操作手册

|  |  |
| --- | --- |
| 题目 | 可信数字资产存证应用 |
| 组员 | 曾凯，刘西宁，刘雨鑫 |
| 版本 | V2.0 |

[1引言 3](#_Toc520742916)

[1.1编写目的 3](#_Toc520742917)

[1.2前景 3](#_Toc520742918)

[1.3定义 3](#_Toc520742919)

[1.4参考资料 3](#_Toc520742920)

[2软件征述 3](#_Toc520742921)

[2.1软件的结构 3](#_Toc520742922)

[2.2程序表 5](#_Toc520742923)

[2.3文卷表 5](#_Toc520742924)

[3安装与初始化 6](#_Toc520742925)

[4运行说明 6](#_Toc520742926)

[4.1运行表 6](#_Toc520742927)

[4.2运行步骤 7](#_Toc520742928)

[4.3启动区块链网络说明 7](#_Toc520742929)

[4.3.1运行控制 7](#_Toc520742930)

[4.3.2操作信息 8](#_Toc520742931)

[4.3.3输入一输出文卷 9](#_Toc520742932)

[4.3.4输出文段 9](#_Toc520742933)

[4.3.5输出文段的复制 11](#_Toc520742934)

[4.3.6恢复过程 11](#_Toc520742935)

[4.4启动图像识别程序说明 11](#_Toc520742936)

[4.4.1运行控制 11](#_Toc520742937)

[4.4.2操作信息 11](#_Toc520742938)

[4.4.3输入一输出文卷 11](#_Toc520742939)

[4.4.4输出文段 11](#_Toc520742940)

[4.4.5输出文段的复制 12](#_Toc520742941)

[4.4.6恢复过程 12](#_Toc520742942)

[4.5运行WEB应用说明 12](#_Toc520742943)

[4.5.1运行控制 12](#_Toc520742944)

[4.5.2操作信息 12](#_Toc520742945)

[4.5.3输入一输出文卷 18](#_Toc520742946)

[4.5.4输出文段 19](#_Toc520742947)

[4.5.5输出文段的复制 20](#_Toc520742948)

[4.5.6恢复过程 20](#_Toc520742949)

[5非常规过程 20](#_Toc520742950)

[6远程操作 20](#_Toc520742951)

**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

《操作手册》V1.0

《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

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 功能 | 输入 | 输出 |
| 创建通道 | 创建一个通道 | 无 | 通道创建成功或失败的日志信息 |
| 安装链码 | 在peer节点上安装一个链码 | 链码名称  路径与版本 | 链码安装成功或失败的日志信息 |
| 更新链码 | 在peer节点上更新某一个链码 | 新链码名称  路径与版本 | 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 | 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 | 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 | 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）图像识别程序

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 功能 | 输入 | 输出 |
| 提取身份证关键信息 | 通过openCV对图像进行处理，然后用Tesseract-OCR进行识别 | 身份证图片文件  资产类型参数 | 识别成功或失败的日志信息 |
| 提取学生卡关键信息 | 通过openCV对图像进行处理，然后用Tesseract-OCR进行识别 | 学生卡图片文件  资产类型参数 | 识别成功或失败的日志信息 |

（3）WEB 应用（主要讲资产上传与鉴权）

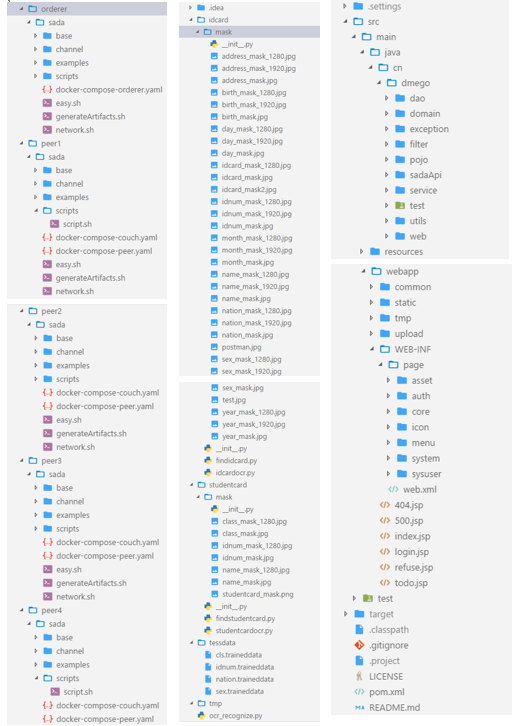
|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 功能 | 输入 | 输出 |
| 上传资产 | 上传一个资产文件，首先通过图片识别技术提取出关键信息，然后将关键信息保存到数据库与区块链 | 资产文件  资产名称  资产类型  资产标签 | 资产上传成功后的提示信息 |
| 查看资产列表 | 查询出所有的个人资产列表 | 用户ID | 用户所有的资产信息列表 |
| 查看资产详情 | 查询出资产的详细信息 | 资产ID | 某一个资产的详细信息 |
| 查询资产 | 可以通过资产名称，资产类型，资产标签对资产进行查询 | 资产名称  资产类型  资产标签 | 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  资产原件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. 进入Orderer节点主机，执行自动化启动网络脚本

