点点数字钱包文档V1.5

[系统简介 2](#_Toc6873)

[部署篇 6](#_Toc28586)

[资料准备 6](#_Toc12300)

[1) 短信申请-飞鸽 6](#_Toc6677)

[2) 邮箱申请 7](#_Toc25442)

[3) trongrid申请（选填） 9](#_Toc5747)

[系统部署 10](#_Toc8387)

[4) 服务器配置 10](#_Toc26104)

[5) 域名配置 10](#_Toc15837)

[6) 依赖环境 11](#_Toc979)

[7) 下载安装包 11](#_Toc15472)

[8) 基础目录创建 11](#_Toc22734)

[9) Jdk安装 11](#_Toc16384)

[10) nginx安装 12](#_Toc26808)

[11) mysql安装及配置 14](#_Toc23059)

[12) redis安装 15](#_Toc28713)

[13) 安装 Zookeeper 16](#_Toc10597)

[14) 安装 Kafka 16](#_Toc7525)

[15) 安装MongoDB 17](#_Toc27578)

[16) 开机启动 18](#_Toc9582)

[17) api部署 21](#_Toc24368)

[18) 管理后台开发部署 23](#_Toc15998)

[19) H5开发部署 23](#_Toc30534)

[20) 数据库部署 23](#_Toc4997)

[21) 上传文件目录 24](#_Toc19185)

[22) 进程守护 24](#_Toc8287)

[23) 重启系统 25](#_Toc3185)

[常用操作命令 25](#_Toc9429)

[24) zookeeper启动停止 25](#_Toc8396)

[25) mongo启动停止 26](#_Toc3250)

[26) 检查jar包是否启动 26](#_Toc23568)

[运营篇 26](#_Toc23920)

[系统初始化配置 26](#_Toc20695)

[27) 登录管理后台 26](#_Toc32321)

[28) 系统配置 27](#_Toc21654)

[29) 币种配置 31](#_Toc5514)

[30) 第3方App配置 32](#_Toc10179)

[31) 平台地址配置 33](#_Toc20232)

[32) 设置承兑商 34](#_Toc26586)

[系统功能手册 34](#_Toc14384)

[33) 承兑商工作流程 35](#_Toc25279)

[34) 第3方app充值提现流程 35](#_Toc3215)

[35) 提币管理 35](#_Toc24090)

[36) 异常或争议订单处理 36](#_Toc23095)

[二次开发篇 38](#_Toc15001)

[说明 38](#_Toc26231)

[系统流程 38](#_Toc23103)

[我方提供的接口 39](#_Toc15730)

[37) 自动登录 39](#_Toc10790)

[38) 充值 39](#_Toc27474)

[39) 提现 39](#_Toc29538)

[第3方需要提供的接口 39](#_Toc5160)

[40) 账号密码查询 39](#_Toc29669)

[41) 充值回调 40](#_Toc1976)

[42) 提现回调 40](#_Toc1936)

[43) 余额查询 41](#_Toc6340)

[44) 发送通知（可选） 41](#_Toc10975)

# 系统简介

点点数字钱包是一款基于波场币（TRX）开发的springboot java版本的数字钱包系统，可以帮助用户快速完成波场币的购买、售卖、提币。同时，为了便于第3方App或Web系统与数字钱包余额相互转化，通过接口，可以查询第3方的余额、使用钱包给第3方充值或从第3方提现到数字货币钱包。

目前系统的主要功能：

1. 用户注册登录、第3方自动登录
2. 波场币充值、提现
3. 波场币转账（提币）
4. 波场币实时汇率
5. 第3方App或Web系统的充值、提现
6. 添加收款方式（支持微信、支付宝、银行）
7. 承兑商订单管理
8. 公告展示

