[1引言 2](#_Toc518043716)

[1.1编写目的 2](#_Toc518043717)

[1.2前景 2](#_Toc518043718)

[1.3定义 2](#_Toc518043719)

[1.4参考资料 2](#_Toc518043720)

[2软件征述 2](#_Toc518043721)

[2.1软件的结构 2](#_Toc518043722)

[2.2程序表 4](#_Toc518043723)

[2.3文卷表 4](#_Toc518043724)

[3安装与初始化 5](#_Toc518043725)

[4运行说明 6](#_Toc518043726)

[4.1运行表 6](#_Toc518043727)

[4.2运行步骤 7](#_Toc518043728)

[4.3启动区块链网络说明 7](#_Toc518043729)

[4.3.1运行控制 7](#_Toc518043730)

[4.3.2操作信息 7](#_Toc518043731)

[4.3.3输入一输出文卷 11](#_Toc518043732)

[4.3.4输出文段 11](#_Toc518043733)

[4.3.5输出文段的复制 12](#_Toc518043734)

[4.3.6恢复过程 13](#_Toc518043735)

[4.4启动交互API说明 13](#_Toc518043736)

[4.4.1运行控制 13](#_Toc518043737)

[4.4.2操作信息 13](#_Toc518043738)

[4.4.3输入一输出文卷 15](#_Toc518043739)

[4.4.4输出文段 15](#_Toc518043740)

[4.4.5输出文段的复制 16](#_Toc518043741)

[4.4.6恢复过程 16](#_Toc518043742)

[4.5运行WEB应用说明 16](#_Toc518043743)

[4.5.1运行控制 17](#_Toc518043744)

[4.5.2操作信息 17](#_Toc518043745)

[4.5.3输入一输出文卷 22](#_Toc518043746)

[4.5.4输出文段 23](#_Toc518043747)

[4.5.5输出文段的复制 24](#_Toc518043748)

[4.5.6恢复过程 24](#_Toc518043749)

[5非常规过程 24](#_Toc518043750)

[6远程操作 24](#_Toc518043751)

**操作手册**

# 1引言

## 1.1编写目的

本文档用于指导用户对可信数字资产存证应用的部署，配置和使用操作等。

## 1.2前景

软件系统名称：可信数字资产存证应用

任务提出者：新华三技术有限公司

开发者：Genesis-Block 团队

项目来源：中国软件杯参赛题目

## 1.3定义

Fabric：IBM开源的区块链项目，本应用后台区块链采用的架构

SADA：可信数字资产存证应用简称

## 1.4参考资料

《国家标准软件开发文档规范》

《软件开发流程》，清华大学出版社，2005年1月版

# 2软件征述

## 2.1软件的结构

（1）底层区块链网络（这里主要讲链码的功能）

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 功能 | 输入 | 输出 |
| 链码实例化 | 通过调用链码的初始化方法，对链码进实例化 | 排序节点地址  通道名称  链码名称与版本  初始化方法与参数 | 实例化成功或失败的日志信息 |
| 调用链码方法 | 通过调用链码交易方法，可实现对信息的保存，删除，更新以及查询操作 | 通道名称  链码名称与版本  调用的方法与参数 | 保存，删除与更新操作成功或失败将返回日志信息，查询操作成功将返回查询结果，否则返回错误信息 |

（2）交互API

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 功能 | 输入 | 输出 |
| 注册管理员 | 使用区块链管理员身份生成一个Token秘钥，以便后续对区块链操作的使用 | 区块链管理员相关信息 | Token秘钥 |
| 创建通道 | 在区块链网络中创建一个通道 | Token秘钥 | 通道创建成功或失败的日志信息 |
| 安装链码 | 在底层区块链网络中安装一个链码 | 链码名称、路径与版本  Token秘钥 | 链码安装成功或失败的日志信息 |
| 更新链码 | 在底层区块链网络中更新某一个链码 | 新链码名称、路径与版本  Token秘钥 | 链码更新成功或失败的日志信息 |
| 实例化链码 | 通过调用链码的初始化方法，对链码进实例化 | 链码的相关信息  初始化方法与参数  Token秘钥 | 链码实例化成功或失败的日志信息 |
| 调用链码方法 | 通过调用链码交易方法，可实现对信息的保存，删除，更新以及查询操作 | 链码的相关信息  调用的方法与参数  Token秘钥 | 调用保存，删除与更新成功或失败的日志信息。查询成功将返回查询结果，否则返回错误信息 |

（2）WEB 应用（主要讲资产上传与鉴权）

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 功能 | 输入 | 输出 |
| 上传资产 | 上传一个资产文件，首先通过图片识别技术提取出关键信息，然后将关键信息保存到数据库与区块链 | 资产文件  资产名称  资产类型  资产标签 | 资产上传成功后的提示信息 |
| 查看资产列表 | 查询出所有的个人资产列表 | 用户ID | 用户所有的资产信息列表 |
| 查看资产详情 | 查询出资产的详细信息 | 资产ID | 某一个资产的详细信息 |
| 查询资产 | 可以通过资产名称，资产类型，资产标签对资产进行查询 | 资产名称  资产类型  资产标签 | 符合条件的资产列表 |
| 资产鉴权 | 上传鉴权资产的文件，提取关键信息与后台区块链的保存的信息进行比对 | 鉴权资产文件 | 关键信息比对结果  资产MD5值比对结果 |
| 加入组织 | 普通用户填写组织邀请码加入一个组织 | 组织邀请码 | 加入成功或失败的提示信息 |
| 查看组织内所有用户资产 | 列出组织内的所有用户以及用户的资产 | 组织ID  组织管理员ID | 组织内用户的列表信息以及资产信息 |

## 2.2程序表

见详细设计说明书中的编号

## 2.3文卷表

（1）区块链网络配置文件（2）交互API程序目录 （3）WEB系统的目录结构



# 3安装与初始化

**底层区块链网络的部署与启动**

1. 准备1台系统版本为ubuntu16.04的Linux主机，
2. 安装1.7及以上版本的GO语言环境
3. 安装最新稳定版本的Docker
4. 安装Docker-Compose 1.12及以上版本
5. 下载Fabric 源码并切换到1.0分支
6. 下载1.0版本对应的Docker 镜像
7. 启动Fabric网络并运行e2e\_cli项目来测试单机是否能成功部署
8. 根据该主机复制出另外4台一模一样的主机
9. 将多机部署的配置文件依次上传到每一个主机中
10. 依次启动Orderer节点主机，Peer1~Peer2节点主机
11. 进入Peer1节点的Cli容器，创建通道，加入通道并更新锚节点

**交互API的部署与启动**

1. 准备一台拥有JDK1.8 和Maven3.5.3 环境的Win 或者 Linux主机
2. 修改交互程序中配置的区块链网络节点IP
3. 通过Eclipse IDE以Spring Boot的方式启动程序；
4. 或者通过命令行窗口进入交互程序目录
5. 先执行 mvn clean install命令
6. 再执行mvn spring-boot: run命令
7. 调用安装链码以及实例化链码操作

