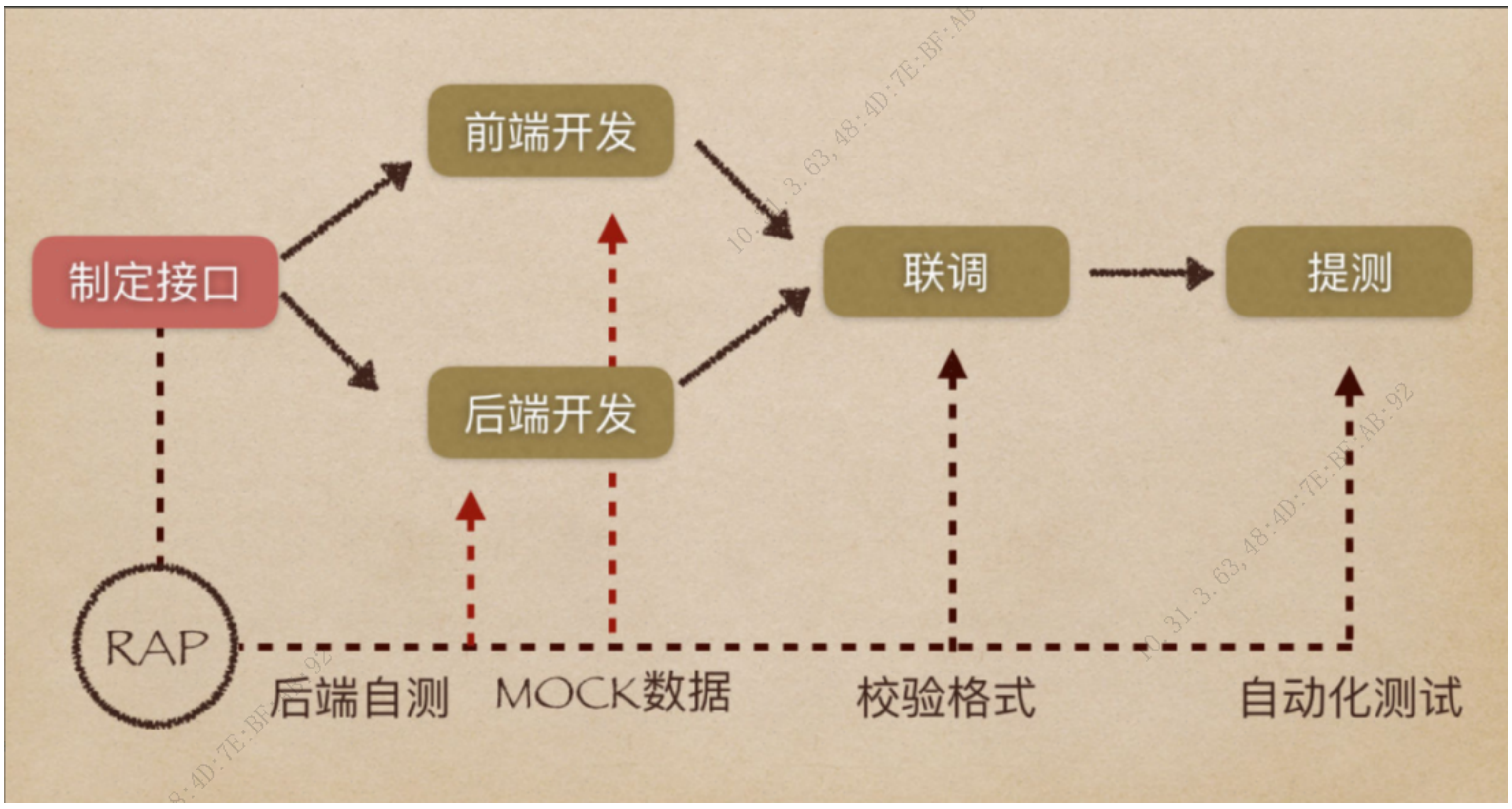
Rap2 接口平台

参考文档：https://github.com/thx/rap2-delos

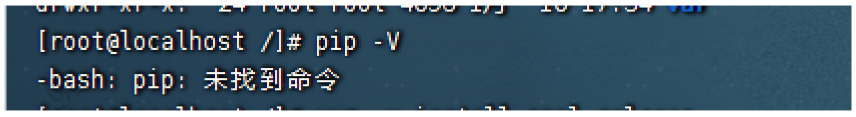
---

docker-compose： 定义和运行多容器分布式部署 Docker 应用的工具



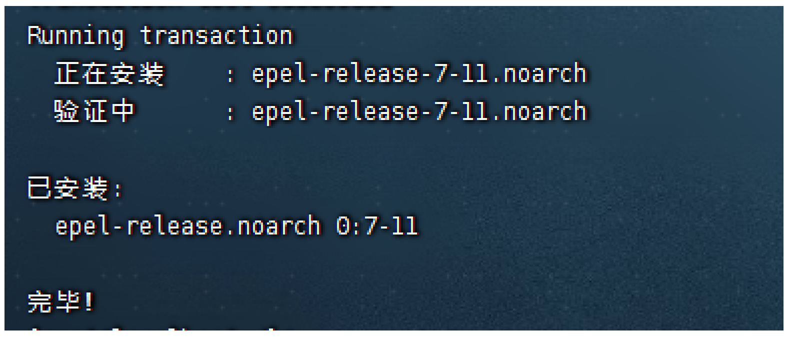
1.安装 python-pip

pip -V　　　　　　　　　　　　#检查是否已经安装pip(pip 是 Python 的包管理工具，一般在安装 Python 的时候，它已经被安装到本地了)



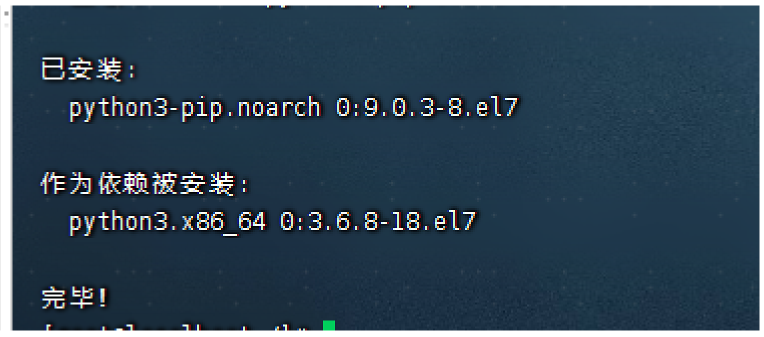
```
[root@localhost /]# pip -V
-bash: pip: 未找到命令
```

yum -y install epel-release



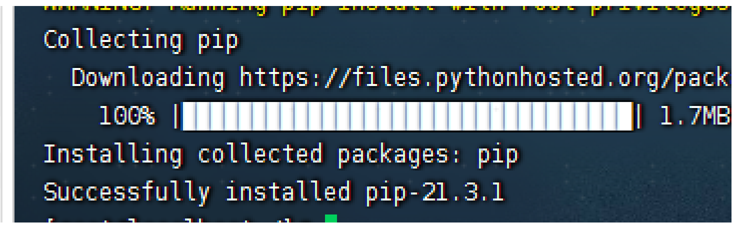
```
Running transaction
  正在安装    : epel-release-7-11.noarch
  验证中      : epel-release-7-11.noarch

已安装:
  epel-release.noarch 0:7-11

完毕!
```

yum install python3-pip　　　　　　#安装(卸载 python -m pip uninstall pip 查看 pip3 show pip) //3操作pip3，没有3操作pip2



```
已安装:
  python3-pip.noarch 0:9.0.3-8.el7

作为依赖被安装:
  python3.x86_64 0:3.6.8-18.el7

完毕!
```