为什么基于波场币开发？

波场币相比较比特币，以太坊交易速度快，交易手续费低的特点，因此选用波场币作为系统货币。

系统特点：

1. springboot+mysql+redis+kafka+nginx开发、部署简单。
2. 文档齐全
3. 系统完整
4. 功能简洁明了

系统组成

1. 后台管理（Api+管理后台）
2. H5系统

后续版本功能

1. 支持更多类型数字货币（USDT/BTC/ETH）等。
2. 币币交换
3. Pc web系统

文档组成

1. 部署篇

本章主要介绍服务器软硬件配置、准备材料、部署流程、开机启动、进程守护等操作。

1. 运营篇

本章主要站在运营的角度介绍使用系统的主要流程和操作步骤。

1. 二次开发篇

本章主要介绍第3方App或Web系统对接本系统。

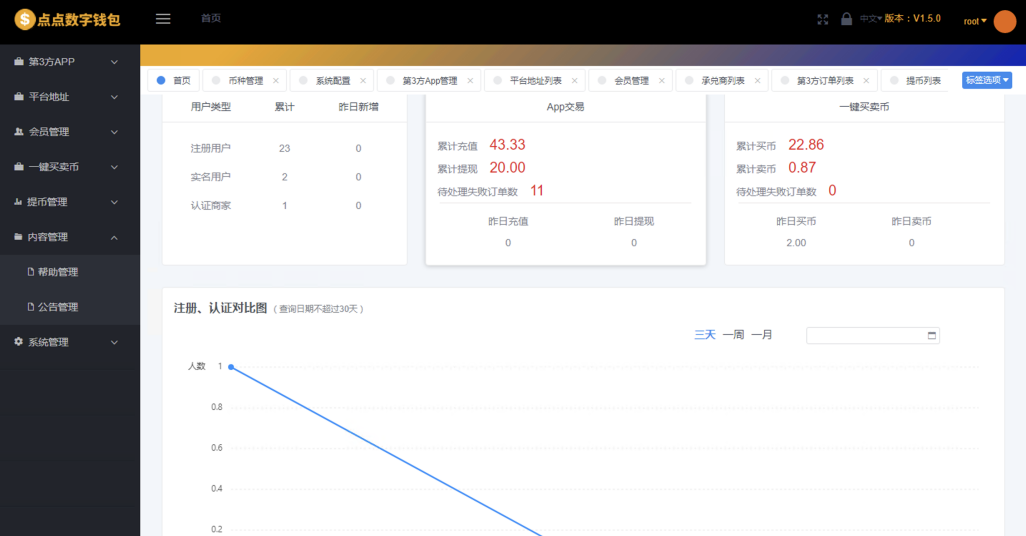
前端页面部分截图

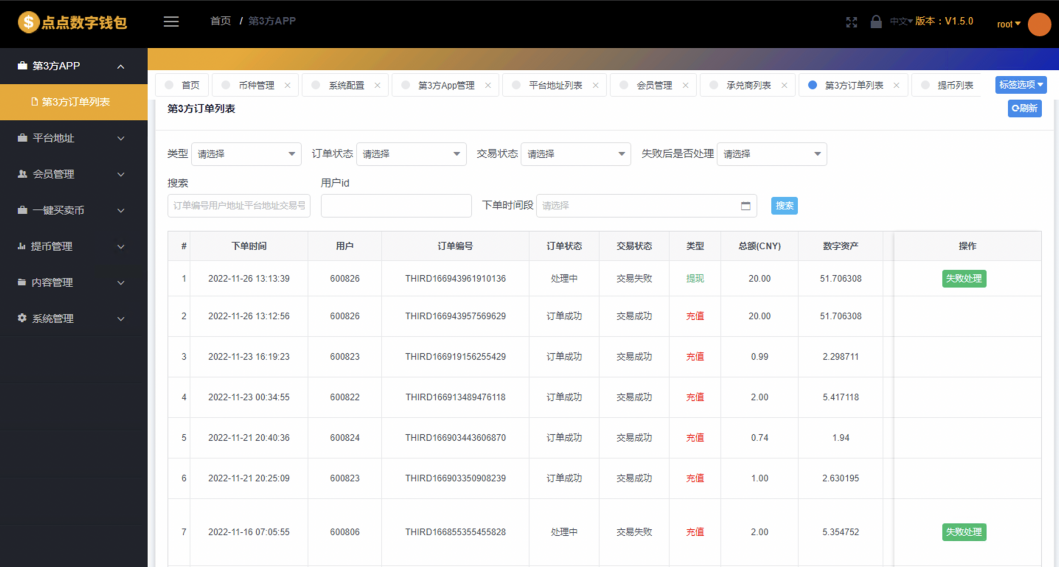


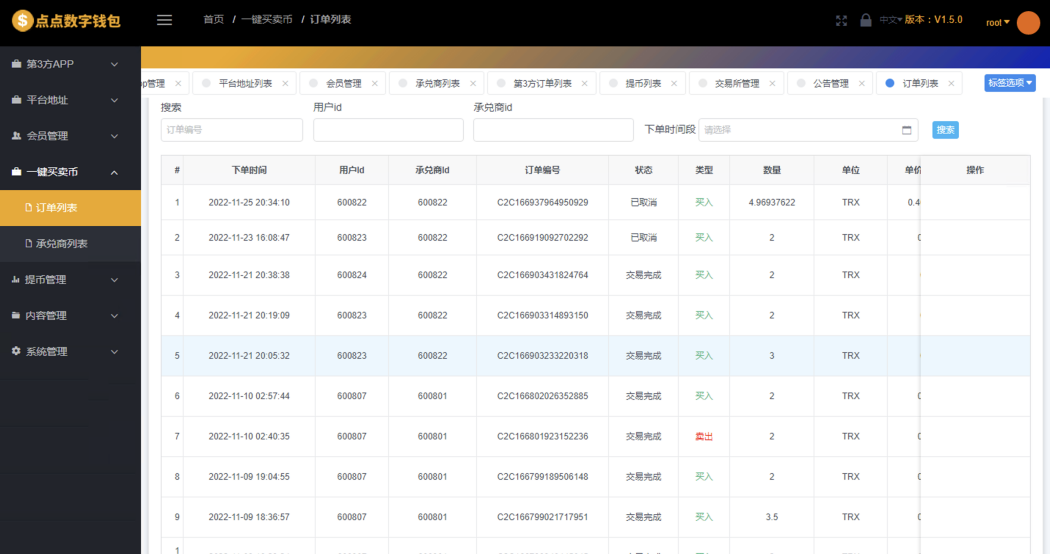


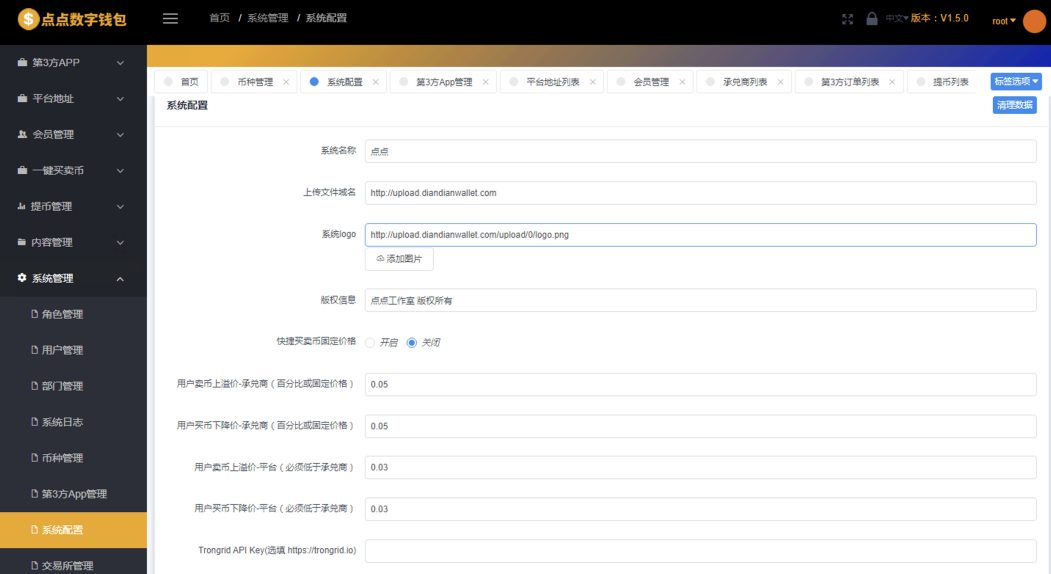


管理后台页面部分截图









更多内容请参考pdf文档。

免责声明

本系统主要用于大家对数字货币的研究和学习，各位看官请勿用于非法用途。如有违规，本作者概不负责。

# 部署篇

## 资料准备

### 短信申请-飞鸽

<https://www.feige.cn/>

1. 注册账号，并进行实名认证，并需要公司营业执照。
2. 创建签名，需要使用公司名字缩写。
3. 购买短信条数，测试时可以购买免费50条。



1. 系统配置信息



其中，短信网址填写【接口地址】、短信用户名填写【账号】、短信密码填写【秘钥】、短信签名填写飞鸽【签名管理】【签名id】，注意是签名id，不是签名内容。

### 邮箱申请

1. 注册outlook邮箱

https://outlook.com/

1. 登录outlook，点击右上角按钮开启smtp。





1. 系统配置

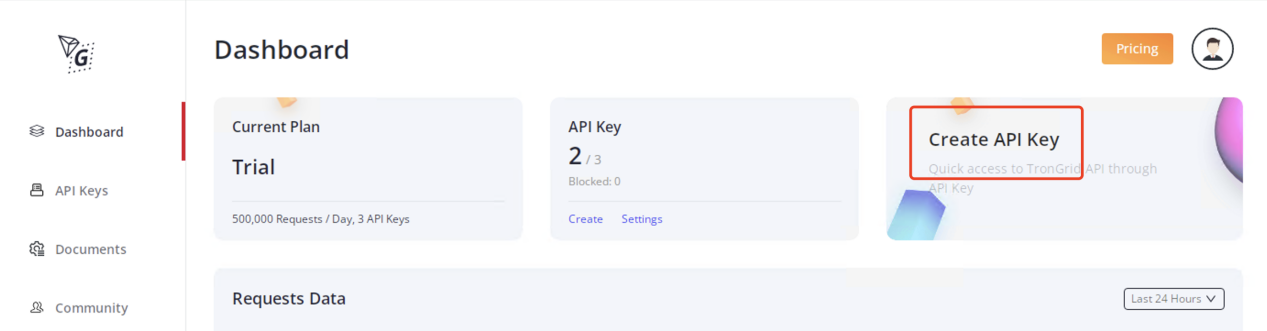


### trongrid申请（选填）

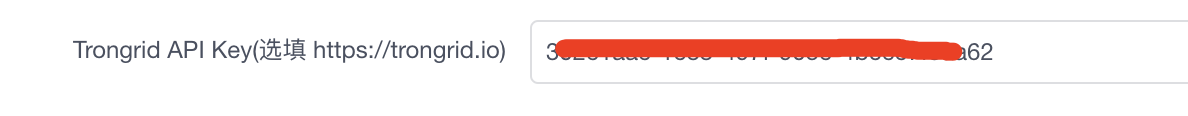
<https://www.trongrid.io/>

主要完成波场币的交易、查询接口。不配置也可以，只是请求次数高会被部分限制。

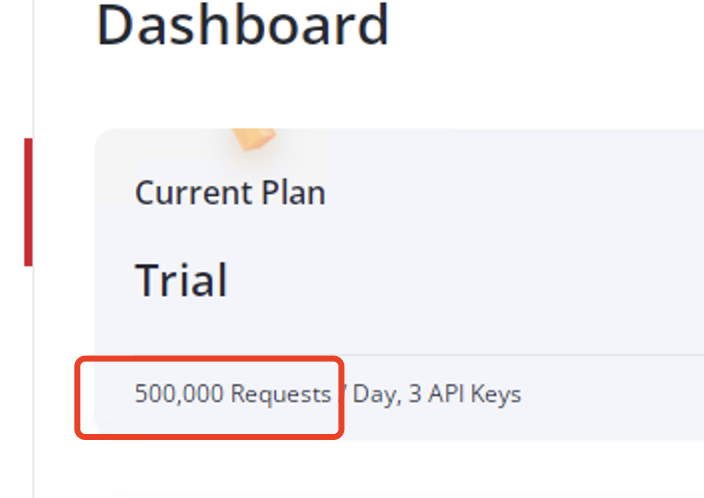
1. 注册账号
2. 创建一个apikey



1. 拷贝apikey设置到后台



1. 默认apikey限制是10万次一天，如果你需要更多访问次数，则可以连接钱包自动增加到50万次每天。如果不设置，接口调用访问量大时，会被限制。



## 系统部署

### 服务器配置

阿里云Centos7.8 64位 4核8G内存100G硬盘

### 域名配置

解析域名api、admin、upload、h5指向服务器外网ip。

确保80端口已开放，默认阿里云服务器开启了该端口。

### 依赖环境

使用ssh连接服务器

```

yum install -y wget

yum install -y vim-enhanced

yum install -y make cmake gcc gcc-c++

yum install -y pcre pcre-devel

yum install -y zlib zlib-devel

yum install -y openssl openssl-devel

```

### 下载安装包

下载安装包地址

<https://www.mediafire.com/folder/a7bylk7wvtnfl/Documents>

下载github相关

<https://github.com/diandianwallet/wallet>

### 基础目录创建

创建基础目录。

mkdir -p /opt/bitcoin/

cd /opt/bitcoin/

上传刚才下载的安装包到【/opt/bitcoin】目录下。

### Jdk安装

解压

cd /opt/bitcoin && tar xf jdk-linux-x64.tar.gz && mkdir -p java && yes | cp -rf jdk1.8.0\_131 ./java/ && rm -rf jdk1.8.0\_131