**图像识别程序部署与启动**

1. 准备一台拥有Python 和openCV 环境的Win 或者 Linux主机
2. 通过pip安装numpy，Pillow，pytesseract等依赖
3. 将模型数据拷贝到tesseract-ocr的tessdata文件夹下
4. 运行ocr\_ recognize.py文件，开启http服务

**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/newsada/访问项目，在虚拟机或云主机中可以通过http://主机IP:8080/ newsada /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，图像识别程序和WEB应用，三者可以独立运行，启动顺序没有硬性规定。

## 4.1运行表

（1）区块链网络

|  |  |  |
| --- | --- | --- |
| 运行情况 | 目的 | 执行程序或命令 |
| 执行自动化部署脚本 | 启动节点，执行创建，加入通道，安装并实例化链码的脚本 | ./autostart.sh |

（2）图像识别程序

|  |  |  |
| --- | --- | --- |
| 运行情况 | 目的 | 执行程序或命令 |
| 启动程序 | 开启http服务，识别远程图片 | ./start.sh |

（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） 按照步骤启动图像识别程序

（3） 启动WEB 应用，运行程序

## 4.3启动区块链网络说明

首先进入Orderer 节点主机，进入 /root/go/src/github.com/hyperledger/sada/ 目录，执行 ./autostart.sh 脚本来依次启动Orderer 节点，Peer1~Peer4节点，启动成功后自动进入Peer4节点的Cli容器，执行srcipt.sh 脚本来完成创建通道、加入通道、更新锚节点、安装并实例化链码的操作。

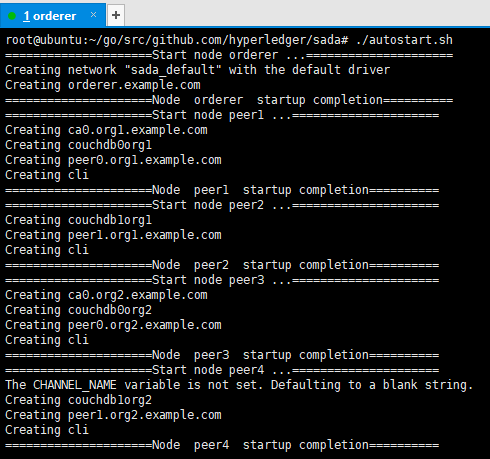
### 4.3.1运行控制

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. 进入Orderer节点主机，执行./autostart.sh脚本
2. 自动进入Peer4节点Cli容器，执行脚本

### 4.3.2操作信息

（1）进入Orderer节点主机，执行./autostart.sh脚本



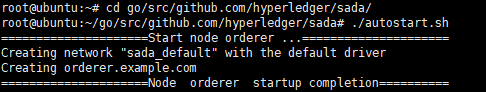
（2）进入Peer4节点Cli容器，执行脚本





### 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节点执行自动化脚本输出

=====================Start node orderer ...=====================

Creating network "sada\_default" with the default driver

Creating orderer.example.com

=====================Node orderer startup completion==========

=====================Start node peer1 ...=====================

Creating ca0.org1.example.com

Creating couchdb0org1

Creating peer0.org1.example.com

Creating cli

=====================Node peer1 startup completion==========

=====================Start node peer2 ...=====================

Creating couchdb1org1

Creating peer1.org1.example.com

Creating cli

=====================Node peer2 startup completion==========

=====================Start node peer3 ...=====================

Creating ca0.org2.example.com

Creating couchdb0org2

Creating peer0.org2.example.com

Creating cli

=====================Node peer3 startup completion==========

=====================Start node peer4 ...=====================

The CHANNEL\_NAME variable is not set. Defaulting to a blank string.

Creating couchdb1org2

Creating peer1.org2.example.com

Creating cli

=====================Node peer4 startup completion==========

（2）自动进入Peer4节点Cli容器，执行脚本启动输出（截取部分）

Installing chaincode on org1/peer1...

