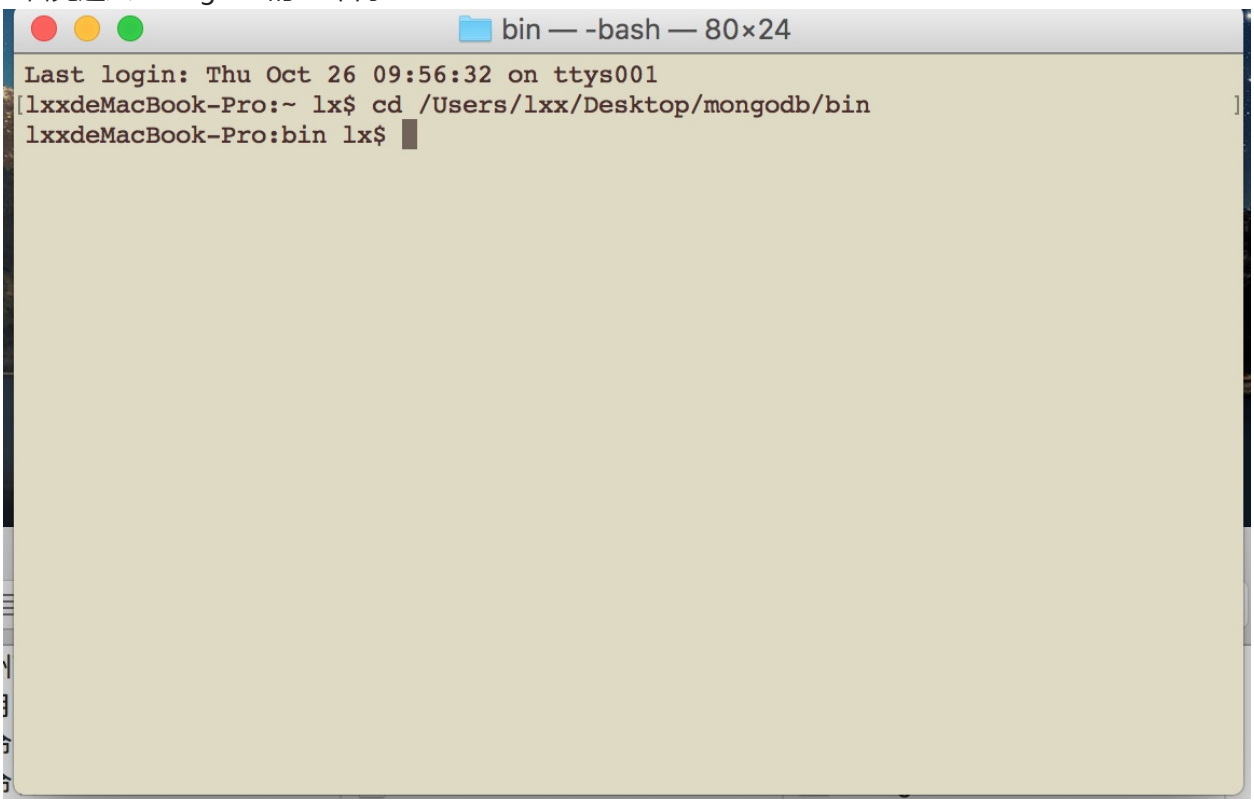
localhost:12000



这样就成功了

2.fork启动

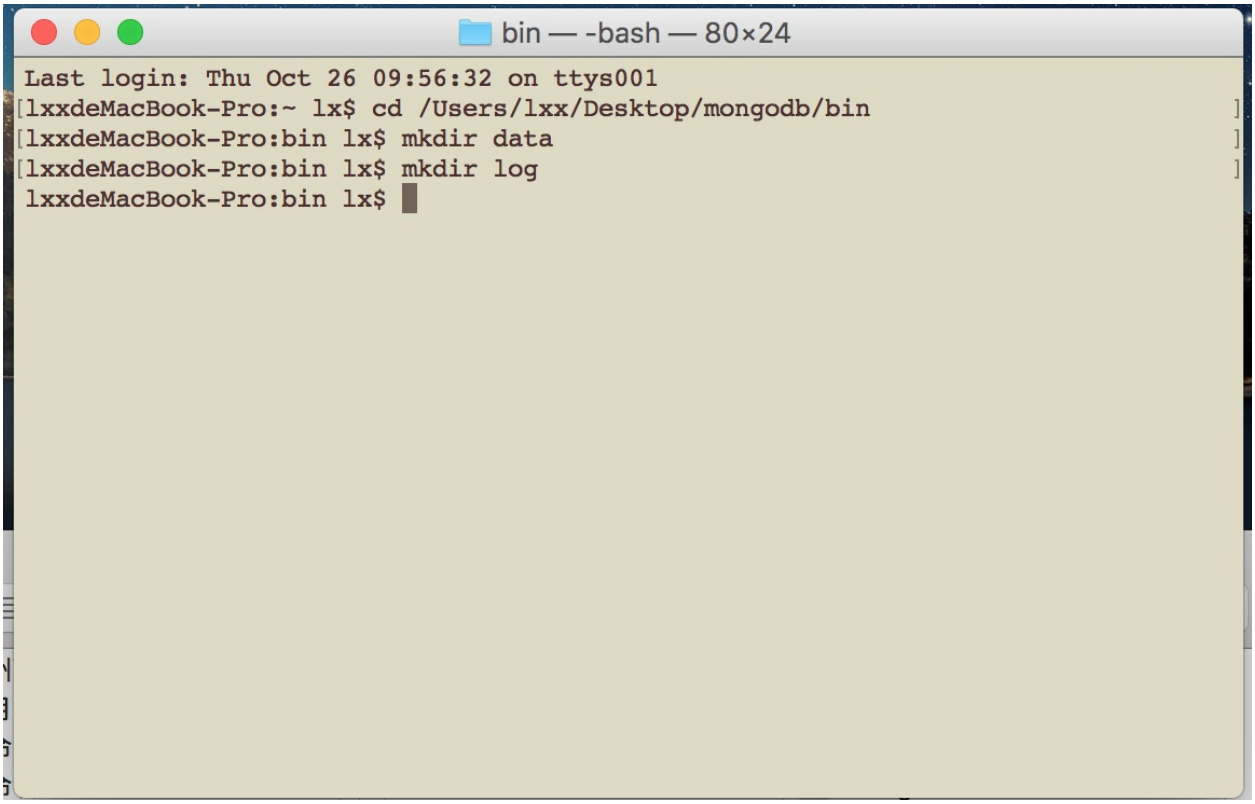
1.首先进入 mongodb的bin目录



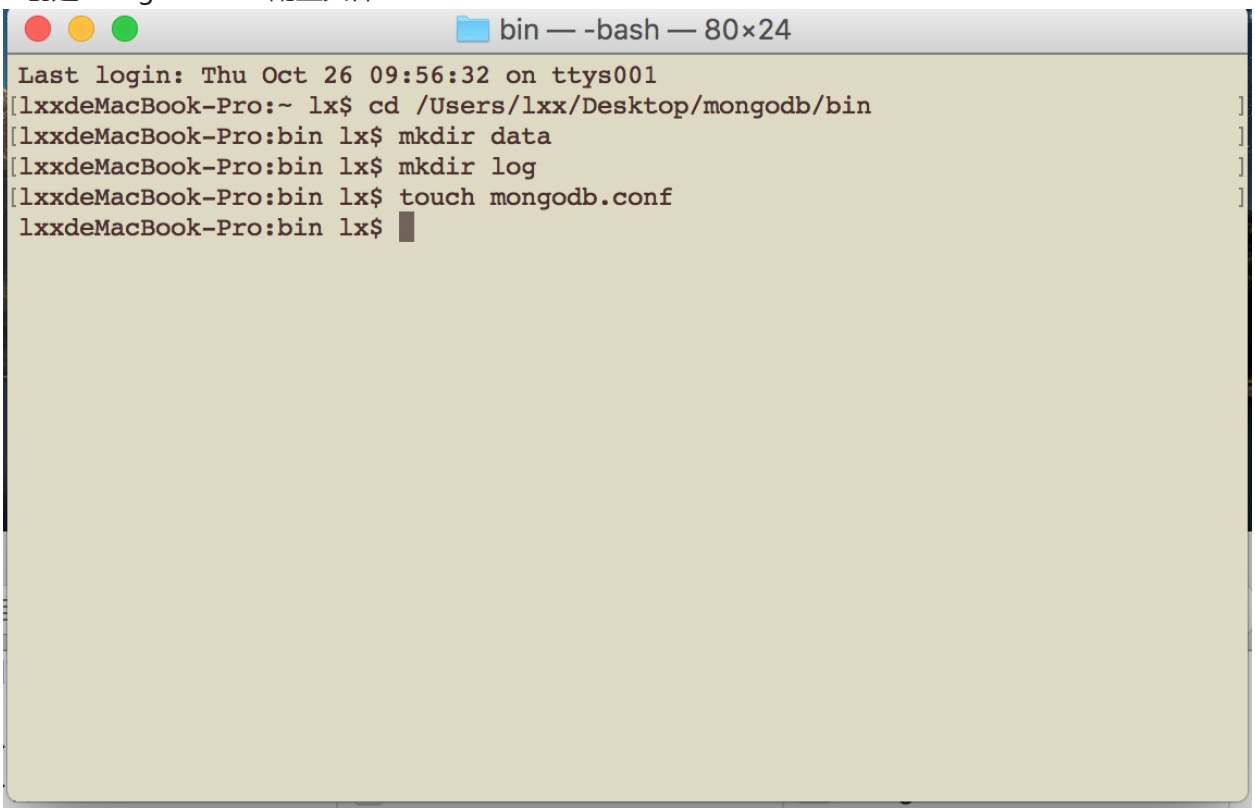
2.建立data文件夹用来记录数据，log文件夹用来记录日志

mkdir data

mkdir log



3.创建mongodb.conf配置文件



4.编辑mongodb.conf 配置文件

mongodb.conf

```
port=27017
dbpath=/Users/lxx/Desktop/mongodb/bin/data
logpath=/Users/lxx/Desktop/mongodb/bin/log/mongodb.log
fork = true
```

port: 数据库服务使用端口
dbpath: 数据存放的文件位置
logpath: 日志文件的存放位置
fork: 后台守护进程运行