#配置jdk路径

grep 'JAVA\_HOME' /etc/profile

if [ $? -ne 0 ]; then

cat>>/etc/profile<<EOF

export JAVA\_HOME=/opt/bitcoin/java/jdk1.8.0\_131

export JRE\_HOME=/opt/bitcoin/java/jdk1.8.0\_131/jre

export PATH=\$JAVA\_HOME/bin:\$JRE\_HOME/lib:\$PATH

export CLASSPATH=.:\$JAVA\_HOME/lib/dt.jar:\$JAVA\_HOME/lib/tools.jar

EOF

fi

设置软连接

ln -fs /opt/bitcoin/java/jdk1.8.0\_131/bin/\* /usr/bin/ && ln -fs /opt/bitcoin/java/jdk1.8.0\_131/bin/\* /usr/sbin/ && ln -fs /opt/bitcoin/java/jdk1.8.0\_131/bin/\* /usr/local/bin/ && ln -fs /opt/bitcoin/java/jdk1.8.0\_131/bin/\* /usr/local/sbin/

source /etc/profile

查看安装是否成功

java -version

### nginx安装

执行下面命令安装：

sudo yum install -y epel-release

sudo yum install -y nginx

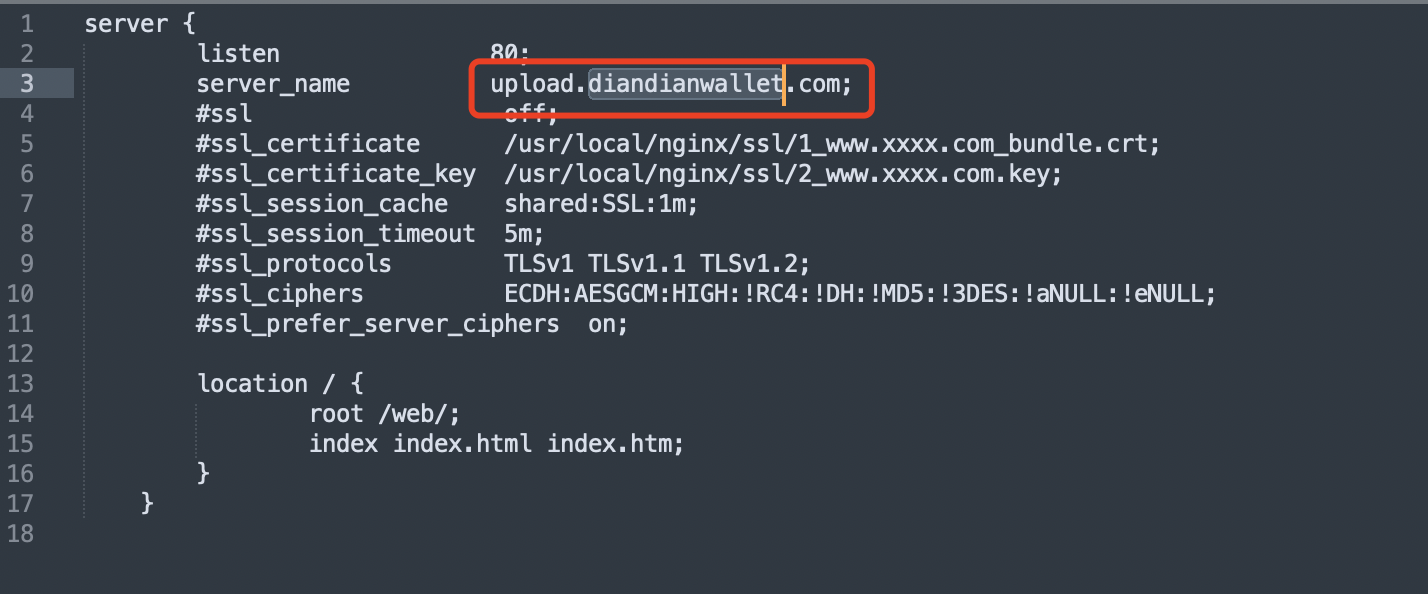
安装成功后，默认的网站目录为： /usr/share/nginx/html

默认的配置文件为：/etc/nginx/nginx.conf

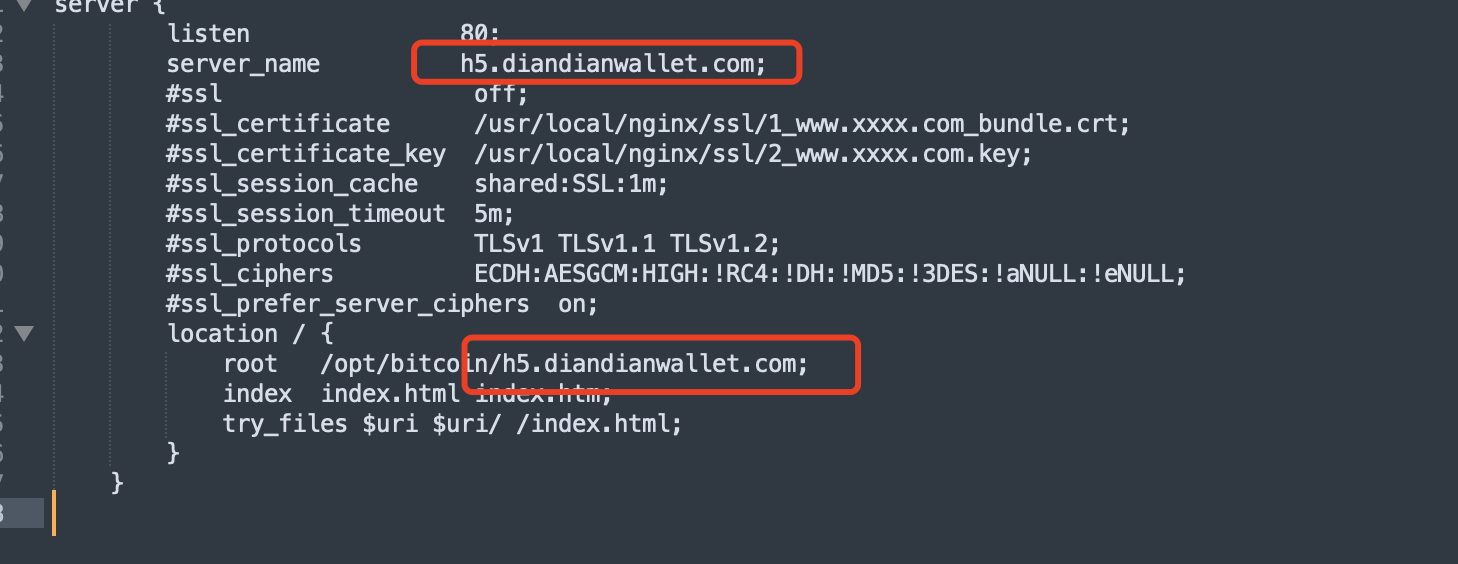
自定义配置文件目录为: /etc/nginx/conf.d/

修改【03\_nginx】下配置文件的域名为自己的域名。

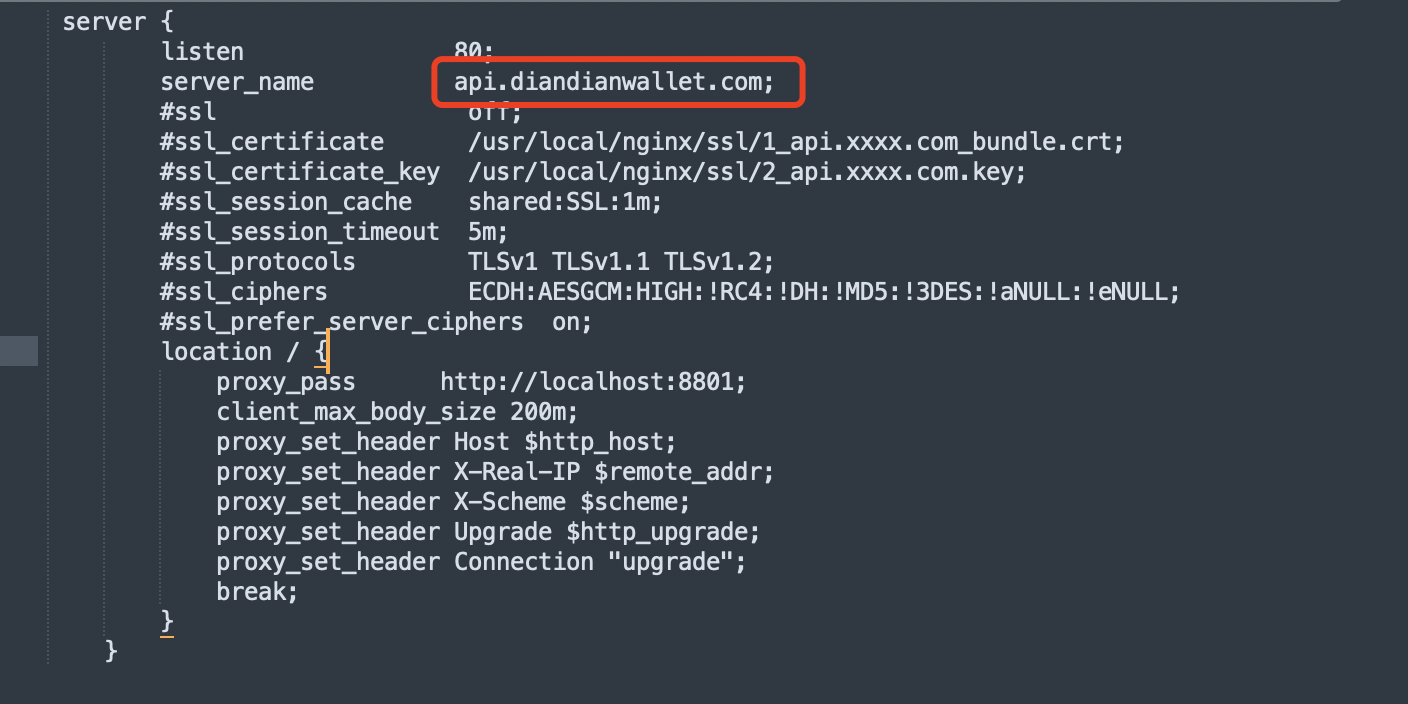
#修改upload.conf



#修改h5.conf



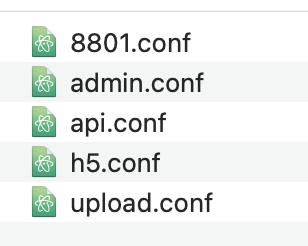
#修改api.conf



#修改admin.conf



上传修改好的【03\_nginx】下的所有文件到服务器【/etc/nginx/conf.d/】目录下。



修改开机启动内容

cd /lib/systemd/system

vi nginx.service

修改为下面的内容

ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf

重启nginx

nginx -c /etc/nginx/nginx.conf

### mysql安装及配置

执行下面命令安装：

cd /opt/bitcoin && rpm -Uvh mysql-community-release-el6-5.noarch.rpm

yum install mysql-community-server

安装过程选择【y】即可

启动mysql

service mysqld start

安全设置

mysql\_secure\_installation

输入当前密码：默认为空，按回车键

是否设置root密码：Y

输入新密码(请把密码改成自己的实际密码) Root123456

其他设置：默认回车即可

初始化数据库

mysql -u root -p

输入数据库密码

create database if not exists bitrade default character set = 'utf8';