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/org2.example.com/peers/preer1.org2.example.com/tls/server.key

CORE\_PEER\_LOCALMSPID=Org1MSP

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

CORE\_PEER\_TLS\_CERT\_FILE=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org2.example.com/peers/preer1.org2.example.com/tls/server.crt

CORE\_PEER\_TLS\_ENABLED=true

CORE\_PEER\_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp

CORE\_PEER\_ID=cli

CORE\_LOGGING\_LEVEL=DEBUG

CORE\_PEER\_ADDRESS=peer1.org1.example.com:7051

2018-07-30 10:48:18.465 UTC [golang-platform] func1 -> DEBU 009 Discarding provided package github.com/hyperledger/fabric/protos/peer

2018-07-30 10:48:18.465 UTC [golang-platform] GetDeploymentPayload -> DEBU 00a done

2018-07-30 10:48:18.466 UTC [msp/identity] Sign -> DEBU 00b Sign: plaintext: 0A8A070A5C08031A0C08F2D9FBDA0510...CBF19F000000FFFF68F9B4B5001C0000

2018-07-30 10:48:18.466 UTC [msp/identity] Sign -> DEBU 00c Sign: digest: 9B589255D80EDCEE4E1667C8384CC714736EB0163933F49942235AD42E4C186B

2018-07-30 10:48:18.473 UTC [chaincodeCmd] install -> DEBU 00d Installed remotely response:<status:200 payload:"OK" >

2018-07-30 10:48:18.473 UTC [main] main -> INFO 00e Exiting.....

===================== Chaincode is installed on remote peer PEER1 =====================

### 4.3.5输出文段的复制

无

### 4.3.6恢复过程

（1）执行 ./autostop.sh 脚本

（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启动图像识别程序说明

运行shell脚本，开启http服务，识别远程图片，并让python程序在后台运行

### 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. 进入程序主目录，执行start.sh脚本

### 4.4.2操作信息

（1）进入程序主目录，执行start.sh脚本



### 4.4.3输入一输出文卷

执行目录：图像识别程序的根目录(idcardoc)