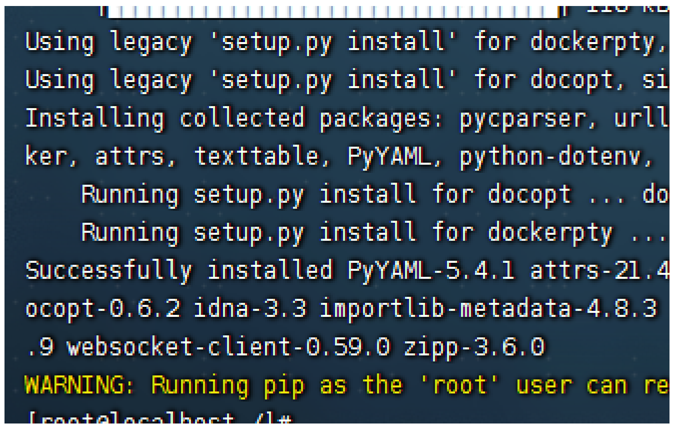
pip3 install --upgrade pip　　　　　#升级pip



```
Collecting pip
  Downloading https://files.pythonhosted.org/pack
    100% |████████████████████████████████| 1.7MB
Installing collected packages: pip
Successfully installed pip-21.3.1
```

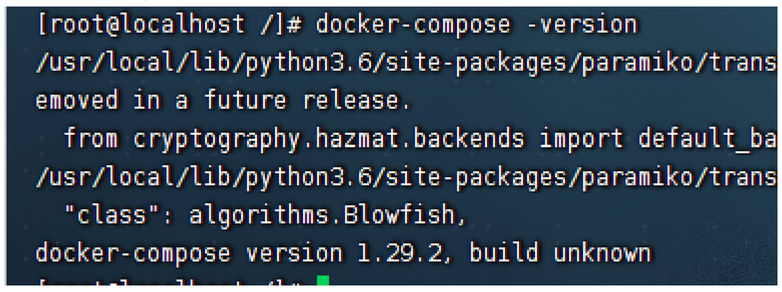
2.安装docker-compose

pip3 install docker-compose



```
Using legacy 'setup.py install' for dockerpty,
Using legacy 'setup.py install' for docopt, si
Installing collected packages: pycparser, urll
ker, attrs, texttable, PyYAML, python-dotenv,
    Running setup.py install for docopt ... do
    Running setup.py install for dockerpty ...
Successfully installed PyYAML-5.4.1 attrs-21.4
ocopt-0.6.2 idna-3.3 importlib-metadata-4.8.3
.9 websocket-client-0.59.0 zipp-3.6.0
WARNING: Running pip as the 'root' user can re
```

docker-compose -version



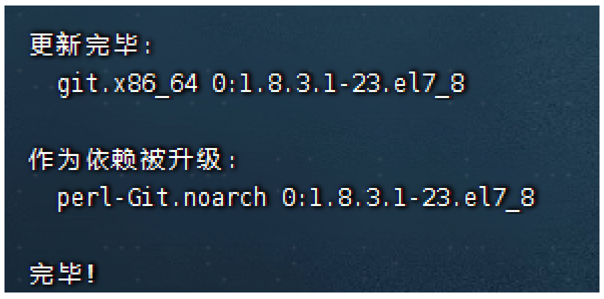
```
[root@localhost /]# docker-compose -version
/usr/local/lib/python3.6/site-packages/paramiko/trans
emoved in a future release.
  from cryptography.hazmat.backends import default_ba
/usr/local/lib/python3.6/site-packages/paramiko/trans
  "class": algorithms.Blowfish,
docker-compose version 1.29.2, build unknown
```

3.rap2后台搭建

【rap2-delos 后端数据API服务器】https://www.jb51.net/article/145595.htm

1.安装git

yum -y install git-core



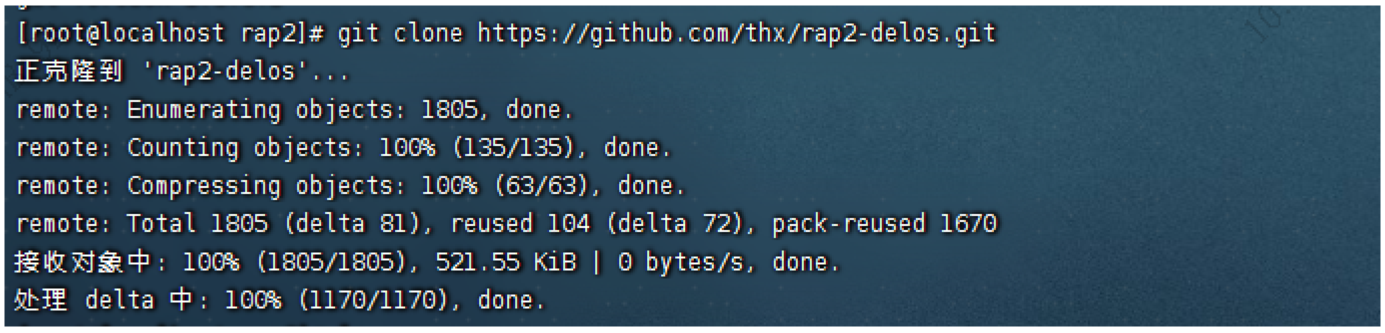
```
更新完毕:
  git.x86_64 0:1.8.3.1-23.el7_8

作为依赖被升级:
  perl-Git.noarch 0:1.8.3.1-23.el7_8

完毕!
```