退出mysql，输入下面命令

exit

mysql忽略大小写

vi /etc/my.cnf

#在配置文件的[mysqld]节点下，加入一行：

lower\_case\_table\_names=1

#重启mysql

systemctl restart mysqld.service

授权允许远程访问（无需执行，需要远程连接才开启）

mysql -u root -p

grant all privileges on \*.\* to 'root'@'%' identified by 'Root123456';

刷新授权

flush privileges;

### redis安装

安装依赖

yum -y install cpp binutils glibc glibc-kernheaders glibc-common glibc-devel gcc make

安装

wget http://rpms.famillecollet.com/enterprise/remi-release-7.rpm && yum -y install remi-release-7.rpm && yum -y --enablerepo=remi install redis

设置redis密码

vi /etc/redis/redis.conf

修改密码（密码建议改成自己的）

requirepass pass1234

设置远程访问(无需设置，远程连接才需要开启)

修改redis.conf文件：

注释 bind 127.0.0.1

修改 protected-mode no

#自启动

systemctl enable redis

#启动redis

systemctl start redis

### 安装 Zookeeper

解压

cd /opt/bitcoin && tar -xzvf apache-zookeeper-3.6.3-bin.tar.gz

配置

cd apache-zookeeper-3.6.3-bin

mkdir data

cd conf

mv zoo\_sample.cfg zoo.cfg

启动

/opt/bitcoin/apache-zookeeper-3.6.3-bin/bin/zkServer.sh start

### 安装 Kafka

安装kafka之前必须先安装zookeeper。

下载安装

cd /opt/bitcoin && tar -zxvf kafka\_2.12-3.1.2.tgz

cd kafka\_2.12-3.1.2

修改配置

vi ./config/server.properties

修改为如下内容

listeners=PLAINTEXT://{本机内网IP}:9092

advertised.listeners=PLAINTEXT://{本机外网IP}:9092

启动zookeeper

/opt/bitcoin/kafka\_2.12-3.1.2/bin/zookeeper-server-start.sh /opt/bitcoin/kafka\_2.12-3.1.2/config/zookeeper.properties &

#启动成功后，使用netstat -ntlp可以查看到2181端口已运行

启动kafka

/opt/bitcoin/kafka\_2.12-3.1.2/bin/kafka-server-start.sh /opt/bitcoin/kafka\_2.12-3.1.2/config/server.properties &

#启动成功后，使用netstat -ntlp可以查看到9092端口已运行

### 安装MongoDB

安装

cd /opt/bitcoin && tar zxvf mongodb-linux-x86\_64-amazon-3.6.4.tgz && mv ./mongodb-linux-x86\_64-amazon-3.6.4 ./mongodb &&

cd ./mongodb

## 安装准备

创建数据库文件夹与日志文件、配置文件：

mkdir ./data && touch ./mongod.log && touch ./mongodb.conf

mongodb.conf添加配置文件

vi mongodb.conf

内容如下：

dbpath=/opt/bitcoin/mongodb/data

logpath=/opt/bitcoin/mongodb/mongod.log

logappend=true

port=27017

fork=true

auth=true

journal=true

bind\_ip=0.0.0.0

启动

/opt/bitcoin/mongodb/bin/mongod --config /opt/bitcoin/mongodb/mongodb.conf

## 用户授权

mongodb安装好后没有任何用户，需要创建用户并授权

cd /opt/bitcoin/mongodb && ./bin/mongo

执行下面命令

use admin

添加管理用户（mongoDB 没有无敌用户root，只有能管理用户的用户 userAdminAnyDatabase）密码改成自己的。

db.createUser( {user: "admin",pwd: "root",roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]})

#认证，返回1表示成功

db.auth("admin","root")

验证之后还是做不了操作，因为admin只有用户管理权限，下面创建用户，用户都跟着库走

```

use bitrade

db.createUser({user: "root",pwd: "root",roles: [{ role: "readWrite", db: "bitrade" }]})

```

使用创建的用户root登录进行数据库操作

```

db.auth("root","root")

show collections

```

按【Ctrl+c】退出mongodb命令模式。

测试登录(无需执行)

./bin/mongo bitrade --port 27017 -u root -p root

### 开机启动

进入目录

cd /lib/systemd/system

配置Zookeeper & kafka

vim zookeeper.service

内容如下：

[Unit]

Description=Zookeeper service

After=network.target

[Service]

Type=simple

Environment="PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/jdk/bin"

User=root

Group=root

ExecStart=/opt/bitcoin/kafka\_2.12-3.1.2/bin/zookeeper-server-start.sh /opt/bitcoin/kafka\_2.12-3.1.2/config/zookeeper.properties

ExecStop=/opt/bitcoin/kafka\_2.12-3.1.2/bin/zookeeper-server-stop.sh

Restart=on-failure

[Install]

WantedBy=multi-user.target

配置kafka

vim kafka.service

内容如下：

[Unit]

Description=Apache Kafka server (broker)

After=network.target zookeeper.service

[Service]

Type=simple

Environment="PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/jdk/bin"

User=root

Group=root

ExecStart=/opt/bitcoin/kafka\_2.12-3.1.2/bin/kafka-server-start.sh /opt/bitcoin/kafka\_2.12-3.1.2/config/server.properties

ExecStop=/opt/bitcoin/kafka\_2.12-3.1.2/bin/kafka-server-stop.sh

Restart=on-failure

[Install]

WantedBy=multi-user.target

执行以下命令：

systemctl daemon-reload

systemctl enable zookeeper

systemctl enable kafka

systemctl start zookeeper

systemctl start kafka

Mongodb配置

vi mongodb.service

内容如下：

[Unit]

Description=mongodb

After=network.target remote-fs.target nss-lookup.target

[Service]

Type=forking

ExecStart=/opt/bitcoin/mongodb/bin/mongod --config /opt/bitcoin/mongodb/mongodb.conf

ExecReload=/bin/kill -s HUP $MAINPID

ExecStop=/opt/bitcoin/mongodb/bin/mongod --shutdown --config /opt/bitcoin/mongodb/mongodb.conf

PrivateTmp=true

[Install]

WantedBy=multi-user.target

执行以下命令：

systemctl daemon-reload

systemctl enable mongodb

systemctl start mongodb

Jar配置

vi /etc/rc.d/init.d/jar\_auto.sh

内容如下：

#!/bin/bash

#配置jdk的路径

export JAVA\_HOME=/opt/bitcoin/java/jdk1.8.0\_131

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export PATH=${JAVA\_HOME}/bin:$PATH

sleep 10

nohup java -Xms256m -Xmx512m -javaagent:"/opt/bitcoin/sjt\_agent.jar" -jar /opt/bitcoin/cloud.jar --spring.config.location=/opt/bitcoin/config/cloud/application.properties >/opt/bitcoin/logs/cloud.log>&1 &

nohup java -Xms256m -Xmx512m -javaagent:"/opt/bitcoin/sjt\_agent.jar" -jar /opt/bitcoin/ucenter-api.jar --spring.config.location=/opt/bitcoin/config/ucenter-api/application.properties >/opt/bitcoin/logs/ucenter-api.log>&1 &

nohup java -Xms256m -Xmx512m -javaagent:"/opt/bitcoin/sjt\_agent.jar" -jar /opt/bitcoin/admin-api.jar --spring.config.location=/opt/bitcoin/config/admin-api/application.properties >/opt/bitcoin/logs/admin-api.log>&1 &

#赋予执行权限

chmod +x /etc/rc.d/init.d/jar\_auto.sh

#编辑/etc/rc.d目录下的rc.local文件

vim /etc/rc.d/rc.local

#添加jar自启动脚本所在的目录，添加到rc.local文件末尾，内容如下：

/etc/rc.d/init.d/jar\_auto.sh

/usr/sbin/nginx -c /etc/nginx/nginx.conf

#将jar自启动目录添加到rc.local文件后，需要为文件重新赋予执行权限，执行以下命令：

chmod +x /etc/rc.d/rc.local

### api部署

上传【admin-api.jar】【cloud.jar】【ucenter-api.jar】【sjt\_agent.jar】到服务器【/opt/bitcoin】目录下