### 4.4.4输出文段

无

### 4.4.5输出文段的复制

无

### 4.4.6恢复过程

（1） 查看系统端口占用情况

（2） 杀死8080端口

（3） 按照上述步骤先启动程序

## 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:/newsada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

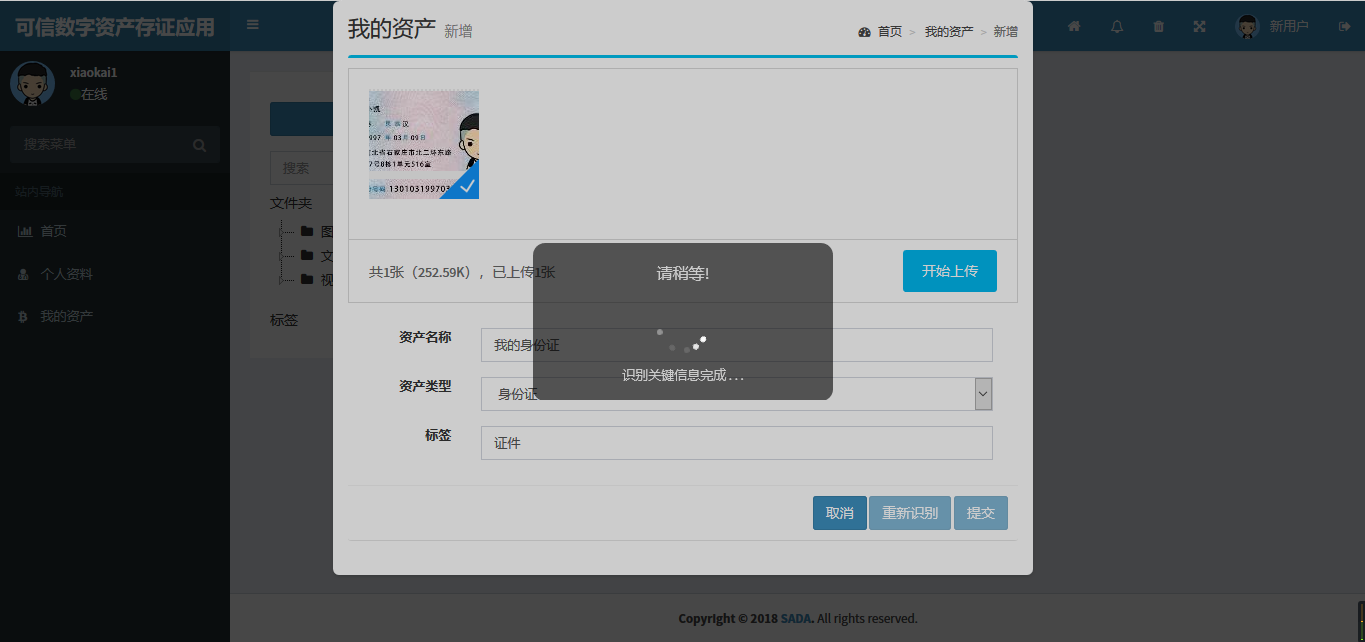