2.搭建rap2-delos

mkdir -p /home/rap2

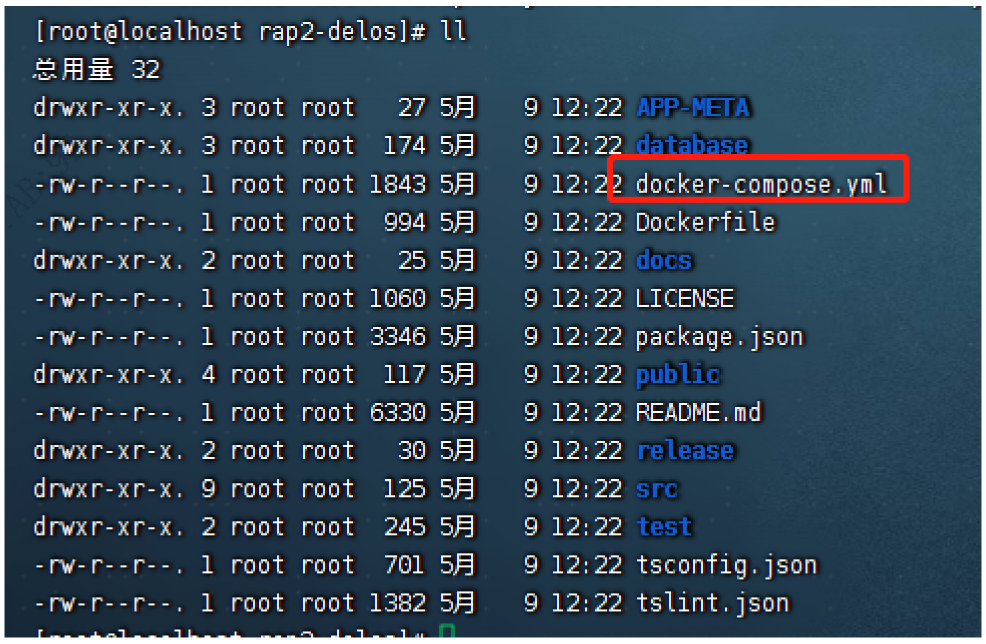
cd rap2

git clone https://github.com/thx/rap2-delos.git



```
[root@localhost rap2]# git clone https://github.com/thx/rap2-delos.git
正克隆到 'rap2-delos'...
remote: Enumerating objects: 1805, done.
remote: Counting objects: 100% (135/135), done.
remote: Compressing objects: 100% (63/63), done.
remote: Total 1805 (delta 81), reused 104 (delta 72), pack-reused 1670
接收对象中: 100% (1805/1805), 521.55 KiB | 0 bytes/s, done.
处理 delta 中: 100% (1170/1170), done.
```

cd rap2-docker



```
[root@localhost rap2-delos]# ll
总用量 32
drwxr-xr-x. 3 root root   27 5月   9 12:22 APP-META
drwxr-xr-x. 3 root root  174 5月   9 12:22 database
-rw-r--r--. 1 root root 1843 5月   9 12:22 docker-compose.yml
-rw-r--r--. 1 root root  994 5月   9 12:22 Dockerfile
drwxr-xr-x. 2 root root   25 5月   9 12:22 docs
-rw-r--r--. 1 root root 1060 5月   9 12:22 LICENSE
-rw-r--r--. 1 root root 3346 5月   9 12:22 package.json
drwxr-xr-x. 4 root root  117 5月   9 12:22 public
-rw-r--r--. 1 root root 6330 5月   9 12:22 README.md
drwxr-xr-x. 2 root root   30 5月   9 12:22 release
drwxr-xr-x. 9 root root  125 5月   9 12:22 src
drwxr-xr-x. 2 root root  245 5月   9 12:22 test
-rw-r--r--. 1 root root  701 5月   9 12:22 tsconfig.json
-rw-r--r--. 1 root root 1382 5月   9 12:22 tslint.json
```

vi docker-compose.yml　　　　　//修改容器名、端口、容器镜像版本(redis存储用户及session、mysql用来存接口数据)