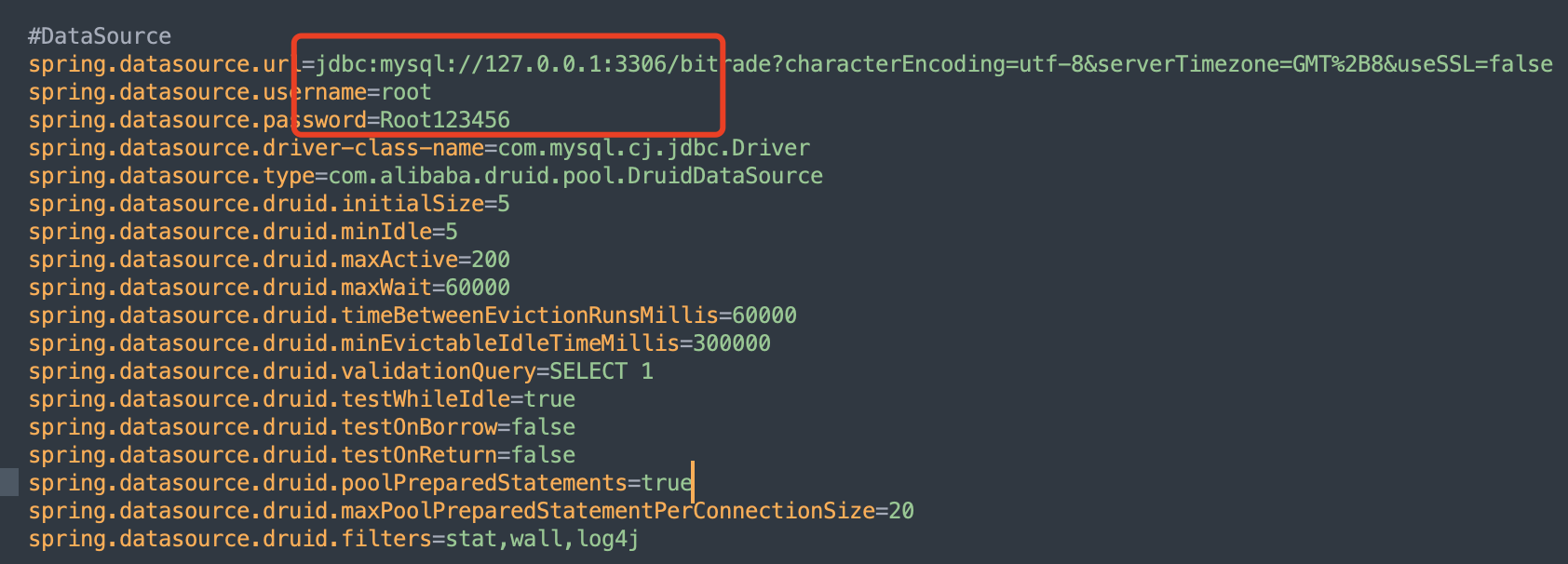
使用文本工具修改配置文件

#### /config/cloud/application.properties配置

无需修改

#### /config/admin-api/application.properties配置

数据库配置修改



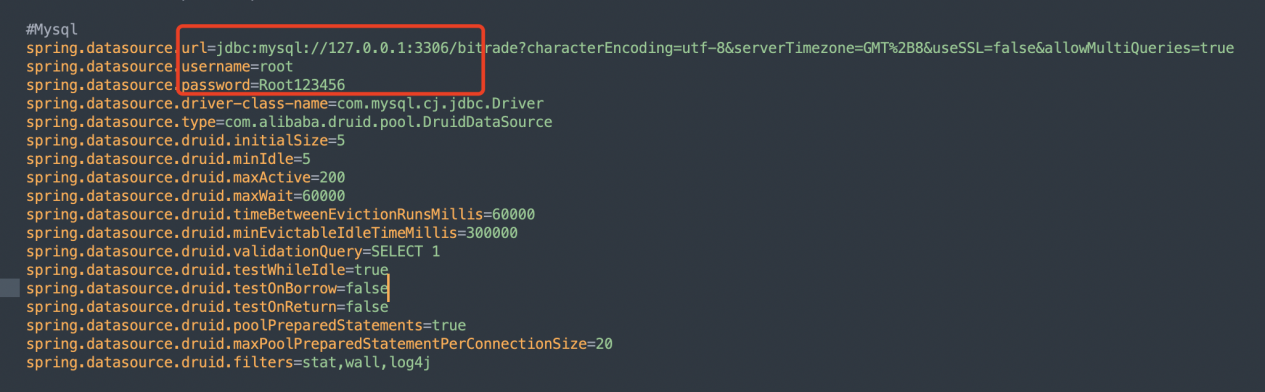
Redis修改



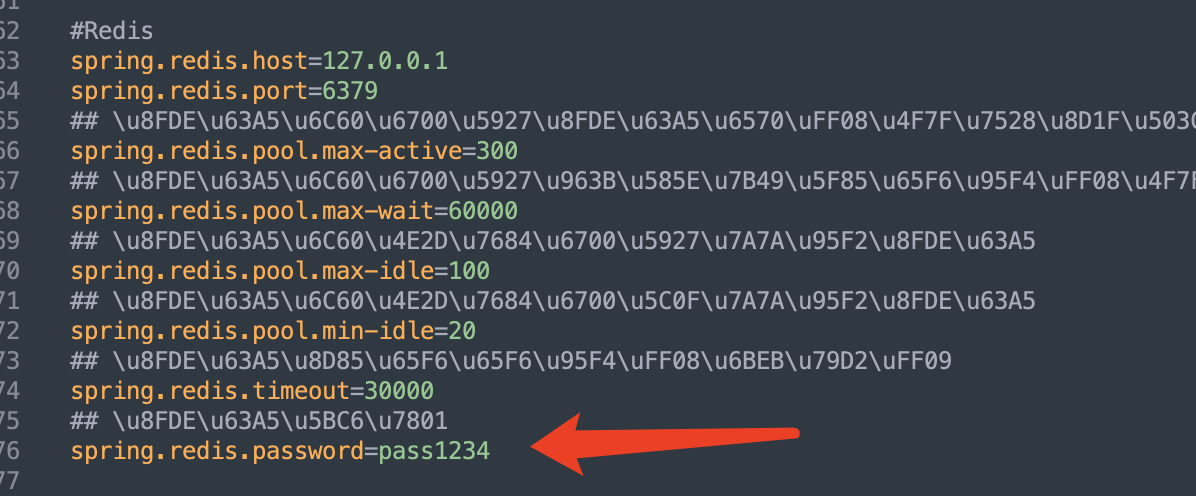
其他保持默认即可

#### /config/ucenter-api/application.properties配置

数据库配置



Redis配置



其他保持默认即可

上传修改好的配置文件【config】整个目录到【/opt/bitcoin】目录下

创建日志文件目录并赋予执行权限

cd /opt/bitcoin && mkdir logs && chmod -R 777 /opt/bitcoin/logs

### 管理后台开发部署

注意，只是修改【admin.diandianwallet.com】下的内容

使用文本工具修改配置

修改管理后台标题

搜索【点点】改成自己的名称。（只有2处）

替换图标和图片

编辑【dist】文件夹下面的ico图标和png图片为自己的。

修改api接口（只有1处）

搜索【api.diandianwallet.com】开头的域名，改成自己的域名地址。

修改【admin.diandianwallet.com】目录为自己的域名目录，比如【admin.test.com】

上传修改的整个文件夹到【/opt/bitcoin】目录下

注意：修改的名称需要跟nginx配置下的admin.conf名称一致

### H5开发部署

注意，只是修改【h5.diandianwallet.com】下的内容

修改api接口（只有1处）

搜索【api.diandianwallet.com】开头的域名，改成自己的域名地址。

修改【h5.diandianwallet.com】目录为自己的域名目录，比如【h5.test.com】

上传整个文件夹到【/opt/bitcoin】目录下

注意：修改的名称需要跟nginx配置下的h5.conf名称一致

### 数据库部署

使用文本工具，编辑【02\_sql】目录下的sql文件

搜索【upload.diandianwallet.com】替换成你自己的上传域名，比如【upload.xxx.com】

上传sql文件【/opt/bitcoin/】目录下，执行以下命令导入sql文件：

mysql -uroot -p

输入数据库密码

切换数据库

use bitrade;

导入

source /opt/bitcoin/bitrade.sql

查看表是否导入成功

show tables;

退出mysql

exit

### 上传文件目录

创建上传文件目录并赋予执行权限

mkdir /web && chmod -R 777 /web

上传【01\_deploy】下的文件夹【upload】到【/web】目录下

### 进程守护

安装supervisor

yum install -y supervisor

开机自启

systemctl enable supervisord

启动supervisord

systemctl start supervisord

查看状态(无需执行)

systemctl status supervisord

编辑cloud配置文件

vi /etc/supervisord.d/cloud.ini

内容如下：

[program:cloud]