**WEB 应用的部署与运行**

1. 准备一台具有JDK 1.8，MySQL5.5及以上和Tomcat 8 环境的 Win 或者 Linux主机
2. 新建相应的数据库，将项目的数据库SQL文件导入数据库中
3. 将war包放入Tomcat wabapp目录下，运行Tomcat即可启动项目
4. 如若想通过运行源码的方式运行项目，可以使用Eclipse IDE
5. 运行成功后，在本地可通过http://localhost:8080/Sada/访问项目，在虚拟机或云主机中可以通过http://主机IP:8080/Sada/来访问项目

# 4运行说明

项目的运行需要依次启动底层区块链网络，交互API和WEB应用，三者可以独立运行，启动顺序没有硬性规定，但是底层区块链网络的启动和运行需要按照一定的顺序依次进行。每执行一步都要等前一步操作执行完成后方可执行

## 4.1运行表

（1）区块链网络

|  |  |  |
| --- | --- | --- |
| 运行情况 | 目的 | 执行程序或命令 |
| 启动Orderer节点 | 启动Orderer节点 | ./network up |
| 启动Peer1节点 | 启动Peer1节点 | ./network up cdb |
| 启动Peer2节点 | 启动Peer2节点 | ./network up cdb |
| 启动Peer3节点 | 启动Peer3节点 | ./network up cdb |
| 启动Peer4节点 | 启动Peer4节点 | ./network up cdb |
| 进入Peer1节点Cli容器 | 进入Peer1节点Cli容器 | docker exec –it cli bash |
| 执行脚本 | 创建并加入通道，更新锚节点 | bash scripts/script.sh mychannel |

（2）交互API

|  |  |  |
| --- | --- | --- |
| 运行情况 | 目的 | 执行程序或命令 |
| 启动程序 | 启动交互程序 | mvn spring-boot:run |
| 安装链码 | 安装链码 | 执行链码安装方法 |
| 实例化链码 | 实例化链码 | 执行实例化链码方法 |

（3）WEB 应用

|  |  |  |
| --- | --- | --- |
| 运行情况 | 目的 | 执行程序或命令 |
| 上传资产 | 上传资产到数据库和区块链 | 执行资产上传方法 |
| 资产查看 | 查看资产列表情况 | 执行资产列表查看方法 |
| 资产详情 | 查看资产的详细信息 | 执行返回资产详情方法 |
| 资产查询 | 查询出符合条件的资产 | 执行资产查询方法 |
| 资产鉴权 | 上传鉴权资产并进行比对 | 执行资产鉴权方法 |
| 加入组织 | 普通用户填写邀请码加入组织 | 执行加入组织的方法 |
| 查看组织内用户及资产 | 组织管理者查看组织内用户及资产 | 执行查询用户列表及资产列表的方法 |
| 鉴权组织内用户资产 | 组织管理者对组织内用户资产进行鉴权 | 执行鉴权资产的方法 |

## 4.2运行步骤

（1） 按照顺序启动区块链网络

（2） 按照步骤启动交互API

（3） 启动WEB 应用，运行程序

## 4.3启动区块链网络说明

首页进入Orderer 节点主机，进入 /root/go/src/github.com/hyperledger/sada/ 目录，执行 ./network up 命令来启动Orderer 节点，然后依次进入Peer1~Peer4节点主机的相同目录下，依次执行./network up cdb 命令，且待前一个主机命令执行成功后方可执行后一个命令。

节点启动成功后可以进入Peer1节点的Cli容器，执行srcipt.sh 脚本来完成创建通道、加入通道和更新锚节点的操作。

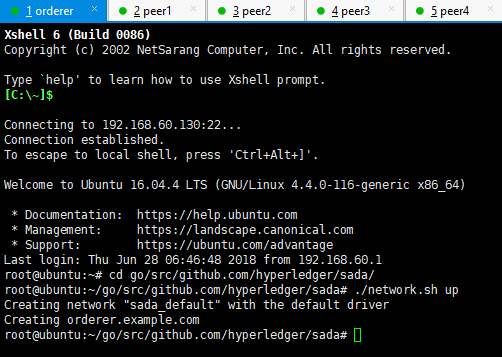
### 4.3.1运行控制

按照下列顺序依次执行，且下一个步骤待上一步骤执行完成之后方可执行

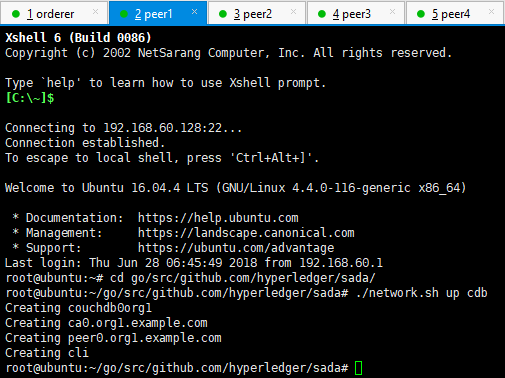
1. 启动Orderer节点
2. 启动Peer1节点
3. 启动Peer2节点
4. 启动Peer3节点
5. 启动Peer4节点
6. 进入Peer1节点Cli容器
7. 执行脚本