5.启动
在bin路径下，执行 ./mongod -f mongodb.conf

打印出类似这样的信息就成功了
about to fork child process, waiting until server is ready for connections.
forked process: 37901
child process started successfully, parent exiting

如果未启动成功，错误信息如下的话：

1. about to fork child process, waiting until server is ready for connections.

2. forked process: 760
3. ERROR: child process failed, exited with error number 1
   一般情况下是权限问题，使用sodu操作来解决，

也可能是配置文件中路径写的有问题。

6.关闭MongoDB服务
在 ./mongo 进入控制台后，输入 use admin,然后输入 db.shutdownServer()
7.查看日志
tail -f log/mongod.log
log/mongod.log 为日志存放路径

接下来将DOClever的源码down到本地，在命令行下运行node DOClever的根目录/Server/bin/www（如果是windows环境下，请修改目录分隔符)，第一次启动，会出现命令行提示符，按照提示符输入即可完成相关的配置，等到DOClever启动成功后， 在浏览器里输入localhost:DOClever启动的端口号,出现首页表示部署成功

最后在浏览器中输入之前的端口号

localhost:12000


red/DOClever/temp) : /Users/lxx/Desktop/list/file/temp
目录创建成功
请输入端口号（比如10000）：12000
DOClever启动成功
GET / 302 11.065 ms - 84
localhost:12000/html/web/index.html
DOClever
开源
QQ群
关于
帮助
登录
注册
DOClever
让接口更懂你
观看视频
协作管理
自动化测试
免费开源