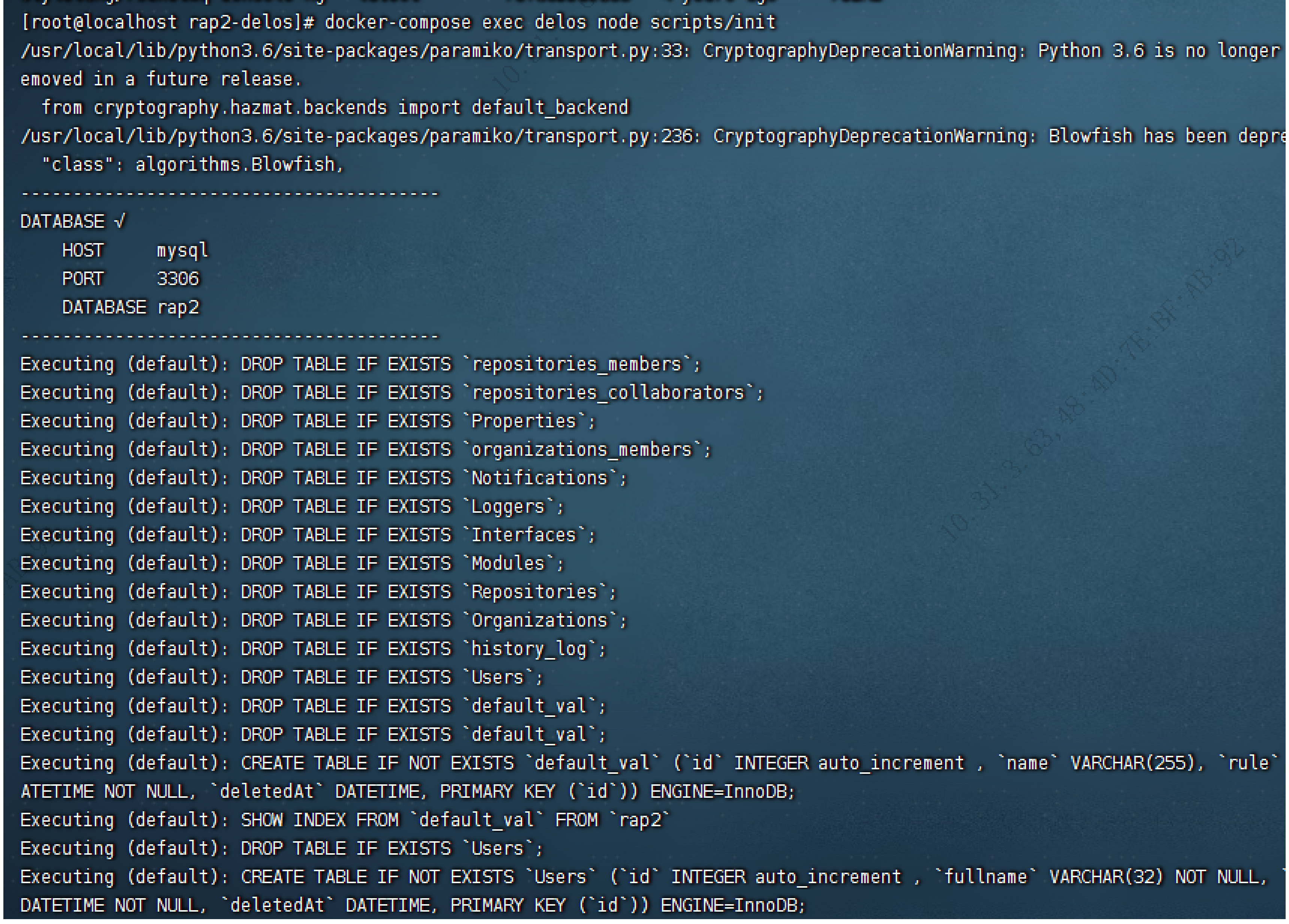
docker-compose.yml
1.91KB

docker-compose up -d　　　　　//docker-compose.yml 可以指定rap2端口号，mysql和redis等容器的构建

//docker-compose down 关闭

3.初始化数据库（只有第一次该这样做）

docker-compose exec delos node scripts/init



```
[root@localhost rap2-delos]# docker-compose exec delos node scripts/init
/usr/local/lib/python3.6/site-packages/paramiko/transport.py:33: CryptographyDeprecationWarning: Python 3.6 is no longer
emoved in a future release.
  from cryptography.hazmat.backends import default_backend
/usr/local/lib/python3.6/site-packages/paramiko/transport.py:236: CryptographyDeprecationWarning: Blowfish has been depre
  "class": algorithms.Blowfish,
----------------------------------------
DATABASE √
    HOST     mysql
    PORT     3306
    DATABASE rap2
----------------------------------------
Executing (default): DROP TABLE IF EXISTS `repositories_members`;
Executing (default): DROP TABLE IF EXISTS `repositories_collaborators`;
Executing (default): DROP TABLE IF EXISTS `Properties`;
Executing (default): DROP TABLE IF EXISTS `organizations_members`;
Executing (default): DROP TABLE IF EXISTS `Notifications`;
Executing (default): DROP TABLE IF EXISTS `Loggers`;
Executing (default): DROP TABLE IF EXISTS `Interfaces`;
Executing (default): DROP TABLE IF EXISTS `Modules`;
Executing (default): DROP TABLE IF EXISTS `Repositories`;
Executing (default): DROP TABLE IF EXISTS `Organizations`;
Executing (default): DROP TABLE IF EXISTS `history_log`;
Executing (default): DROP TABLE IF EXISTS `Users`;
Executing (default): DROP TABLE IF EXISTS `default_val`;
Executing (default): DROP TABLE IF EXISTS `default_val`;
Executing (default): CREATE TABLE IF NOT EXISTS `default_val` (`id` INTEGER auto_increment , `name` VARCHAR(255), `rule`
ATETIME NOT NULL, `deletedAt` DATETIME, PRIMARY KEY (`id`)) ENGINE=InnoDB;
Executing (default): SHOW INDEX FROM `default_val` FROM `rap2`
Executing (default): DROP TABLE IF EXISTS `Users`;
Executing (default): CREATE TABLE IF NOT EXISTS `Users` (`id` INTEGER auto_increment , `fullname` VARCHAR(32) NOT NULL, `
DATETIME NOT NULL, `deletedAt` DATETIME, PRIMARY KEY (`id`)) ENGINE=InnoDB;
```

4.上步无异常可跳过此步骤。重新启动服务（如果访问不了可能是数据库没有链接上，关闭 rap 服务）

docker-compose down

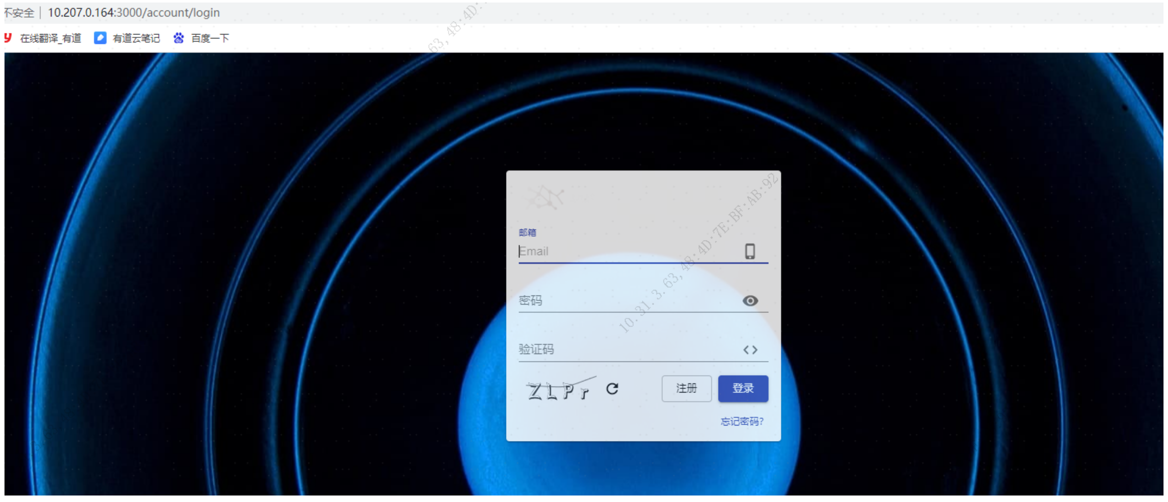
docker-compose up -d

docker-compose exec delos node scripts/updateSchema (如果 Sequelize 报错可能是数据库表发生了变化，运行下面命令同步)