### 4.5.2操作信息

1. 注册登录系统



1. 上传资产

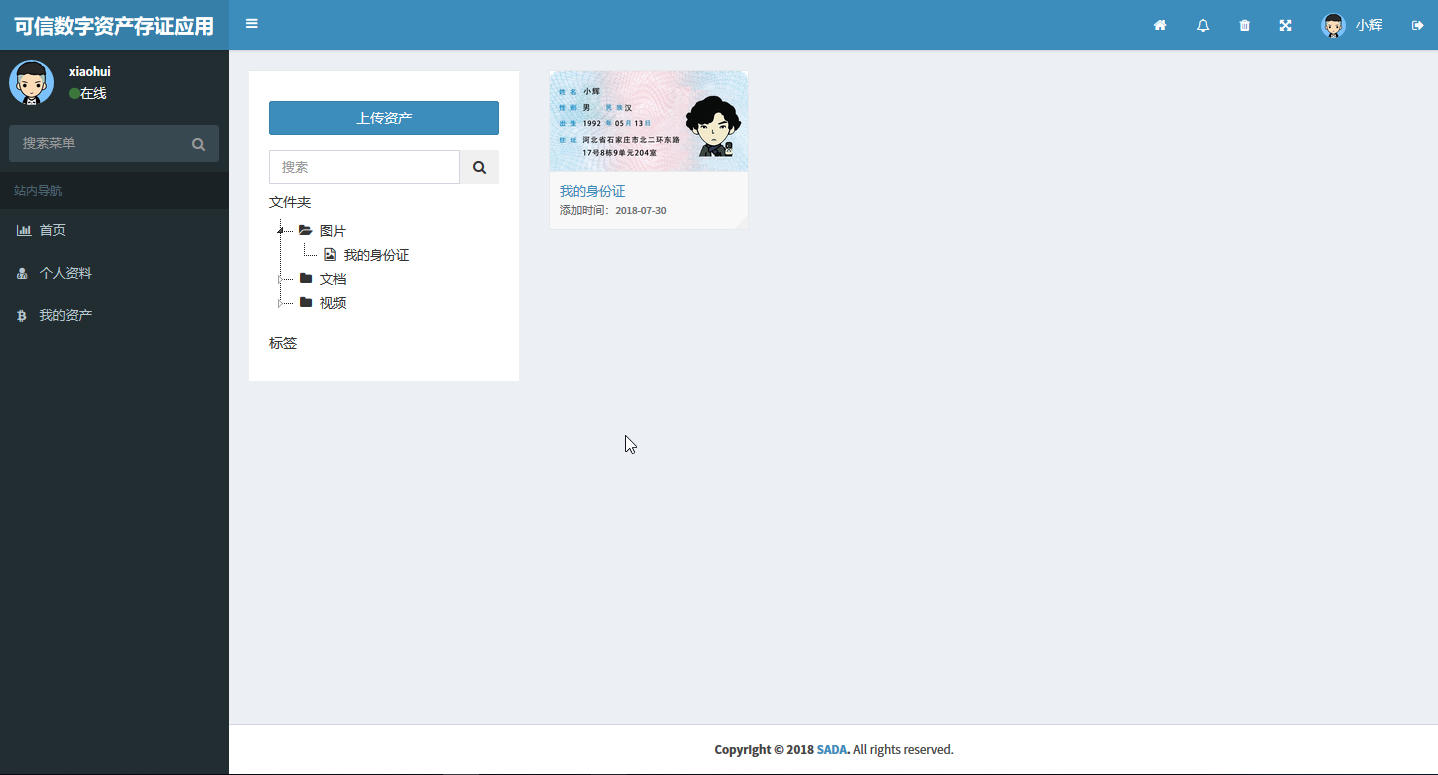
1.识别关键信息



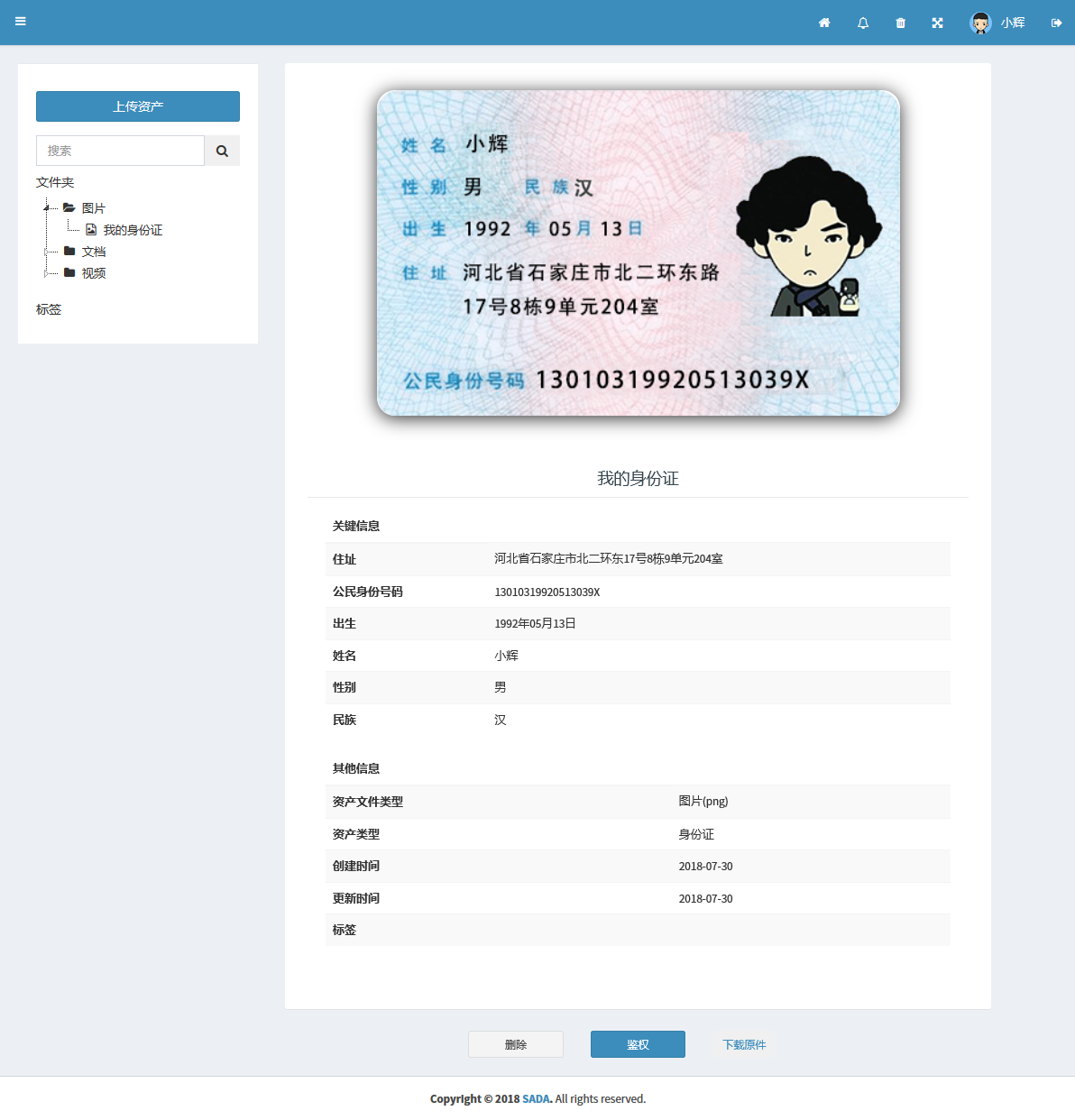
2、提交到数据库与区块链网络