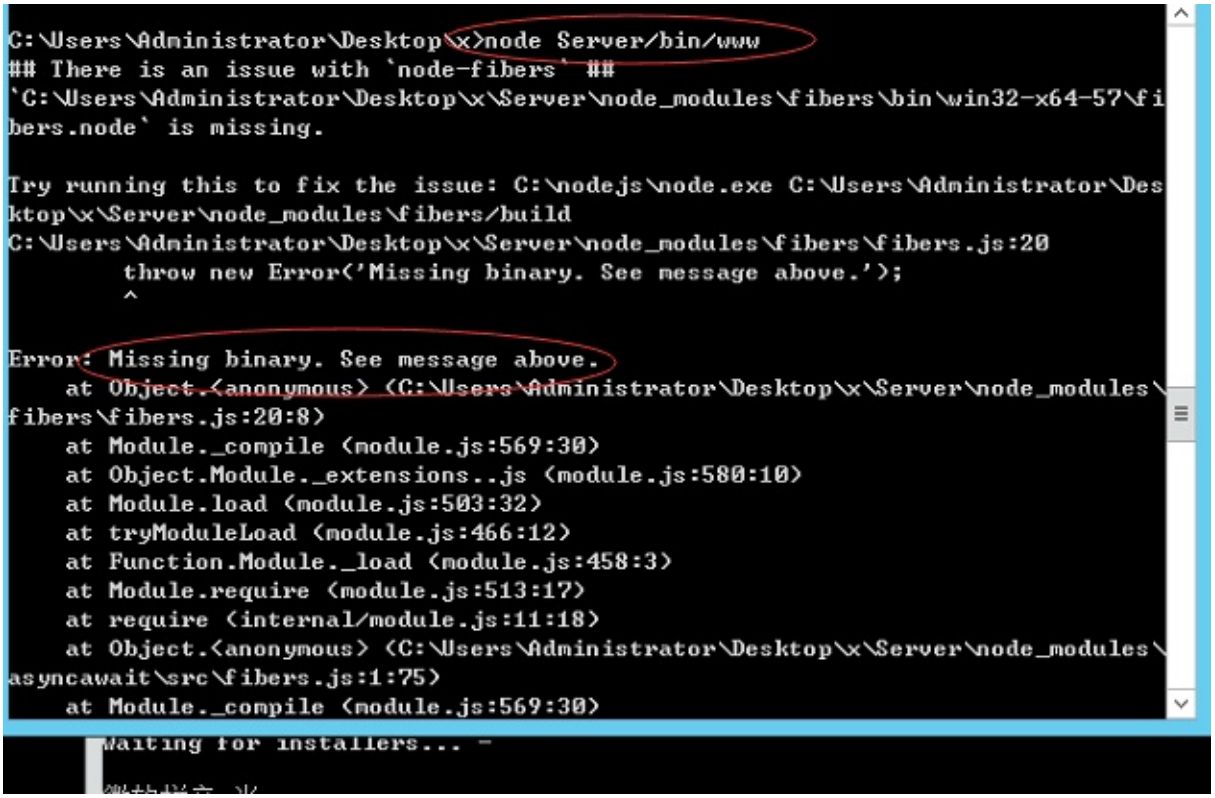
这样就成功了

常见问题

如果我用nginx接管了前端，那么需要注意什么呢？

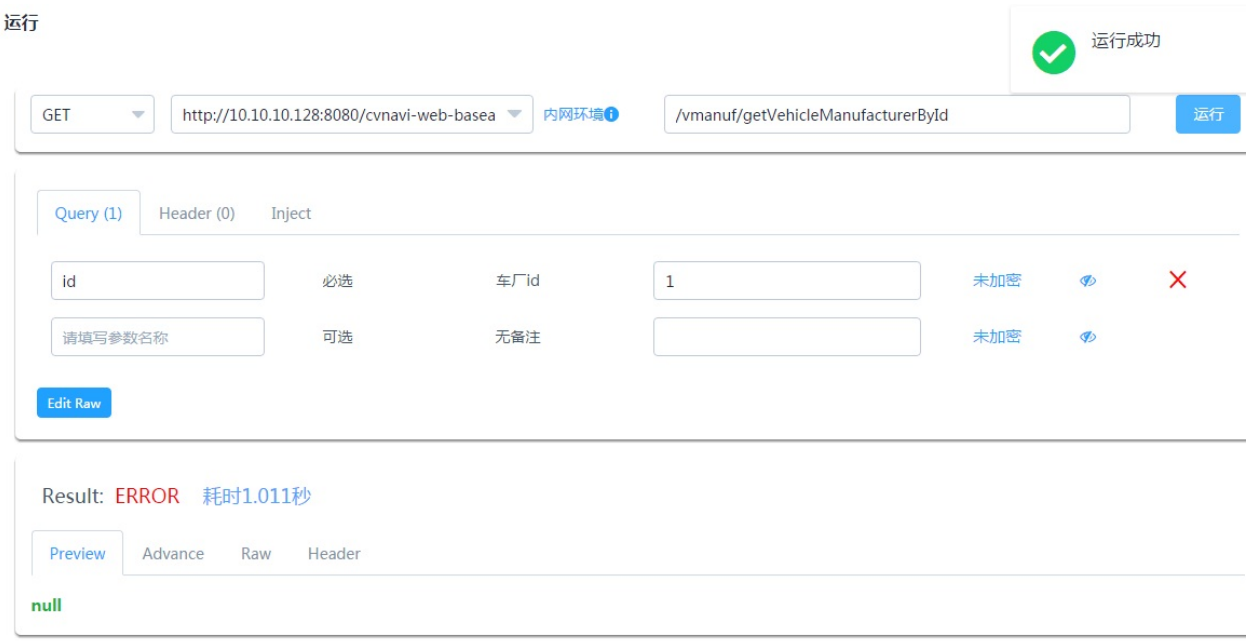
> 如果你用nginx接管了前端页面，那么需要对nginx做一些简单的设置，因为DOClever在调试接口的时候会发送一些下划线开头的自定义http头部，而nginx默认会过滤掉这些头部，所以我们需要在nginx的配置文件中http或者server的配置项中添加underscores_in_headers on;允许在header的字段中带下划线，然后重启nginx即可