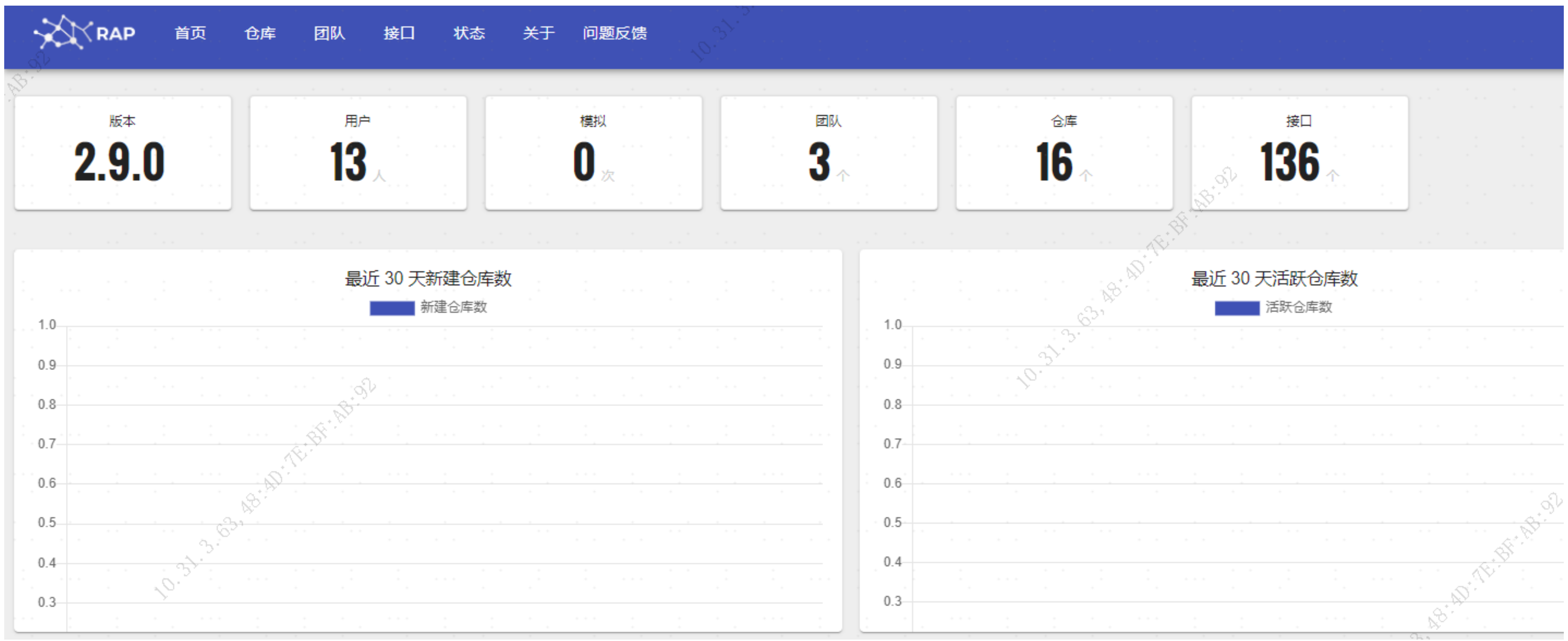
5.访问

http://10.207.0.164:3000　　　　# 前端（可自定义端口号）

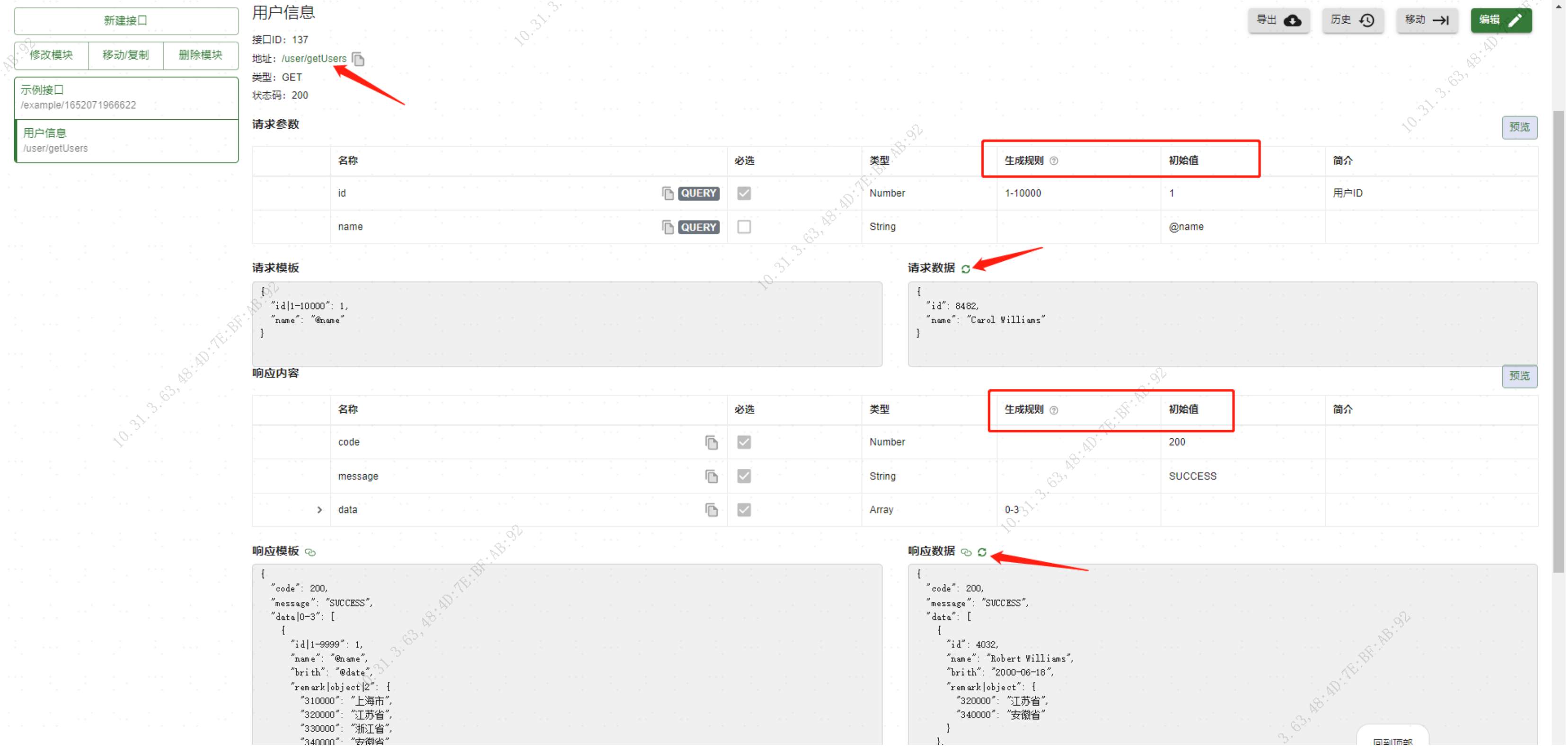
http://10.207.0.164:38080　　# 后端



6.登录 admin/邮箱/admin(退出时，切勿点击注销)



7.使用（mock.js语法：http://mockjs.com/examples.html）



用户信息

接口ID：137
地址：/user/getUsers
类型：GET
状态码：200

请求参数

	名称		必选	类型	生成规则	初始值	简介
	id	QUERY	☑	Number	1-10000	1	用户ID
	name	QUERY	□	String		@name	

请求模板

```
{
  "id|1-10000": 1,
  "name": "@name"
}
```

请求数据

```
{
  "id": 8482,
  "name": "Carol Williams"
}
```

响应内容

	名称	必选	类型	生成规则	初始值	简介
	code	☑	Number		200	
	message	☑	String		SUCCESS	
›	data	☑	Array	0-3		