### 4.3.2操作信息

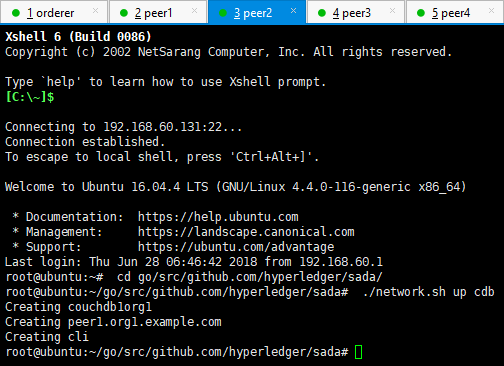
（1）启动Orderer节点



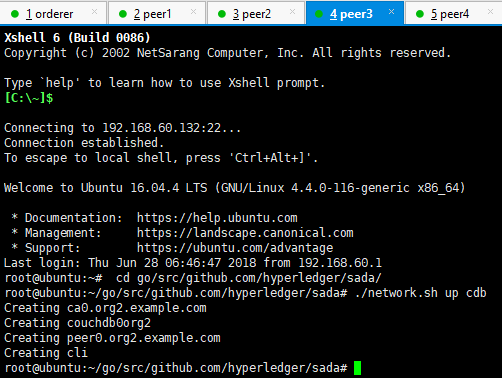
（2）启动Peer1节点



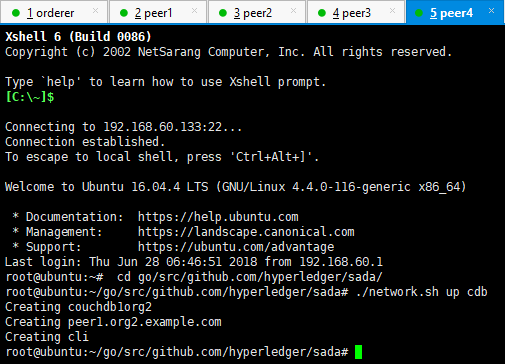
（3）启动Peer2节点



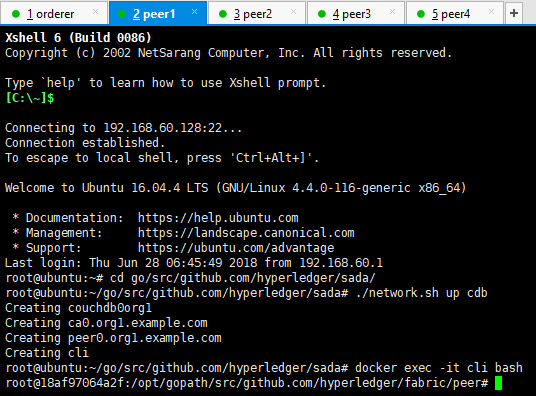
（4）启动Peer3节点



（5）启动Peer4节点



（6）进入Peer1节点Cli容器



（7）执行脚本





### 4.3.3输入一输出文卷

执行目录：$GOPATH/src/github.com/hyperledger/sada/



### 4.3.4输出文段

（1）Orderer节点启动输出

Creating network "sada\_default" with the default driver

Creating orderer.example.com

（2）Peer1节点启动输出

Creating couchdb0org1

Creating ca0.org1.example.com

Creating peer0.org1.example.com

Creating cli

（3）Peer2节点启动输出

Creating couchdb1org1

Creating peer1.org1.example.com

Creating cli

（4）Peer3节点启动输出

Creating ca0.org2.example.com

Creating couchdb0org2

Creating peer0.org2.example.com

Creating cli

（5）Peer4节点启动输出

Creating couchdb1org2

Creating peer1.org2.example.com

Creating cli

（6）进入Peer1节点Cli容器启动输出

无

（7）执行脚本启动输出（截取部分）

\_\_\_\_ \_\_\_\_\_ \_ \_\_\_\_ \_\_\_\_\_ \_\_\_\_ \_ \_\_\_\_ \_

/ \_\_\_| |\_ \_| / \ | \_ \ |\_ \_| / \_\_\_| / \ | \_ \ / \

\\_\_\_ \ | | / \_ \ | |\_) | | | \_\_\_\_\_ \\_\_\_ \ / \_ \ | | \ | / \_ \

\_\_\_) | | | / \_\_\_ \ | \_ < | | |\_\_\_\_\_| \_\_\_) | / \_\_\_ \ | |\_/ | / \_\_\_ \

|\_\_\_\_/ |\_| /\_/ \\_\ |\_| \\_\ |\_| |\_\_\_\_/ /\_/ \\_\ |\_\_\_\_/ /\_/ \\_\

Channel name : mychannel

Creating channel...

CORE\_PEER\_TLS\_ROOTCERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt

CORE\_PEER\_TLS\_KEY\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/server.key

CORE\_PEER\_LOCALMSPID=Org1MSP

CORE\_VM\_ENDPOINT=unix:///host/var/run/docker.sock

================All GOOD End-2-End execution completed================

\_\_\_\_\_ \_ \_ \_\_\_\_ \_\_\_\_ \_ \_\_\_\_ \_

| \_\_\_\_| | \ | | | \_ \ / \_\_\_| / \ | \_ \ / \

| \_| | \| | | | | | \_\_\_\_\_ \\_\_\_ \ / \_ \ | | \ | / \_ \

| |\_\_\_ | |\ | | |\_| | |\_\_\_\_\_| \_\_\_) | / \_\_\_ \ | |\_/ | / \_\_\_ \

|\_\_\_\_\_| |\_| \\_| |\_\_\_\_/ |\_\_\_\_/ /\_/ \\_\ |\_\_\_\_/ /\_/ \\_\

### 4.3.5输出文段的复制

无

### 4.3.6恢复过程

（1）执行 ./network down 命令

（2）执行 docker rm -rf $(docker ps -aq)

（3）执行 docker network prune -f

（4）执行docker volume prune –f

（5）重新按照上述步骤运行

## 4.4启动交互API说明

以命令行方式启动为例，首先通过Spring-Boot的方式启动交互程序，执行：mvn clean install 命令后执行 mvn spring-boot:run，项目启动成功后通过在命令行下执行curl 命令来完成链码的安装与实例化操作

### 4.4.1运行控制

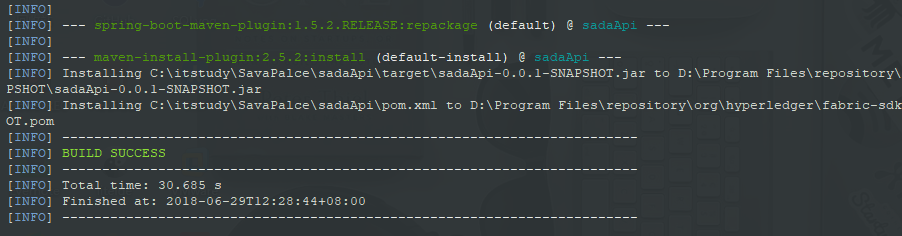
按照下列顺序依次执行，且下一个步骤待上一步骤执行完成之后方可执行

1. 执行mvn clean install
2. 执行mvn spring-boot:run
3. 注册管理员得到Token秘钥
4. 执行安装链码操作
5. 执行实例化链码操作

### 4.4.2操作信息

（1）执行mvn clean install

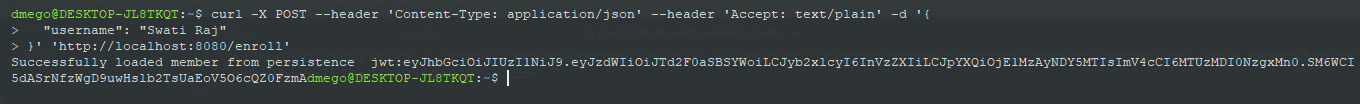




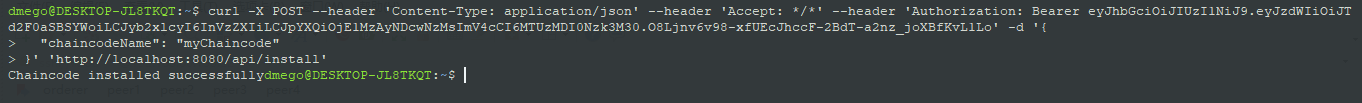
（2）执行mvn spring-boot:run



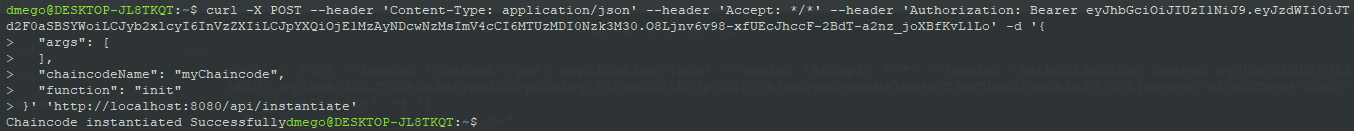
（3）注册管理员得到Token秘钥



（4）执行安装链码操作

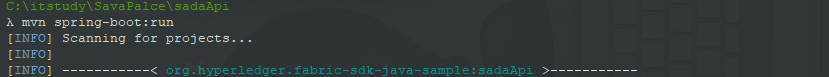


（4）执行实例化链码操作



### 4.4.3输入一输出文卷

执行目录：交互程序的根目录(sadaApi)



### 4.4.4输出文段

1. 执行mvn clean install（截取部分）

[INFO] Scanning for projects...