出现如下错误:



> 说明你的node版本高了，导致编译不过，建议你使用6.10.0版本，这个版本为lts版本，长期维护，下载地址

出现如下错误:



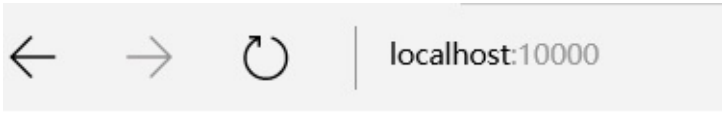
> 因为你的url是内网环境，如果你是用的线上版本，请确保net.js 已经开启，如果你用的是内网部署版本，说明你内网部署的DOClever服务器和你测试的接口服务器不在同一个内网里，请确保Proxy开关打开：

> 然后启动net.js即可！

出现如下错误:

> 说明你的mongodb没有连接上，检查mongodb是否已经开启，可以用romongodb这样的客户端根据检测，然后确保你config.json里面的数据库配置信息是否填写正确，数据库是否存在。

出现如下错误：



{"info":"Invalid request!"}

> 说明你的端口号暂时不可用，这个时候你进行DOClever的根目录，找到config.json文件，修改里面的port，保存后重启

>常见问题会不定时更新！如果你遇见一些问题可到官网或扣扣群反馈给我们!

更新日志

更新日志

2017-10-23 发布4.1.3版本 DOClever 开源地址

1.修复了部分swagger导入的接口不能显示的bug
2.修复了总后台管理员找不到账户的bug
3.修复了总后台列表重用的bug

2017-10-22 发布4.1.2版本 DOClever 开源地址

1.修复了postman 导出json不能导入的bug
2.修复了自动化测试不能插入接口的bug

2017-10-21 发布4.1.1版本 DOClever 开源地址

1.修复了接口两次保存提示失败的bug
2.修复了项目public情况下可以修改pulic属性的bug
3.修复了切换版本不能返回主版本的bug

2017-10-20 发布4.1.0版本 DOClever 开源地址

1.保存接口支持邮件通知到项目成员
2.支持接口入参JSON和出参JSON的拖拽排序
3.深度支持swagger，通过swagger导入的工程可根据swagger json或者url更新本工程
4.在出参mock里面可通过@mj前缀来支持mockjs
5.引入总后台管理，默认用户名：DOClever 默认密码：DOClever
6.markdown文档编辑增强，可以实时预览
7.项目可以被设置成公开化，允许所有注册成员浏览该项目
8.重写postman的导入，支持新版postman导出JSON
9.自动化测试接口编辑页面添加了动态参数注入优先级降低选项，可以先注入参数，再执行inject
10.修复了上一个版本的大量bug