Command=java -Xms256m -Xmx512m -javaagent:"/opt/bitcoin/sjt\_agent.jar" -jar /opt/bitcoin/cloud.jar --spring.config.location=/opt/bitcoin/config/cloud/application.properties >/opt/bitcoin/logs/cloud.log>&1 &

directory=/opt/bitcoin

autostart=true

;;autorestart=false

stderr\_logfile=/tmp/cloud\_stderr.log

stdout\_logfile=/tmp/cloud\_stdout.log

编辑ucenter配置文件

vi /etc/supervisord.d/ucenter.ini

内容如下：

[program:ucenter]

command=java -Xms256m -Xmx512m -javaagent:"/opt/bitcoin/sjt\_agent.jar" -jar /opt/bitcoin/ucenter-api.jar --spring.config.location=/opt/bitcoin/config/ucenter-api/application.properties >/opt/bitcoin/logs/ucenter-api.log>&1 &

directory=/opt/bitcoin

autostart=true

;;autorestart=false

stderr\_logfile=/tmp/ucenter\_stderr.log

stdout\_logfile=/tmp/ucenter\_stdout.log

编辑admin配置文件

vi /etc/supervisord.d/admin.ini

内容如下：

[program:admin]

command=java -Xms256m -Xmx512m -javaagent:"/opt/bitcoin/sjt\_agent.jar" -jar /opt/bitcoin/admin-api.jar --spring.config.location=/opt/bitcoin/config/admin-api/application.properties >/opt/bitcoin/logs/admin-api.log>&1 &

directory=/opt/bitcoin

autostart=true

;;autorestart=false

stderr\_logfile=/tmp/admin\_stderr.log

stdout\_logfile=/tmp/admin\_stdout.log

重启supervisord

systemctl restart supervisord

### 重启系统

系统重启后，稍等几分钟，系统自动启动所有程序。输入下面命令：

reboot

至此，系统部署完成。

## 常用操作命令

### zookeeper启动停止

启动

/opt/bitcoin/apache-zookeeper-3.6.3-bin/bin/zkServer.sh start

停止

/opt/bitcoin/apache-zookeeper-3.6.3-bin/bin/zkServer.sh stop

### mongo启动停止

启动

/opt/bitcoin/mongodb/bin/mongod --config /opt/bitcoin/mongodb/mongodb.conf

停止

/opt/bitcoin/bin/mongod --shutdown --dbpath=/opt/bitcoin/mongodb/data/

### 手动启动jar包

java -Xms256m -Xmx512m -javaagent:"/opt/bitcoin/sjt\_agent.jar" -jar /opt/bitcoin/cloud.jar --spring.config.location=/opt/bitcoin/config/cloud/application.properties >/opt/bitcoin/logs/cloud.log>&1 &

java -Xms256m -Xmx512m -javaagent:"/opt/bitcoin/sjt\_agent.jar" -jar /opt/bitcoin/ucenter-api.jar --spring.config.location=/opt/bitcoin/config/ucenter-api/application.properties >/opt/bitcoin/logs/ucenter-api.log>&1 &

java -Xms256m -Xmx512m -javaagent:"/opt/bitcoin/sjt\_agent.jar" -jar /opt/bitcoin/admin-api.jar --spring.config.location=/opt/bitcoin/config/admin-api/application.properties >/opt/bitcoin/logs/admin-api.log>&1 &

### 检查jar包是否启动

#cloud.jar端口7000

#admin-api.jar端口6001

#ucenter-api.jar端口6010

ps -ef | grep cloud.jar

# 运营篇

## 系统初始化配置

### 登录管理后台

<http://admin.diandianwallet.com>

默认超级管理员账号密码

root/123456

客服（工作人员）账号

demo/123456

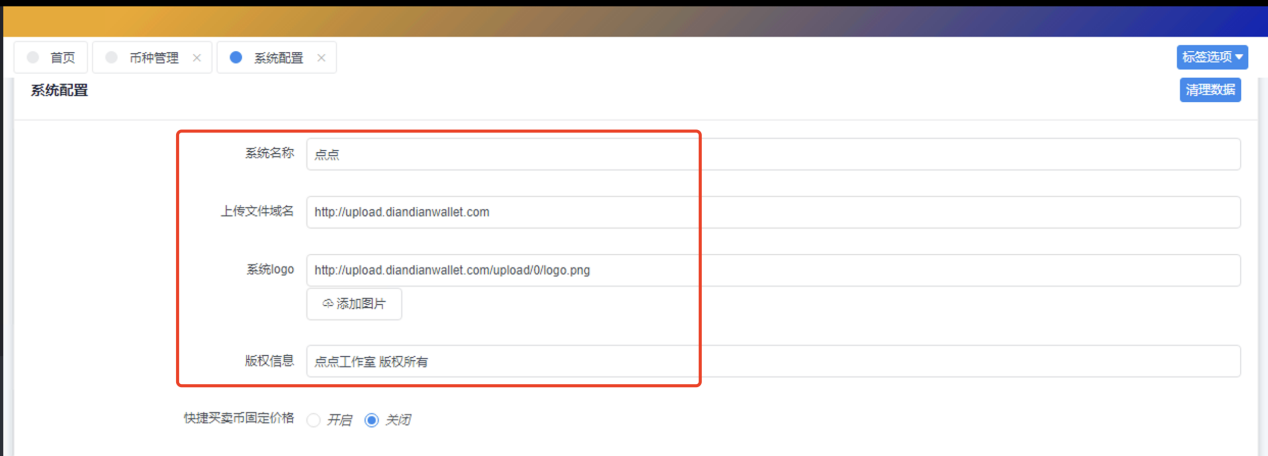
登录后，建议修改密码。

同时，也可以创建更多的角色、账号等。默认相关权限已经配置好，不需要再配置。

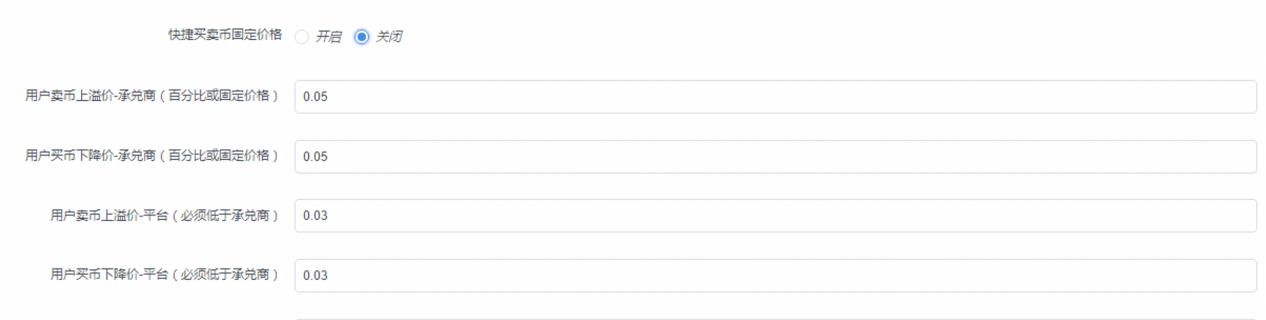


### 系统配置

1. 配置系统名称、域名、logo、版本信息



1. 快捷交易配置。



快捷买卖币固定价格：关闭就是按比例，开启则是固定价格。建设设置按比例，因为每天数字货币价格波动较大。

用户卖币上溢价-承兑商（百分比或固定价格）：

承兑商卖币给用户时，相对于汇率上浮比例。

用户买币下降价-承兑商（百分比或固定价格）：

用户卖币给承兑商时，相对于汇率下降比例。

用户卖币上溢价-平台（必须低于承兑商）：

承兑商卖币给用户时，平台与承兑商分成比例。比如，上面配置说明承兑商收取5个点，其中3个点分成给平台。注意，技术交易费会收取平台的2个点，平台分成建议不要低于2个点。比例必须低于承兑商的比例。