[INFO]

[INFO] -----------< org.hyperledger.fabric-sdk-java-sample:sadaApi >-----------

[INFO] Building sadaApi 0.0.1-SNAPSHOT

[INFO] --------------------------------[ jar ]---------------------------------

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 30.685 s

[INFO] Finished at: 2018-06-29T12:28:44+08:00

[INFO] ------------------------------------------------------------------------

1. 执行mvn spring-boot:run（截取部分）

[INFO] Scanning for projects...

[INFO]

[INFO] -----------< org.hyperledger.fabric-sdk-java-sample:sadaApi >-----------

[INFO] Building sadaApi 0.0.1-SNAPSHOT

[INFO] --------------------------------[ jar ]---------------------------------

[INFO]

[INFO] >>> spring-boot-maven-plugin:1.5.2.RELEASE:run (default-cli) > test-compile @ sadaApi >>>

[INFO] --- spring-boot-maven-plugin:1.5.2.RELEASE:run (default-cli) @ sadaApi ---

. \_\_\_\_ \_ \_\_ \_ \_

/\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \

( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \

\\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )

' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /

=========|\_|==============|\_\_\_/=/\_/\_/\_/

:: Spring Boot :: (v1.5.2.RELEASE)

1. 注册管理员得到Token秘钥

Successfully loade dmember from persistence jwt:

eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJTd2F0aSBSYWoiLCJyb2xlcyI6InVzZXIiLCJpYXQiOjE1MzAyNDY5MTIsImV4cCI6MTUzMDI0NzgxMn0.SM6WCI5dASrNfzWgD9uwHslb2TsUaEoV5O6cQZ0FzmA

1. 执行安装链码操作

Chaincode installed successfully

1. 执行实例化链码操作

Chaincode instantiated Successfully

### 4.4.5输出文段的复制

无

### 4.4.6恢复过程

（1） 强行停止程序运行

（2） 更改程序的配置文件

（3） 将链码名称或版本进行更改

（4） 按照上述步骤先启动程序

（5） 再执行更新链码的方法

（6） 如果运行出错，执行区块链的恢复过程后在执行这一步操作

## 4.5运行WEB应用说明

在运行WEB应用前，应先执行数据库SQL文件将数据与表导入到主机数据库中，运行WEB应用有两种方式，一种是通过war 包方式启动，另一种是通过Eclipse 运行源码的方式启动程序。第一种是先将war包放置在Tomcat 的 webapps目录下，启动tomcat后将自动解包并运行项目。启动成功后，在浏览器中输入http://主机IP:/Sada即可进入系统，进入后可以进行上传资产，资产鉴权等一些列操作。第二种方式启动比较简单，启动成功后按照统一的操作即可运行应用

### 4.5.1运行控制

主要以资产的上传与鉴权流程为例，下列步骤一般没有先后顺序，可以依次执行也可以独立执行来验证某项功能

1. 注册登录系统
2. 上传资产
3. 资产查询
4. 查看资产详细信息
5. 资产鉴权
6. 加入组织
7. 鉴权组织下某一用户的资产
8. 查看组织内资产与鉴权统计详情
9. 查看区块链相关信息