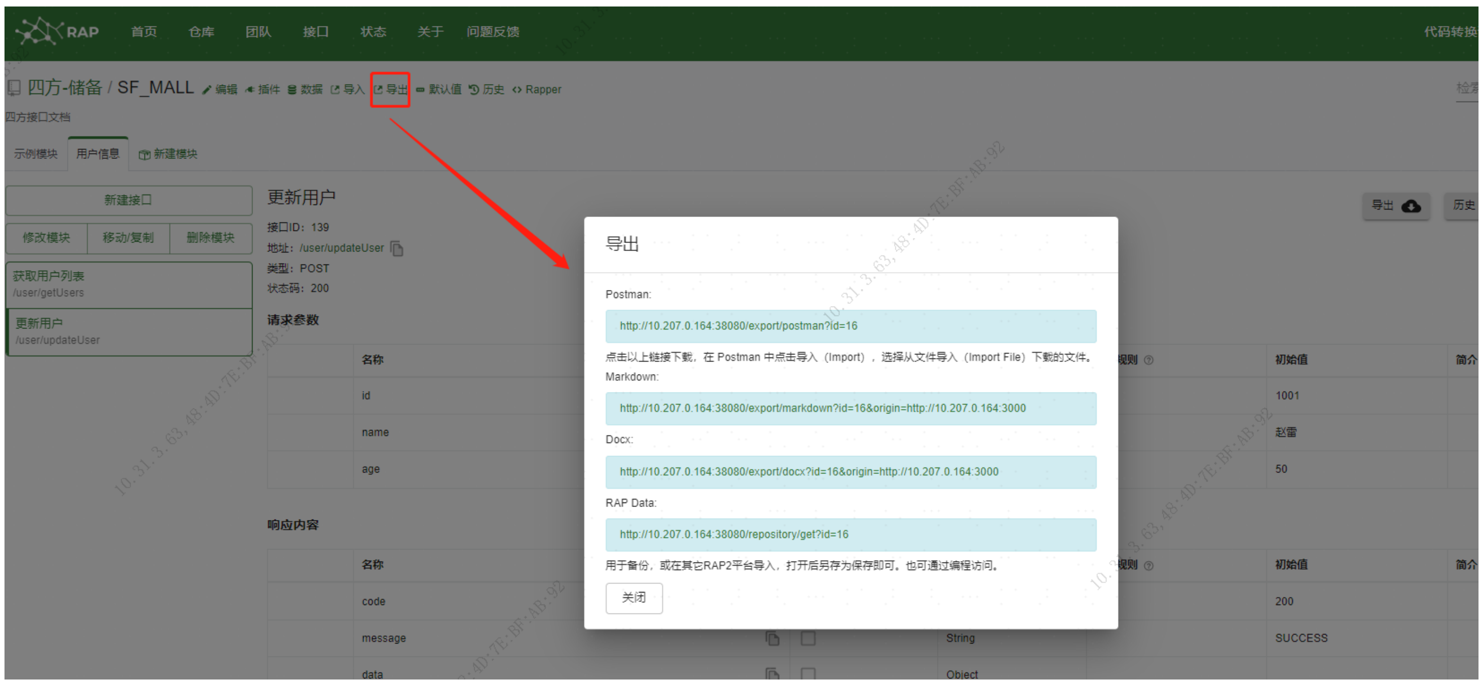
响应模板

```
{
  "code": 200,
  "message": "SUCCESS",
  "data|0-3": [
    {
      "id|1-9999": 1,
      "name": "@name",
      "brith": "@date",
      "remark|object|2": {
        "310000": "上海市",
        "320000": "江苏省",
        "330000": "浙江省",
        "340000": "安徽省"
```

响应数据

```
{
  "code": 200,
  "message": "SUCCESS",
  "data": [
    {
      "id": 4032,
      "name": "Robert Williams",
      "brith": "2000-06-18",
      "remark|object": {
        "320000": "江苏省",
        "340000": "安徽省"
      }
    },
```

导出



更新用户

接口ID：139
地址：/user/updateUser
类型：POST
状态码：200

请求参数

名称	初始值
id	1001
name	赵雷
age	50

响应内容

名称	类型	初始值
code		200
message	String	SUCCESS
data	Object	

导出

Postman:

http://10.207.0.164:38080/export/postman?id=16

点击以上链接下载，在 Postman 中点击导入（Import），选择从文件导入（Import File）下载的文件。

Markdown:

http://10.207.0.164:38080/export/markdown?id=16&origin=http://10.207.0.164:3000

Docx:

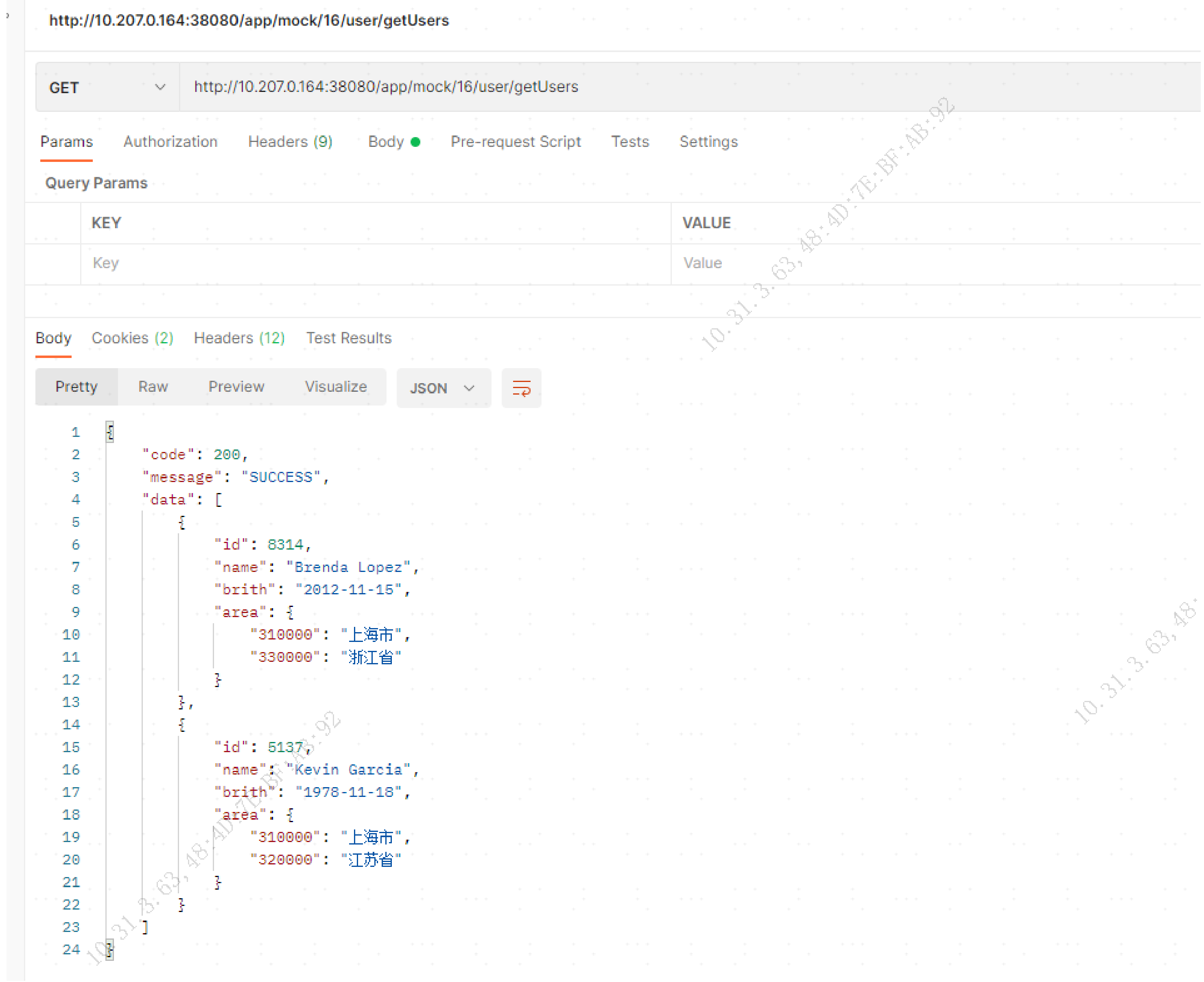
http://10.207.0.164:38080/export/docx?id=16&origin=http://10.207.0.164:3000