用户买币下降价-平台（必须低于承兑商）：

同上概念。

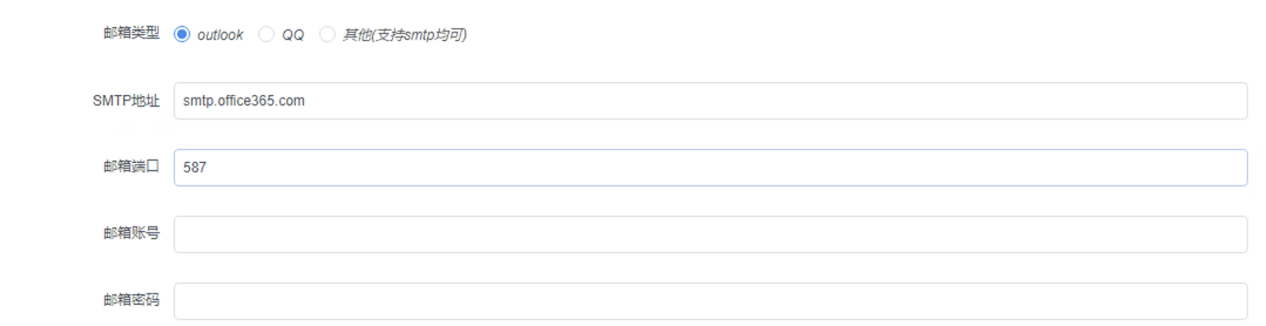
1. trongrid配置

配置后，每天查询波场币交易次数可以增加。具体参考申请资料中相关说明。



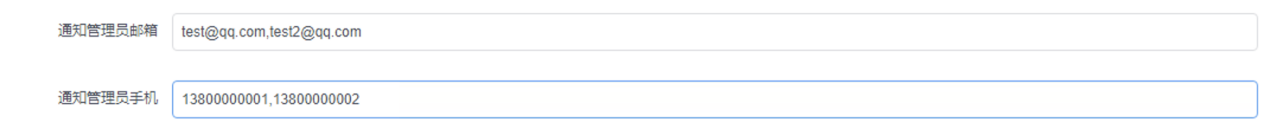
1. 邮箱配置

建议申请outlook邮箱配置，配置该账号，系统中一些需要邮件通知地方，会使用该账号信息。具体申请流程，参考申请资料。



1. 配置管理员邮箱和手机号

多个账号可以英文【,】分隔,关键信息将会以邮箱或手机短信通知管理员，比如管理后台登录，平台提币等。



1. 短信配置

建议使用飞鸽短信，具体参考申请资料。



1. 开关配置



是否快速买卖币：

h5端是否显示快捷买卖币。如果本平台不开启快捷买卖币，就可以关闭。

H5返回模拟验证码（正式运营请关闭）：

在没配置短信通道时，开启该选项，注册、登录、获取验证码地方，均会模拟返回给用户。正式运营，请勿开启返回验证码。

是否短信通知：

是否开启发送短信功能。关闭，则即使配置来短信通道，也不会发送短信。

是否IM通知(第3方App需要支持IM通知)：

如果对接了第3方im，开启了，则会通知第3方。

是否邮件通知：

是否开启发送邮件。关闭，则即使配置了邮件通道，也不会发送邮件。

是否开启交易所：

H5端是否显示交易所按钮。

是否开启第3方：

系统是否支持第3方对接，关闭则不支持。

H5显示注册按钮

H5是否显示注册按钮。关闭了则用户不能主动注册，可以通过第3方同步登录进入系统。

1. 第3方提现审核



第3方app开启且配置了，则用户app提现是不是需要审核。

是否App提现审核（开启所有提现需审核）：

开启表示提现每笔都需要提现。

超过金额审核（元）：

配置了超过0的金额，表示超过多少元需要审核，否则无需审核。

1. 背景图及教程配置



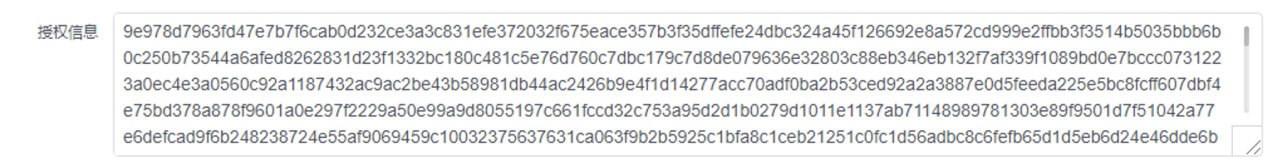
配置h5端显示的背景图片，默认是淡蓝色图片。

具体教程内容，可以在帮助中编辑。



H5端对应的提示文字内容配置，可以根据平台实际运营配置文本内容。

1. 授权信息（可选）



该选项配置可选，不配置系统会按以下比例收取成功交易手续费。

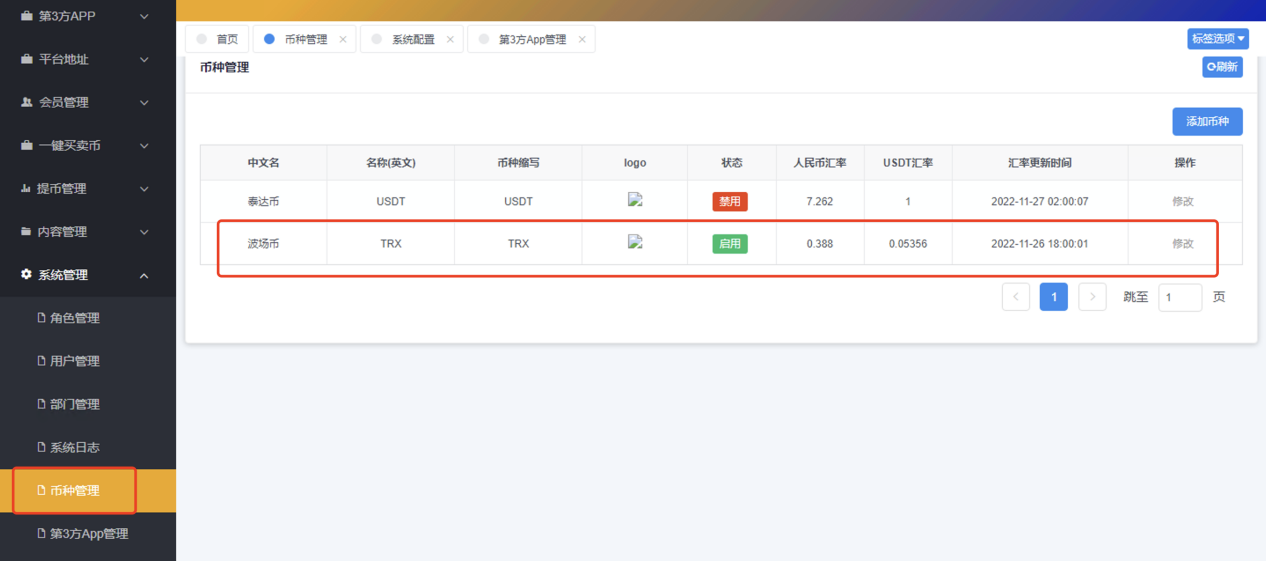
第3方app充值提现：0.03

一键买卖币：0.02

用户提币：0.01

### 币种配置

目前只支持波场币，其他币种请勿开启。后续版本将支持更多币种。



其中，注意汇率最新更新时间和价格。默认每几分钟更新一次汇率。如果更新不通畅，可以手动修改。



以下对币种进行一些配置。



充值(买入)：是否可以快捷买币

提现(卖出)：是否可以快捷卖币

状态：是否开启该币种

最小充值(买入)数: 用户快捷买币最小数量

提币最小手续费(固定 0不限制):用户转币最小手续费

提币最大手续费(固定 0不限制):用户转币最大手续费

提币手续费比例:默认转币比例

最小提币(卖出)量:快捷卖币最小量

最大提币(卖出)量:快捷卖币最大量

### 第3方App配置

如果你要通过系统将数字资产转化成自己系统上面的金额，则需要第3方app配置。



默认只能添加1个，将来会同时支持多个。

开启来第3方，需要自己的业务系统实现配置中的接口，便于本系统交互数据。



App的关键字，只能是英文且唯一：随便起个英文名字

是否开启：是否开启

App的名字：该名字会显示到h5端

APP授权码：本系统与第3方业务系统交互时需要的授权码

提现手续费百分比(0~100)：用户app提现手续费比例

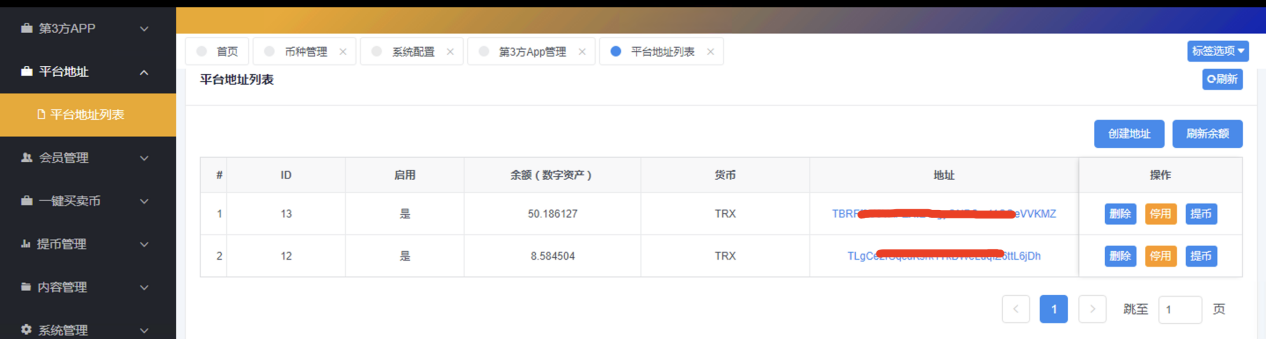
最低提现费用（元）：用户app提现最低手续费

自动登录时，是否查询第3方登录：h5自动登录时需不需要去第3方系统认证登录。

更多接口配置，需要了解文档下面的开发篇内容。

### 平台地址配置

在平台地址列表会显示你当前系统总的虚拟资产：



默认系统，没有平台地址，建议创建1-2个。



刷新余额：重新获取该地址下面的最新余额。

删除：删除该地址之前，一定要把自己对应地址的金额提走，否则，一旦删除无法找回。

停用：该地址停用后，则交易时不再使用该地址。

提币：将地址中的数字资产可以转账到其他地址。

### 设置承兑商

#### 承兑商配置

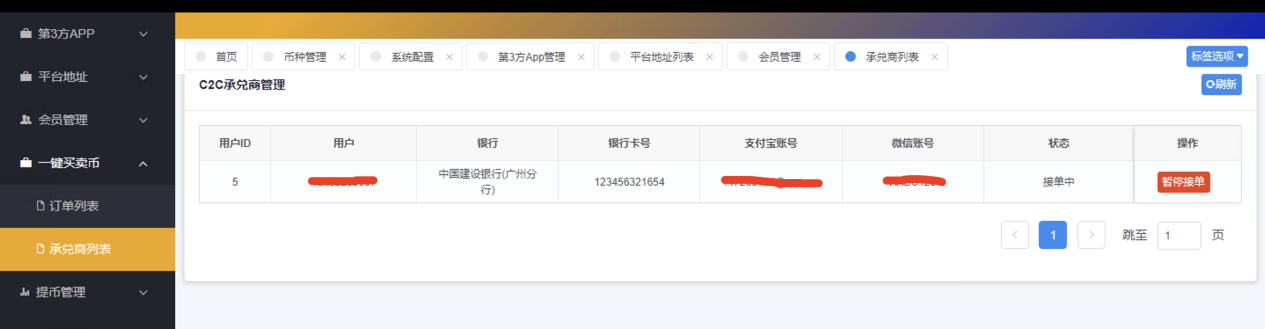
承兑商在主要出售波场币给用户或购买用户的波场币，赚钱部分的手续费。

在会员管理列表，可以授权某个用户为承兑商。

授权承兑商之前，最好要求承兑商转账给平台部分押金，一旦用户与承兑商发生争议，平台可以从押金扣除转账给用户。



在承兑商列表中，展示目前的承兑商：



## 系统功能手册

### 承兑商工作流程

1. 注册普通用户
2. 设置收款方式，最好银行、微信、支付宝都设置。如果没有设置，用户下单时，就会过滤该承兑商。
3. 线下缴纳押金给平台，平台授权该用户为承兑商。
4. 该用户登录h5，会显示承兑商栏目。
5. 用户一键买币，系统会自动随机分配订单给承兑商，且承兑商自动接单。
6. 用户按订单提示，支付款项给承兑商，上传支付截图并标记已付款。系统会自动短信或im通知承兑商。
7. 承兑商确认放币，用户收到对应的数字资产。如果承兑商30分钟内没有放币，用户可以发起申诉。如果用户未付款，但标记已付款，则承兑商也可以发起申诉。产生申诉，系统自动通知管理员。申诉订单处理，参考【异常或争议订单处理】处理方式。
8. 订单交易成功，系统自动从承兑商转对应的技术费和平台分成费到平台，完成分成。
9. 用户设置收款方式后且一键卖币，系统会自动随机分配订单给承兑商，等待承兑商接单。系统会自动通知承兑商。用户卖币，系统会自动冻结用户的数字资产，就是将用户数字资产转到平台地址，以防用户赖单。
10. 承兑商接单后，按订单提示线下转账给用户，并上传截图标记已付款。系统自动通知用户。