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

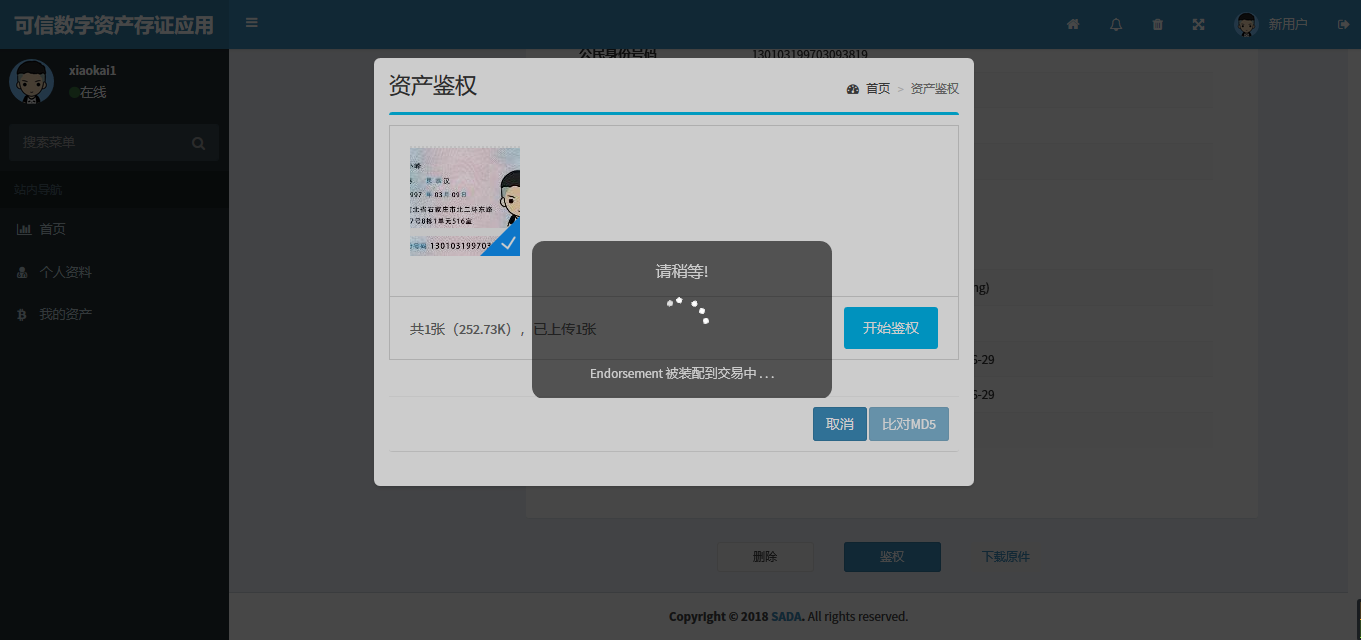


1. 查看资产详细信息

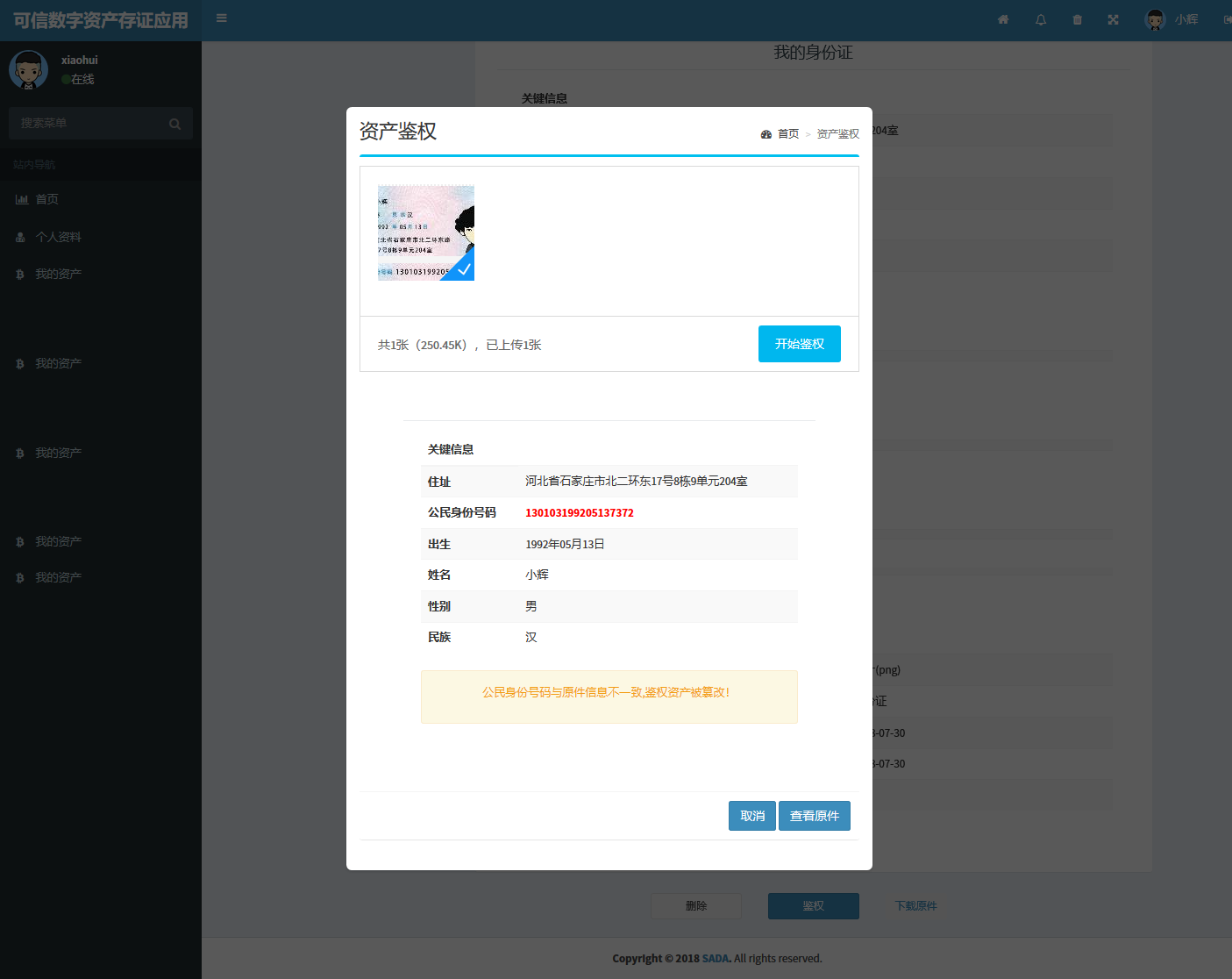