RAP Data:

http://10.207.0.164:38080/repository/get?id=16

用于备份，或在其它RAP2平台导入，打开后另存为保存即可。也可通过编程访问。

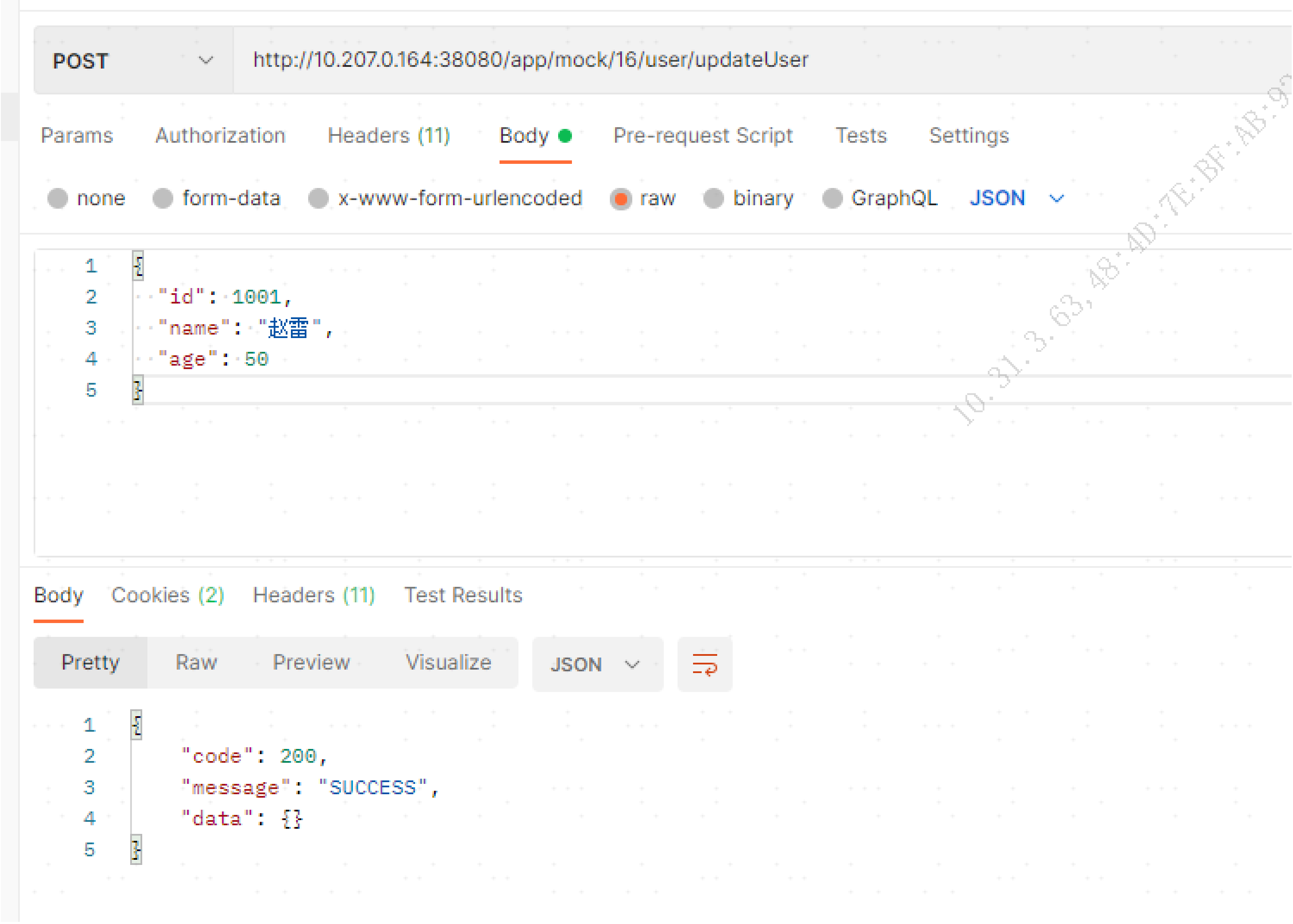
postman测试



http://10.207.0.164:38080/app/mock/16/user/getUsers

GET http://10.207.0.164:38080/app/mock/16/user/getUsers

```
{
    "code": 200,
    "message": "SUCCESS",
    "data": [
        {
            "id": 8314,
            "name": "Brenda Lopez",
            "brith": "2012-11-15",
            "area": {
                "310000": "上海市",
                "330000": "浙江省"
            }
        },
        {
            "id": 5137,
            "name": "Kevin Garcia",
            "brith": "1978-11-18",
            "area": {
                "310000": "上海市",
                "320000": "江苏省"
            }
        }
    ]
}
```



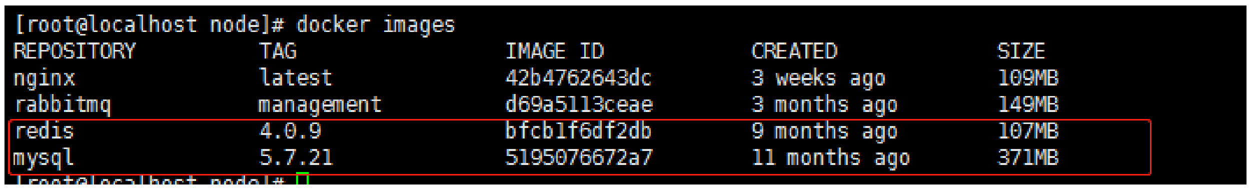
POST http://10.207.0.164:38080/app/mock/16/user/updateUser

```
{
  "id": 1001,
  "name": "赵雷",
  "age": 50
}
```

```
{
    "code": 200,
    "message": "SUCCESS",
    "data": {}
}
```

【非docker方式安装】 https://blog.csdn.net/hww237756557/article/details/80977270

1.redis、mysql使用docker安装



```
[root@localhost node]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx               latest              42b4762643dc        3 weeks ago         109MB
rabbitmq            management          d69a5113ceae        3 months ago        149MB
redis               4.0.9               bfcb1f6df2db        9 months ago        107MB
mysql               5.7.21              5195076672a7        11 months ago       371MB
[root@localhost node]#
```

2.安装node

mkdir -p /data/node

cd /data/node

wget http://nodejs.org/dist/v8.9.4/node-v8.9.4-linux-x64.tar.gz

tar zxvf node-v8.9.4-linux-x64.tar.gz

vim /etc/profile

[内容]

    export NODE_HOME=/data/node/node-v8.9.4-linux-x64

    export PATH=$NODE_HOME/bin:$PATH

source /etc/profile

node -v

npm -v

【rap2-delos 后端数据API服务器】https://blog.csdn.net/hww237756557/article/details/80977055