2017-10-17 发布4.0.5版本 DOClever 开源地址

修复了接口运行后生成数据时多标签数据丢失的bug

2017-09-25 发布4.0.4版本 DOClever 开源地址

1.修复了接口快照保存失败的问题
2.大幅增强的swagger的导入功能

2017-09-22 发布4.0.3版本 DOClever 开源地址

1.修复了swagger导入的时候出现的一些错误
2.修复了在运行界面不能edit raw的问题

2017-09-20 发布4.0.2版本 DOClever 开源地址

1.修复了团队普通成员不能快速切换项目的bug
2.修复了接口分组移动到子分组的bug
3.第一次启动的时候不需要配置config.json文件了 ，原先的已经配置过的不受影响，第一次启动会在命令行下提示你输入相关的配置信息

2017-09-17 发布4.0.1版本 DOClever 开源地址

1.修复了4.0.0版本大量bug
2.添加了团队转让功能
3.优化了不少界面样式

2017-09-15 发布4.0.0版本 DOClever 开源地址

1.vue，element升级到最新版
2.登陆支持记住登录状态，支持第三方qq登陆
3.项目列表页进行了归类
4.支持在项目详情页进行项目快速切换，支持项目的所有者转让
5.支持接口分组的无限分组功能
6.支持doclever启动的时候自定义配置启动参数
7.支持对项目观察者进行自定义权限的设置
8.支持对项目的入参和出参建立多个实例
9.

2017-08-19 发布3.0.11版本 DOClever 开源地址

修复了自定义header里面设置cookie无效的bug

2017-08-16 发布3.0.10版本 DOClever 开源地址

1.修复了导入json不能导入多层array的bug
2.接口编辑页全局标签对观察者开放

2017-08-14 发布3.0.9版本 DOClever 开源地址

1.修复了xml不显示的bug
2.修复了导入json只显示第一项的bug
3.修复了自动化测试后台轮询返回类型不正确的bug
4.修复了内网部署中proxy开关失效的bug

2017-08-05 发布3.0.8版本 DOClever 开源地址

1.修复接口编辑里raw类型切换的bug
2.在接口注入里添加了param对象
3.修复了自动化测试后台轮询报错的bug
4.修复了接口返回为数组不能生成数组模型的bug

2017-08-03 发布3.0.7版本 DOClever 开源地址

1.修复了mock数据Boolean不能为false的bug

2.修复了自动化测试后台轮询输出错误的bug

3.修复了接口运行返回的字符串为html时不能preview的bug

2017-08-03 发布3.0.6版本 DOClever 开源地址

修复了接口运行不能保存登陆状态的bug

2017-08-02 发布3.0.5版本 DOClever 开源地址

1.修复了接口编辑页里raw格式下json的值列表不能重新生成的bug

2.修复了在团队模式下团队管理员进入非项目成员项目的多处bug

2017-07-31 发布3.0.4版本 DOClever 开源地址

1.修复了自动化测试若干bug

2.支持接口测试返回xml和html

3.添加接口状态已废弃

2017-07-25 发布3.0.3版本 DOClever 开源地址

1.修复了share接口报错的问题

2.修复了团队管理里面提示没有权限的bug

2017-07-23 发布3.0.2版本 DOClever 开源地址

1.完善了视频文档

2.修复了导入swagger可能出现的bug

3.修复了接口快照的一些隐藏问题

2017-07-21 发布3.0.1版本 DOClever 开源地址

1.修复了接口分享打不开的bug

2.修复了无法接口导入的bug

3.为net.js添加了日志输出

4.更新了文档视频（更多视频在完善中），新增了疑难解答

2017-07-19 发布3.0.0版本 DOClever 开源地址

1.SBDoc更名啦，正式启用DOClever，域名变为doclever.cn ，原先sbdoc.cn仍然可以使用

2.首页大改版，修复了若干bug

2017-07-18 发布2.3.0版本 DOClever 开源地址

1.添加了rap导入功能

2.添加了swagger导入功能

3.修复了在观察者状态下提示没有权限的bug

2017-07-13 发布2.2.3版本 DOClever 开源地址

1.修复了proxy代理接口的一些bug
2.修正了postman新版本导出json不能导入的bug

2017-07-12 发布2.2.2版本 DOClever 开源地址

1.修复了自动化测试中接口不能自动同步的bug
2.在接口参数中，添加了JSON Array编辑选项
3.修复了net.js代理里Access-Control-Expose-Headers为空的bug