11. 用户确认放币，系统从平台地址转账给商家的地址，完成交易。如果30分钟用户未放币，商家可以发起申诉，平台后台手动放币给商家。如果商家未付款却标记已付款，用户可以发起申诉。
12. 订单交易成功，系统自动从承兑商转对应的技术费和平台分成费到平台，完成分成。
13. 为了便于平台管理，平台方也可以在承兑商列表，帮助承兑商完成接单、放币等操作。

### 第3方app充值提现流程

1. 确定后台开启来第3方app
2. 确定后台配置来第3方app
3. 确保用户拥有一定的数字资产
4. 用户点击h5 app充值，输入充值金额，系统扣用户数字资产转化为app的余额。在这个过程中，实际是用户的数字资产转账到来平台方的数字资产中。
5. 用户点击h5 app提现，输入提现金额，系统扣用户app的余额为数字资产，在这个过程中，实际是平台的数字资产转账到用户的数字资产。

### 提币管理

主要展示用户的提币记录。

### 异常或争议订单处理

#### 一键买卖币

1. 首页查看还未处理的数量



1. 列表中筛选未处理的订单
2. 争议订单处理

说明：

在进行一键买卖币交易之前，需要承兑商上交一定的押金或数字货币给平台，平台给对应用户开通为承兑商。

用户买币，不冻结承兑商任何数字资产。

用户卖币，需要先冻结用户数字资产，会把用户的卖币数量先转到平台方，用户放币的时候，平台方将数字资产扣除对应的分成后转给承兑商。

* 用户申诉已付款未放币

判断依据：

通过订单状态来判断，如果是未完成，说明承兑商还未放币。如果是已完成，通过交易号连接查看，是否交易成功。如果交易失败，则可能是承兑商余额不足。

非事实处理：

点击【处理失败】，输入理由大概为“商家已放币，注意查收”。

事实处理：

方法一：喊承兑商私下提币给用户

方法二：平台付币给用户，扣除承兑商的对应押金

方法三：如果是承兑商还未放币操作，催促承兑商放币。

* 用户申诉卖币未收到款项

判断依据：

需要承兑商提供买币付款截图或凭据。

事实处理：

方法一：催促承兑商转款

方法二：扣承兑商押金平台支付给用户，通过平台地址手动提币给用户

非事实处理：

点击【处理失败】，输入理由大概为“商家已付款，注意查收”。

* 承兑商申诉未收到款项

判断依据：

喊用户提供付款凭证

事实处理：

方法一：催促用户付款

方法二：取消订单

非事实处理：

点击【处理失败】，输入理由大概为“用户已付款，注意查收”。

* 承兑商申诉已付款用户未放币

判断依据：

通过订单状态来判断，如果是未完成，说明用户还未放币。如果是已完成，通过交易号连接查看，是否交易成功。如果交易失败，则可能是平台方余额不足。

事实处理：

方法一：用户未放币，则催促放币。

方法二：用户已放币，交易失败，则说明是平台方放币给承兑商失败来，平台手动转币给商家。

非事实处理：

点击【处理失败】，输入理由大概为“用户已放币，注意查收”。

#### App交易

1. 首页查看还未处理的数量。之所以交易失败，主要是app充值提现的时候，本系统与第3方系统交互发生了网络错误、接口对接失败或虚拟资产交易失败（比如用户虚拟资产不足或平台方虚拟资产不足）。



交易状态显示交易失败：

一般是用户或平台余额不足或gas不足，就会导致交易失败。线下转账给用户即可。

# 二次开发篇

## 说明

本系统支持第3方app或其他web系统调用，通过数字货币购买，然后给第3方app或web系统完成充值、提现、通知等功能。

## 系统流程

1. 第3方系统传参账号、密码调用本系统h5端的同步登录接口。
2. h5端调用系统后台登录接口。
3. 本系统后台使用该账号、密码从第3方系统查询用户是否存在。
4. 本系统判断用户是否存在，不存在则创建。存在则返回给h5端。
5. 用户在h5端购买数字货币。
6. 用户在h5端发起app充值或第3方主动调起充值页面，输入充值金额，系统扣除用户的数字货币，转换为第3方的人民币。充值成功后，本系统通知第3方系统。
7. 用户在h5端发起app提现或第3方主动调起提现页面，输入提现金额，系统扣除用户在第3方系统的余额转换为数字货币。提现成功后，本系统通知第3方系统。
8. 用户在h5端售卖数字货币。

## 我方提供的接口

### 自动登录

http://h5.diandianwallet.com/?account=xx&password=base64(xxx)

### 充值

http://h5.diandianwallet.com/?account=xx&password=base64(xxx)&url=http://h5.diandianwallet.com/pages/app/recharge/recharge

### 提现

http://h5.diandianwallet.com/?account=xx&password=base64(xxx)&url=http://h5.diandianwallet.com/pages/app/withdraw/withdraw

其中：

account后面跟第3方系统的账号或手机号。

password后面跟第3方系统的密码，需要使用base64编码传给h5。

url后面跟需要跳转的地址，如果带有url参数，自动登录成功后会跳转到该url地址。

注意：实际当中，要把域名换成自己部署的域名。

## 第3方需要提供的接口

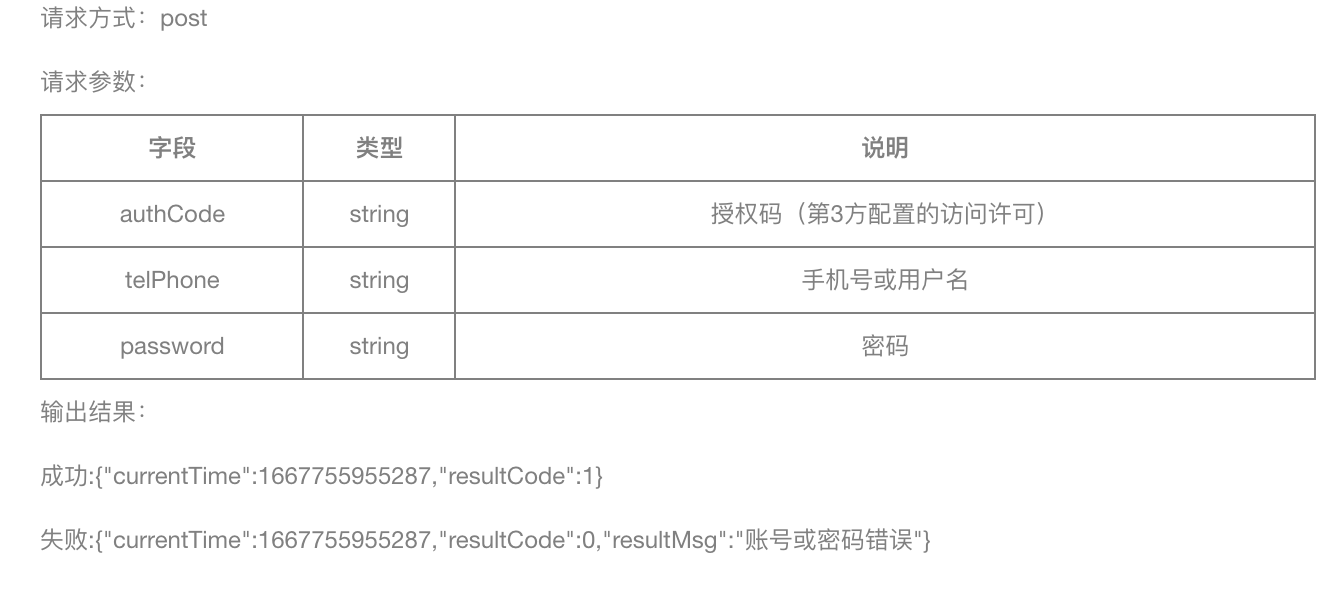
具体参数可参考管理后台【系统管理】【第3方App管理】【添加App】页面说明。

返回结果需要严格按每个接口的【输出结果】，否则无法解析。

resultCode--成功为1 失败为非1.失败必须返回resultMsg字段，且不能为空.

resultMsg--错误提示信息

### 账号密码查询



### 充值回调



### 提现回调



### 余额查询



### 发送通知（可选）



注意：实际当中，要把域名换成自己部署的域名。

免责声明

本系统主要用于大家对数字货币的研究和学习，各位看官请勿用于非法用途。如有违规，本作者概不负责。