1.拉取源码

cd /data/rap2

git clone https://github.com/thx/rap2-delos.git

修改src/config/下的三套环境的连接配置

2.建库

CREATE DATABASE IF NOT EXISTS RAP2_DELOS_APP DEFAULT CHARSET utf8 COLLATE utf8_general_ci;

3.安装依赖&初始化数据库

npm install   #rap2-delos目录下执行

npm install -g pm2   #全局安装PM2 使用见：https://www.cnblogs.com/zhoujie/p/nodejs4.html

npm install typescript -g

npm run build   #报错不用管

```
npm run create-db
```

4.编译启动

```
npm run dev                        #dev启动（不能后台运行，仅测试用）
npm start                          #prod启动（修改配置后先 npm run build）
pm2 list  项目id                   #查看项目详情
pm2 start 项目id                   #启动项目
pm2 stop 项目id                    #停止项目
pm2 del 项目id                     #删除项目
```

5.访问

http://192.168.1.28:38080

提示：RAP2后端服务已启动，请从前端服务(rap2-dolores)访问。 RAP2 back-end server is started, please visit via front-end service (rap2-dolores).

【rap2-dolores 前端静态资源】https://blog.csdn.net/hww237756557/article/details/80977177

1.拉取源码

```
cd  /data/rap2
git clone https://github.com/thx/rap2-dolores.git
```

2.初始化： 修改package.json配置项"node-sass": "4.11.0"（最新版本，防止下载报错）

```
npm install node-sass --unsafe-perm --save-dev
npm install
```

3.修改/src/config配置

config.dev.js

config.prod.js

4.编译生成 build 目录

```
npm run build
```

5.将生成的build目录移到nginx目录，配置nginx静态资源访问（如果我们打开界面是一直在加载的话，配置不要使用127.0.0.1）

在nginx.conf添加新监听端口，docker启动开放改端口

```
http {
    #rap2
      server {
            listen      8088;
            server_name  localhost;

            location / {
                root   /usr/share/nginx/html/rap2/build/;
                index  index.html index.htm;
            }
      }
}
```

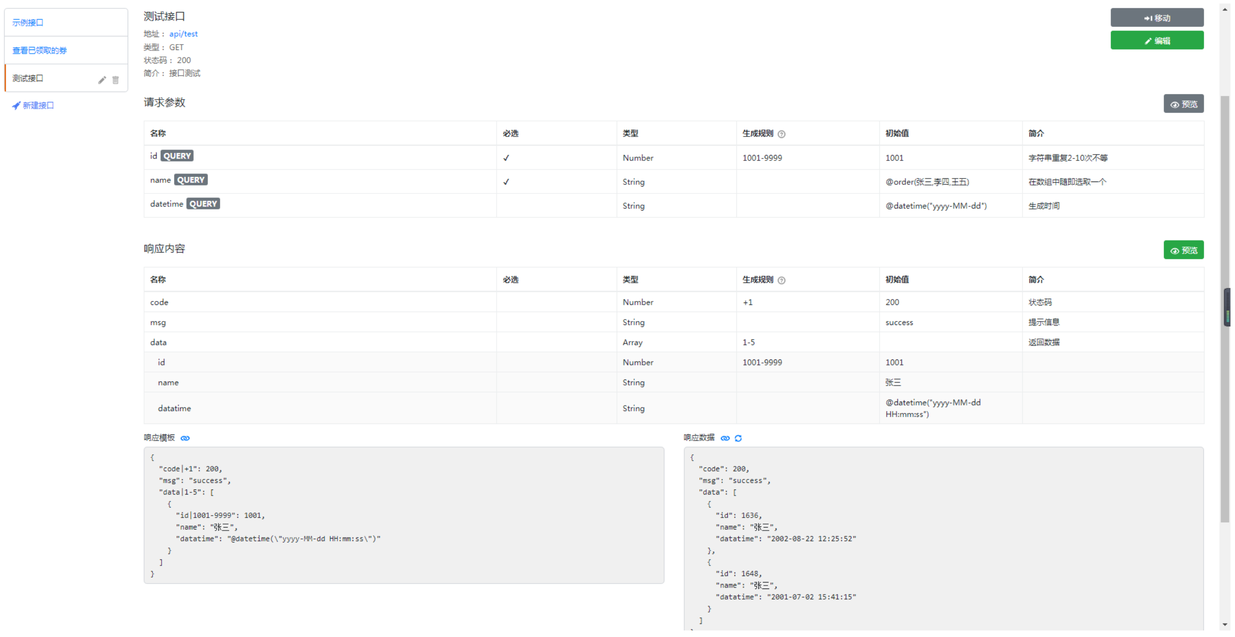
修改nginx.conf相关配置，解决单页面应用刷新 404

```
  #rap2
  server {
        listen      8088;
        server_name  localhost;

        location / {
            root   /usr/share/nginx/html/rap2/build/;
            index  index.html index.htm;
        }

  }
```

改为

```
server {
        listen       8088;
        server_name  localhost;

        location / {
                rewrite .* /index.html break;
                root   /usr/share/nginx/html/rap2/build/;
        }

        location /static {
                alias   /usr/share/nginx/html/rap2/build/static;
                index  index.html;
        }
}
```

【mock.js的使用】



测试接口

地址： api/test

类型： GET

状态码： 200

简介： 接口测试

请求参数

名称	必选	类型	生成规则	初始值	简介
id QUERY	✓	Number	1001-9999	1001	字符串重复2-10次不等
name QUERY	✓	String		@order(张三,李四,王五)	在数组中随即选取一个
datetime QUERY		String		@datetime("yyyy-MM-dd")	生成时间

响应内容

名称	必选	类型	生成规则	初始值	简介
code		Number	+1	200	状态码
msg		String		success	提示信息
data		Array	1-5		返回数据
id		Number	1001-9999	1001	
name		String		张三	
datatime		String		@datetime("yyyy-MM-dd HH:mm:ss")	

响应模板

```
{
  "code|+1": 200,
  "msg": "success",
  "data|1-5": [
    {
      "id|1001-9999": 1001,
      "name": "张三",
      "datatime": "@datetime(\"yyyy-MM-dd HH:mm:ss\")"
    }
  ]
}
```

响应数据

```
{
  "code": 200,
  "msg": "success",
  "data": [
    {
      "id": 1636,
      "name": "张三",
      "datatime": "2002-08-22 12:25:52"
    },
    {
      "id": 1648,
      "name": "张三",
      "datatime": "2001-07-02 15:41:15"
    }
  ]
```

http://mockjs.com/examples.html

https://github.com/thx/RAP/wiki/user_manual_cn

https://www.jianshu.com/p/869fa17dc9d4?from=singlemessage