2017-07-11 发布2.2.1版本 DOClever 开源地址

1.修复了团队中项目成员管理筛选成员不能选择的bug
2.为baseUrl添加了备注

2017-07-10 发布2.2.0版本 DOClever 开源地址

1.添加了团队管理功能
2.添加了项目版本和接口快照功能，可以在不同的版本和快照间切换和回滚
3.支持自动化测试的轮询功能，可以定时在后台执行job
4.在接口运行中支持application/octet-stream类型的json解析
5.在项目中添加markdown文档，可以管理和项目有关的文档信息
6.添加消息中心，所有用户和项目消息都在这里
7.修复了自动化测试里测试脚本不能含有单引号的bug
8.在独立部署版本里用户可以选择是否通过net.js来做接口代理

2017-07-07 发布2.2.0beta1版本 DOClever 开源地址

1.添加团队管理的功能
2.添加项目版本和接口快照的功能
3.添加自动化测试的轮询功能

2017-05-26 发布2.1.2版本 DOClever 开源地址

1.添加了接口分享的功能
2.修复了接口入参的key没有编码的bug
3.修复了运行接口遇到302跳转报错的bug
4.修复了导入json不能新增字段的bug
5.修复了运行接口时json输入框不会出现的bug

2017-05-23 发布2.1.1版本 DOClever 开源地址

1.修复了在接口编辑页面下query和post切换出现的bug
2.修复了在body json编辑下不能展开折叠的bug
3.对前端页面进行了显示优化，缩短了加载时间
4.修复了导入输入框不能导入json文件的bug
5.添加了http patch方法
6.修复了接口baseurl自动记忆导致的bug

2017-05-21 发布2.1.0版本 DOClever 开源地址

1.前端页面做了大幅度的重构
2.添加了导出到html文档的功能
3.支持工程和接口导入时候的文件拖拽
4.在添加参数的时候去掉了加号
5.绑定状态码样式修改
6.在运行接口的时候添加了baseurl自动记忆功能
7.修复了raw导入json未填值的bug
8.合并pr,添加docker的支持

2017-05-12 发布2.0.3版本 DOClever 开源地址

1.在接口自动化测试中新增了input方法，可以等待用户输入。
2.修复了在firefox下不能导出项目，导出接口的bug
3.修复了接口自动化测试中批量测试无法显示文件上传选择页的bug
4.修复了自动化测试中一个测试用例可以递归的嵌套自身的bug
5.修复了多个页面样式的错误

2017-05-10 发布2.0.2版本 DOClever 开源地址

1.修复了在windows环境下上传头像不能显示的bug
2.修复了自动化测试里运行脚本错误和不能编辑接口的bug
3.在参数编辑组件里参数名称添加了去除前后空格的处理
4.自动化测试中如果脚本运行错误可以捕获错误并显示
5.修正了若干页面的样式

2017-05-06 发布2.0.1版本 DOClever 开源地址

1.修复了vue和element版本不兼容导致的诸多bug
2.修复了在firefox下接口运行返回图片不能显示的bug
3.修复了接口编辑页面路径粘贴报错的bug
4.修复了本地不能存储sessionid的bug
5.修复了在生成接口的时候添加新的baseurl没有响应的bug

2017-05-04 发布2.0.0版本 DOClever 开源地址

1.支持项目的导出导入，支持接口，以及分组的导入导出
2.支持导入postman的v2 json格式文件
3.在接口编辑页支持入参为json格式的可视化编辑
4.添加了全局的状态码功能，并且入参和出参可以和状态码进行绑定。
5.返回参数支持数组格式的可视化编辑
6.添加了全局注入功能。
7.在mock中支持@code方法，可以运行自定义的js代码
8.参数的默认值可以添加备注说明了。

9.在添加项目成员里新增了导入功能，可以将其他项目的成员导入进来。
10.接口支持按名称和url关键词来进行搜索了
11.支持接口的自动化测试了，可以编写测试用例，支持统一运行。
12.对接口编辑页面进行了UI调整，结构更清晰。