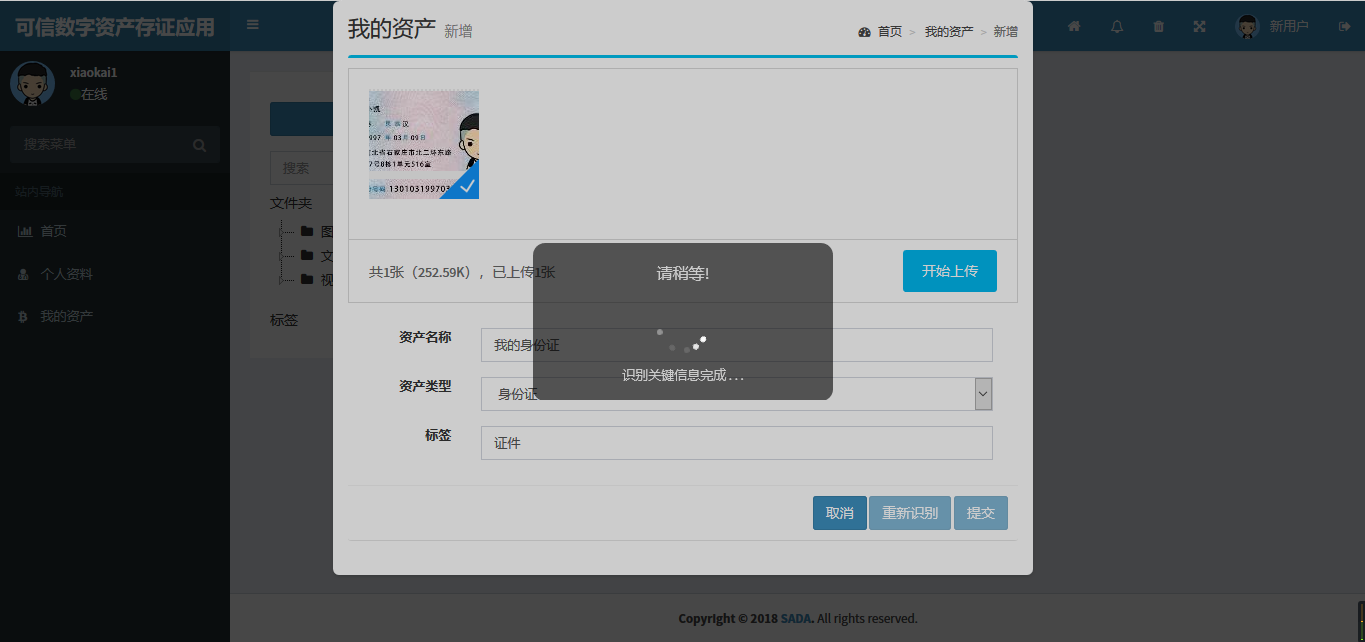
### 4.5.2操作信息

1. 注册登录系统



1. 上传资产

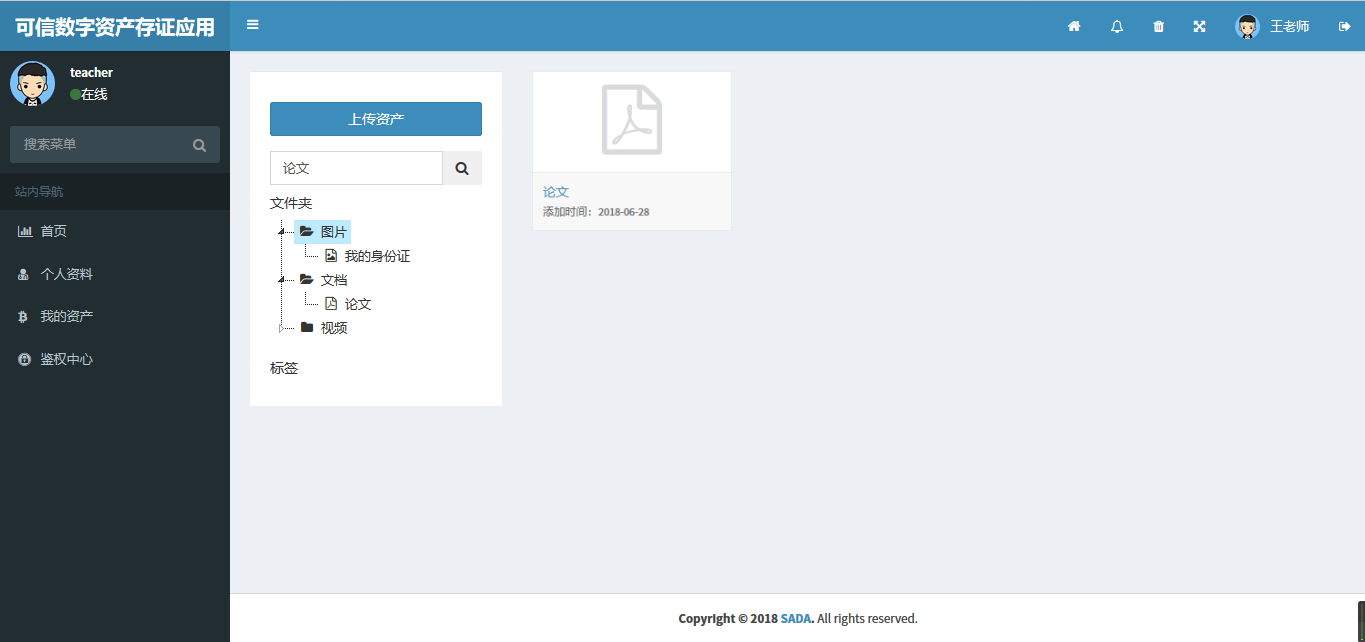
1.识别关键信息



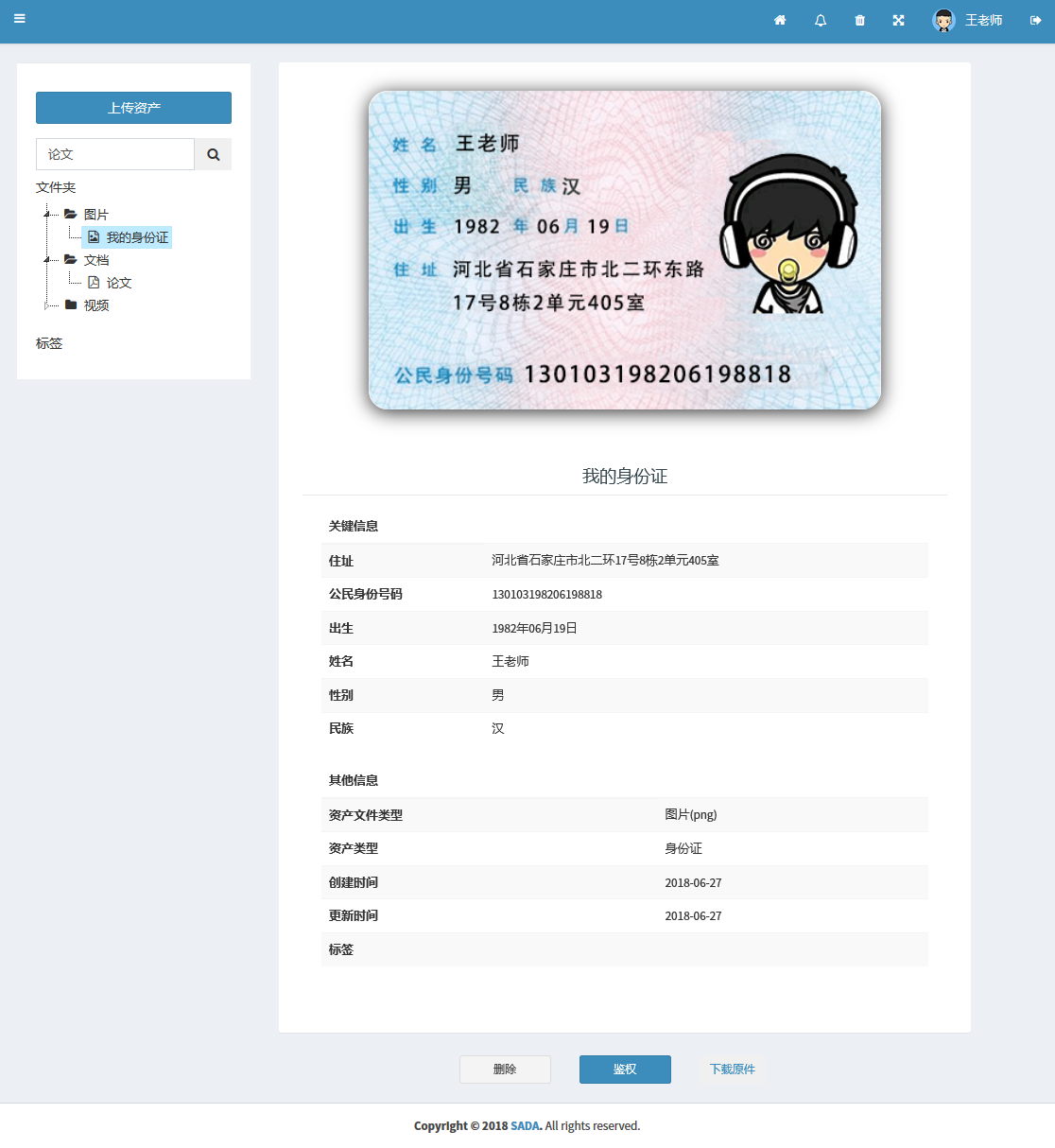
2、提交到数据库与区块链网络



1. 资产查询：输入框搜索资产名称，通过资产类型，以及标签进行检索

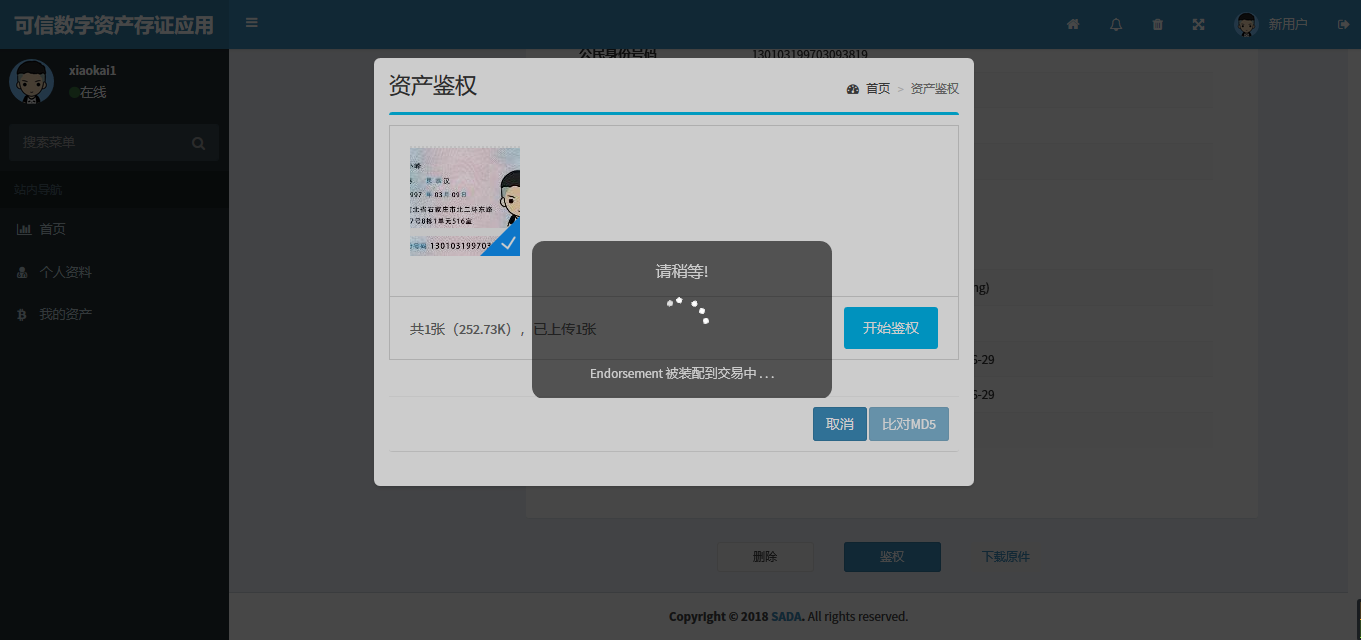


1. 查看资产详细信息

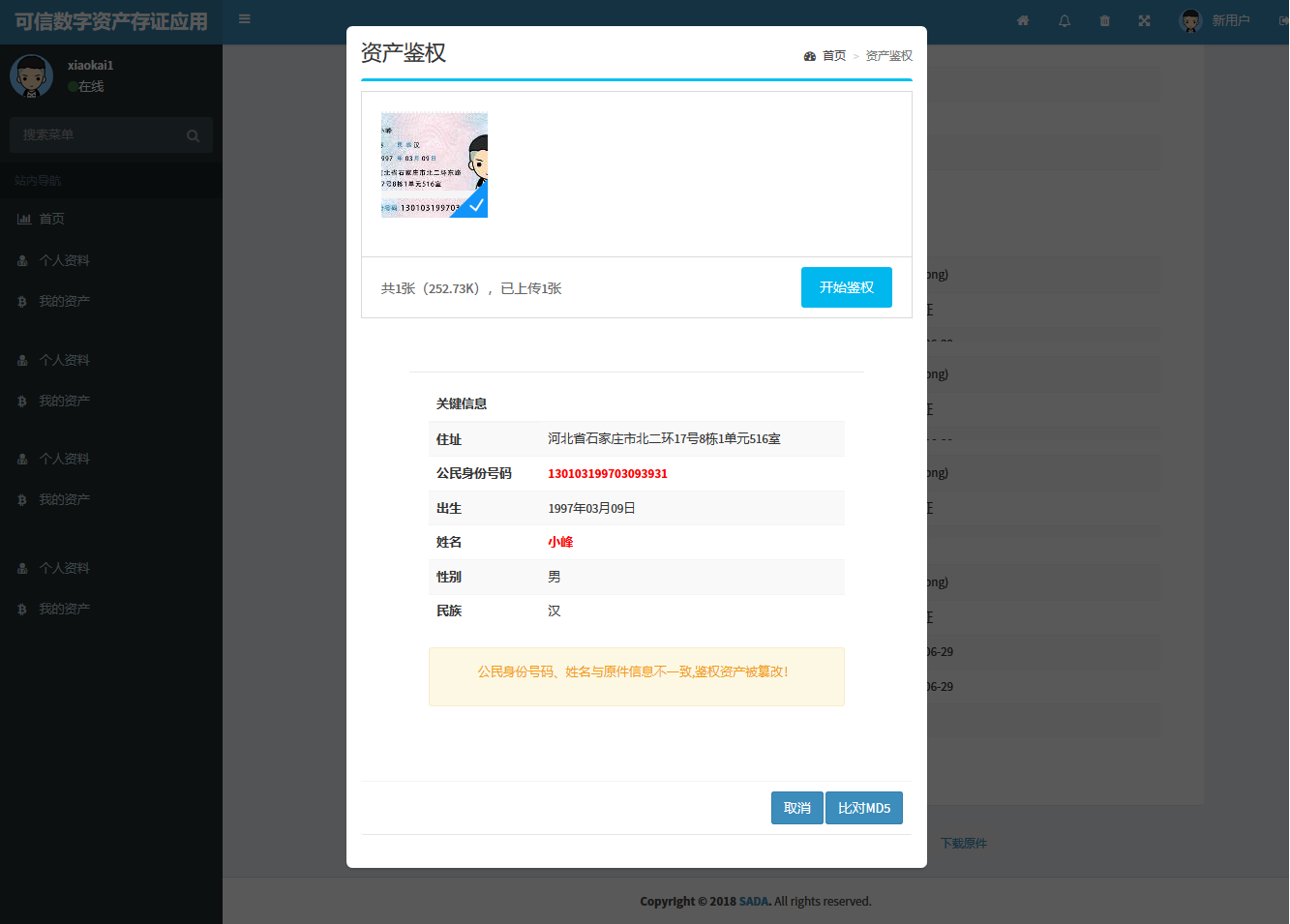


1. 资产鉴权

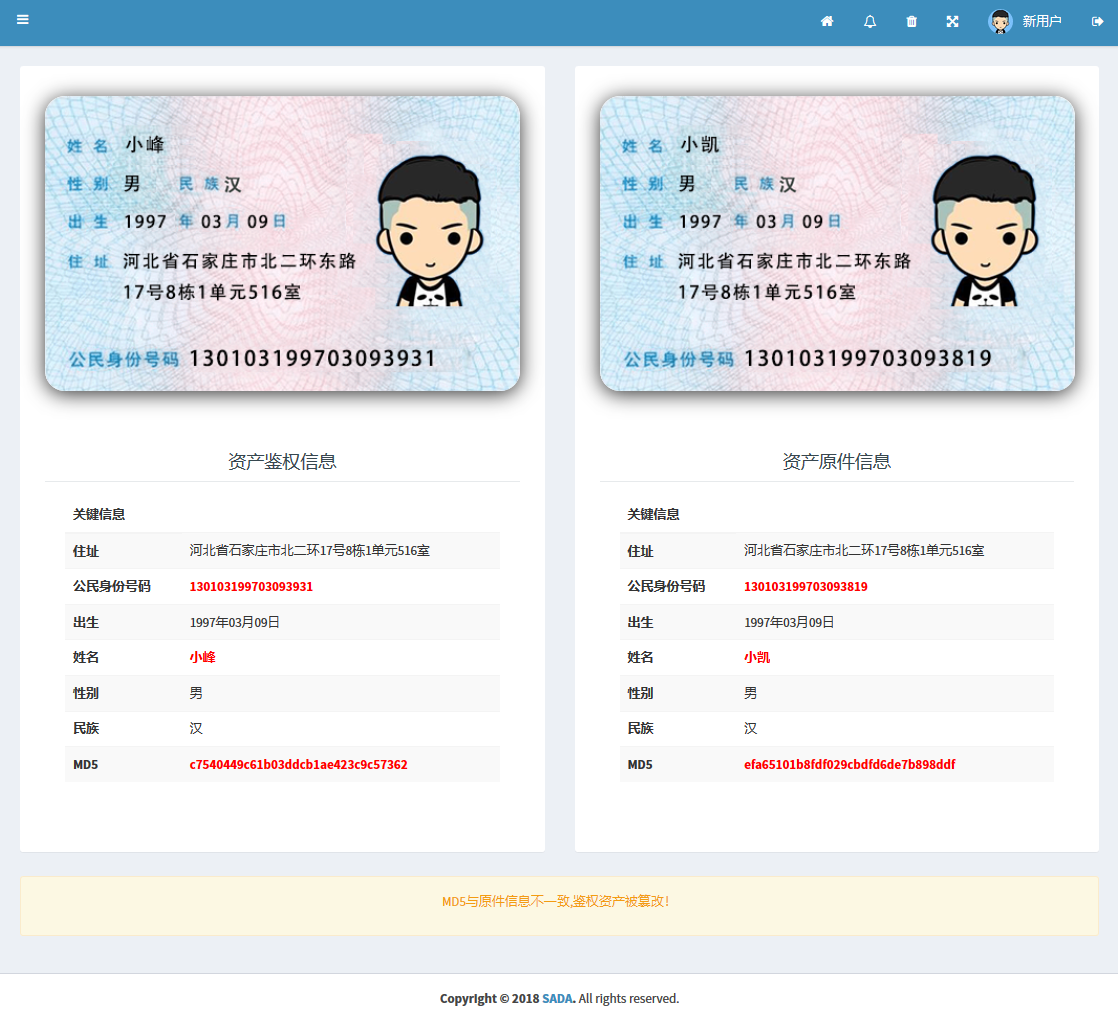
1．上传鉴权资产



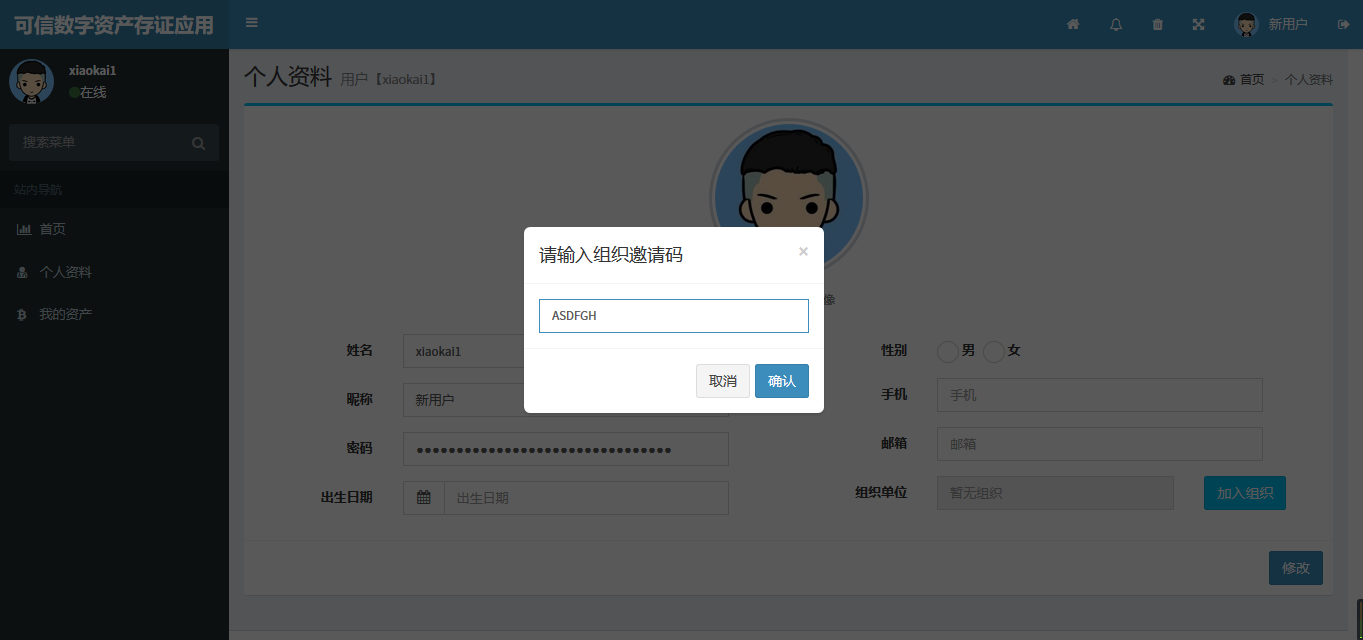
2.显示出鉴权结果



3.继续比对MD5值



1. 加入组织



1. 鉴权组织下某一用户的资产



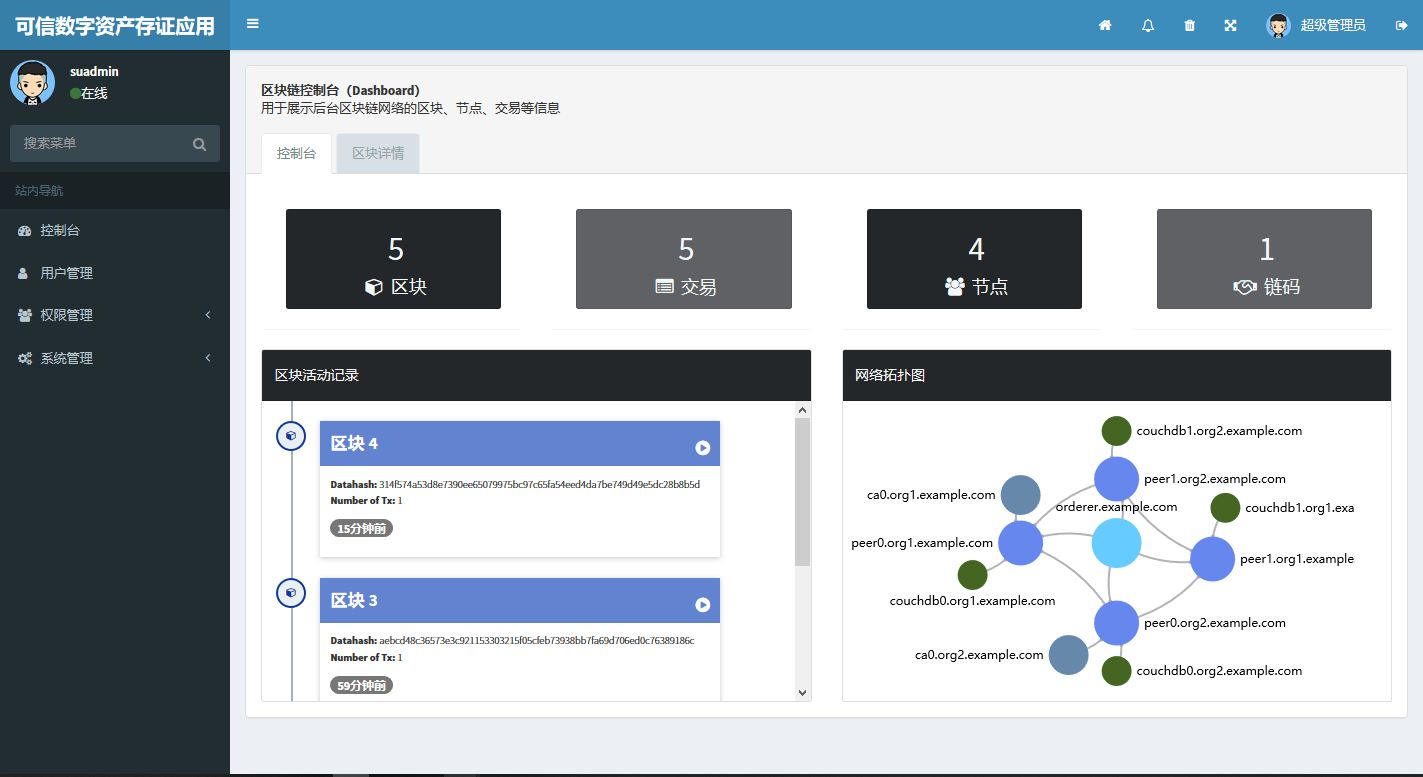
点击鉴权，下面过程与上述鉴权操作相同

1. 查看组织内资产与鉴权统计详情

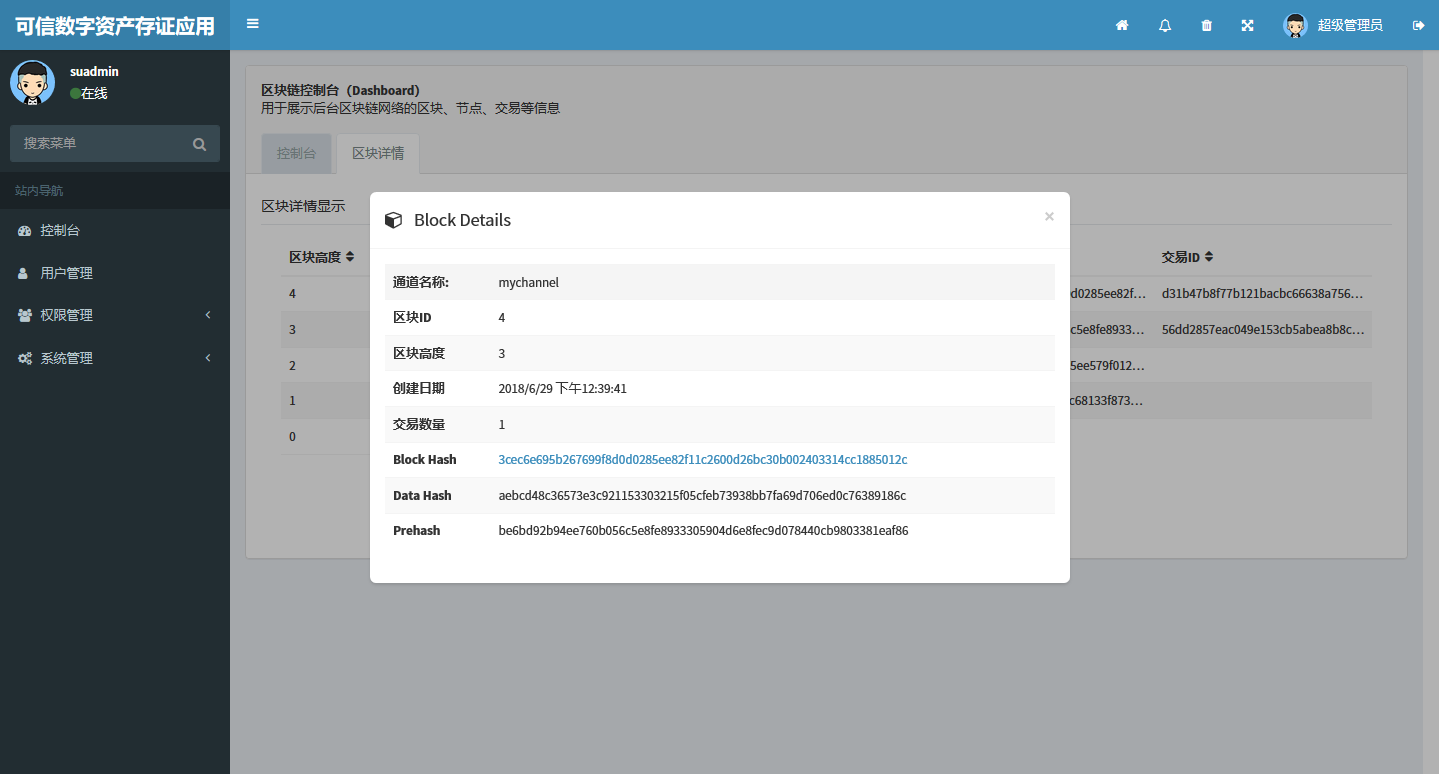


1. 查看区块链相关信息