2017-03-28 发布1.1.9版本 DOClever 开源地址

1.修复了导入query，header，body字符串不能在运行界面里显示的bug
2.对预览界面做了微调，优化显示效果
下一个版本将是2.0.0版本 很多重大更新将呈现

2017-03-25 发布1.1.8版本 DOClever 开源地址

1.修复了mock可能存在的bug
2.新增了MockServer选项，可以测试mock服务器
3.在接口列表的分组里添加了分组里接口的个数
4.mock规则现在生成支持图片的大小啦

2017-03-20 发布1.1.7版本 DOClever 开源地址

1.修复了接口编辑页面在新建接口的情况下不能把信息带入到运行页面里面的bug
2.修复了项目管理可以修改自身权限的bug
3.修复了接口加密根据加密类型动态显示salt的bug
4.mock数据为array类型添加了@count(min,max)规则，可以根据min和max随机生成数组的个数
5.优化了query和body的填值文字展示
6.在登陆页面回车即可登陆
7.添加了检查更新按钮，可以检测当前的版本并下载最新版本。
8.更新了帮助中心里的教学视频

2017-03-16 发布1.1.6版本 DOClever 开源地址

1.把接口测试页面移动到了接口编辑页面内作为一个弹出层显示
2.修复了接口列表点击分组不能展开和接口名称过长出现的bug
3.在接口列表里增加了对勾显示该接口已开发完成
4.去掉了接口测试里MockServer，目前点击预览即可实现Mock的功能

2017-03-14 发布1.1.5版本 DOClever 开源地址

1.修复了返回结果匹配智能分析可能出现的bug
2.修复了result列表里数组元素下添加子元素可以输入name的bug

2017-03-14 发布1.1.4版本 DOClever 开源地址

1.修复了mock时候出现错误的bug
2.修复了query和body情况下填值的列表没有情况的bug

2017-03-13 发布1.1.3版本 DOClever 开源地址

1.添加了接口的复制粘贴功能

2017-03-12 发布1.1.2版本 DOClever 开源地址

1.修复了修改头像的bug
2.修复了网络传输时对特殊字符的编码解码bug
这一版强烈建议更新

2017-03-10 发布1.1.1版本 DOClever 开源地址

1.修复了query列表里复选框不能更改的bug
2.新增了在接口编写页和测试页面可以粘贴path路径，系统会自动识别path和query的功能

2017-03-09 发布1.1.0版本 DOClever 开源地址

1.修改了首页样式
2.在编辑接口的时候body为raw的状态下可以快捷修改content-type
3.在result列表添加了mixed类型，表示任意数据类型，同时mock添加了三种规则@null,@arr,@obj（后两种只针对mixed类型）
4.可以在接口运行前后注入js代码，比如我要对接口字段进行自定义的加密，这个时候就可以在接口运行前注入自定义的js代码，同时SBDoc内置了很多变量和函数供用户使用
5.接口测试页面的返回结果里，mixed标签更改为advance啦，加入了接口数据智能解析功能，可以识别接口数据类型是否正确，以及必有字段是否存在于返回数据中

2017-03-08 发布1.0.6版本 DOClever 开源地址

1.修改了首页的背景图片
2.修复了在测试接口页面下header和body不能commit raw的bug

2017-03-08 发布1.0.5版本 DOClever 开源地址

1.修复了个别情况下不能显示项目列表的bug
2.修复了在观察者状态下可以通过项目列表修改项目信息的bug
3.修复了在火狐下拖动会打开新标签的bug
4.修改了首页的顶部样式

2017-03-07 发布1.0.4版本 DOClever 开源地址

1.修复了result列表里面的对其显示问题

2017-03-07 发布1.0.3版本 DOClever 开源地址

1.修复了接口运行结果显示的样式问题
2.修复了测试接口不能自定义部分http header问题
3.在新建运行的状态下不能进行mock

2017-03-07 发布1.0.2版本 DOClever 开源地址

1.修复了头像显示的问题

2.解决了网络依赖问题，不再依赖外网资源，完全的纯内网版本

2017-03-07 发布1.0.1版本DOClever 开源地址

1.修复了头像不显示倒是网页显示一直在加载的bug
2.修复了运行接口下除status为200的情况外不显示返回数据的bug

2017-03-06 发布1.0.0版本 DOClever 开源地址