1.查看控制台



2.查看区块详情



### 4.5.3输入一输出文卷

资产将上传至应用的upload 文件夹下，并按照用户名称，资产类型进行分类

### 4.5.4输出文段

1. 注册登录系统



1. 上传资产



1. 资产查询

显示查询结果

1. 查看资产详细信息

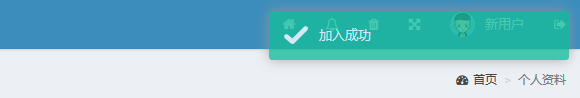
显示资产详细信息

1. 资产鉴权





1. 加入组织



1. 鉴权组织下某一用户的资产

输出信息与上述鉴权操作一样

1. 查看组织内资产与鉴权统计详情

在页面上显示相关信息

1. 查看区块链相关信息

在页面上显示相关信息

### 4.5.5输出文段的复制

无

### 4.5.6恢复过程

（1）关闭Tomcat服务器

（2）在Eclipse环境下调试程序，修复出错的BUG

（3）重新启动Tomcat运行项目

（4）进入系统

# 5非常规过程

根据不同的出错情况给出不同的出错信息，一般用对话框给出。

出错处理对策对一般错误，给用户提示信息，让用户重新输入或退出。

对于严重错误，启动备份文件恢复。并为每个操作人员设置账号及口令，规定每一级操作人员的权限。

# 6远程操作

本项目的底层区块链网络以及交互API程序，WEB项目的启动均可通过远程访问的方式进行启动部署操作，具体的操作方式与上述的启动操作步骤一致。