1. 资产鉴权

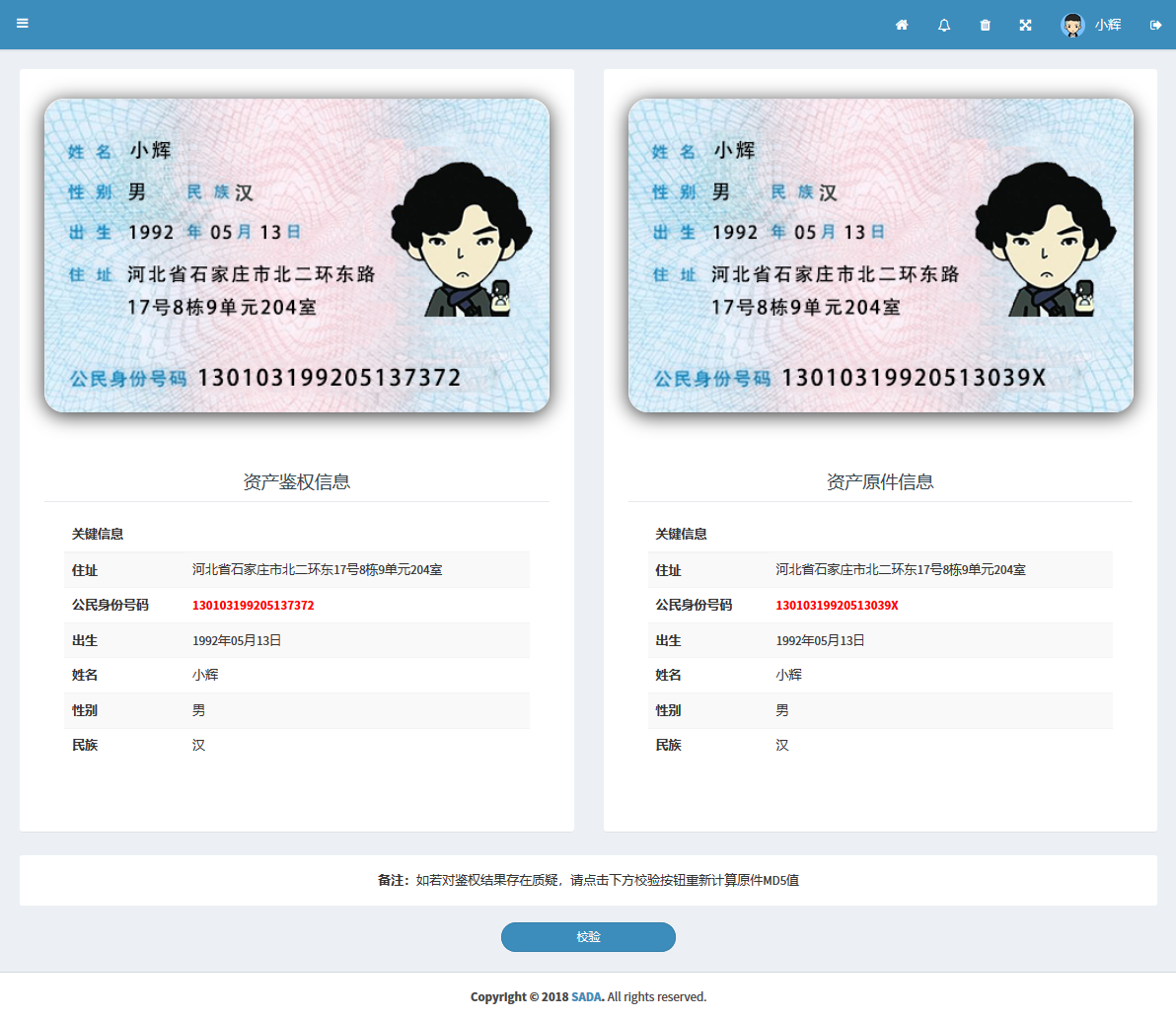
1．上传鉴权资产



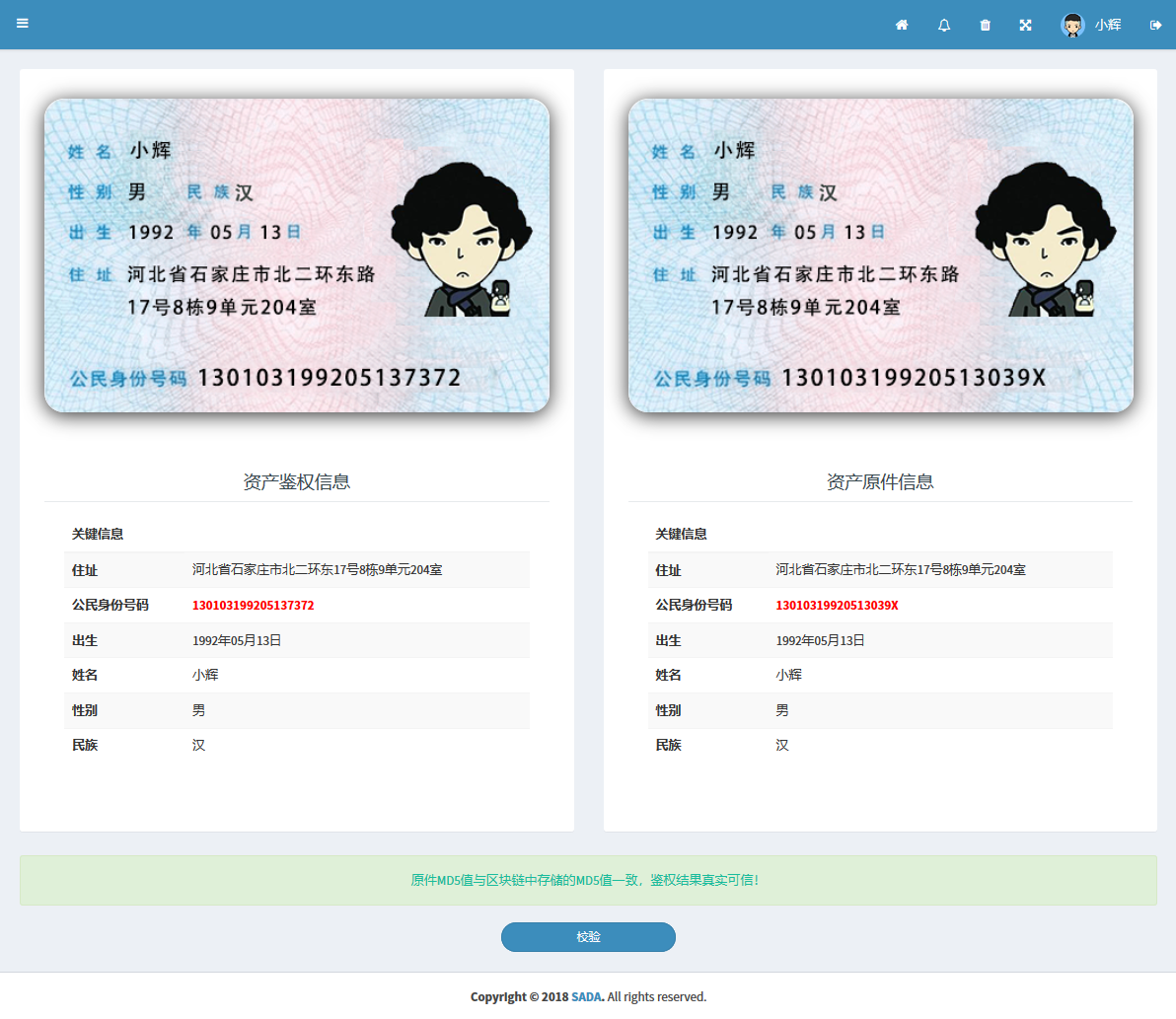
2.显示出鉴权结果



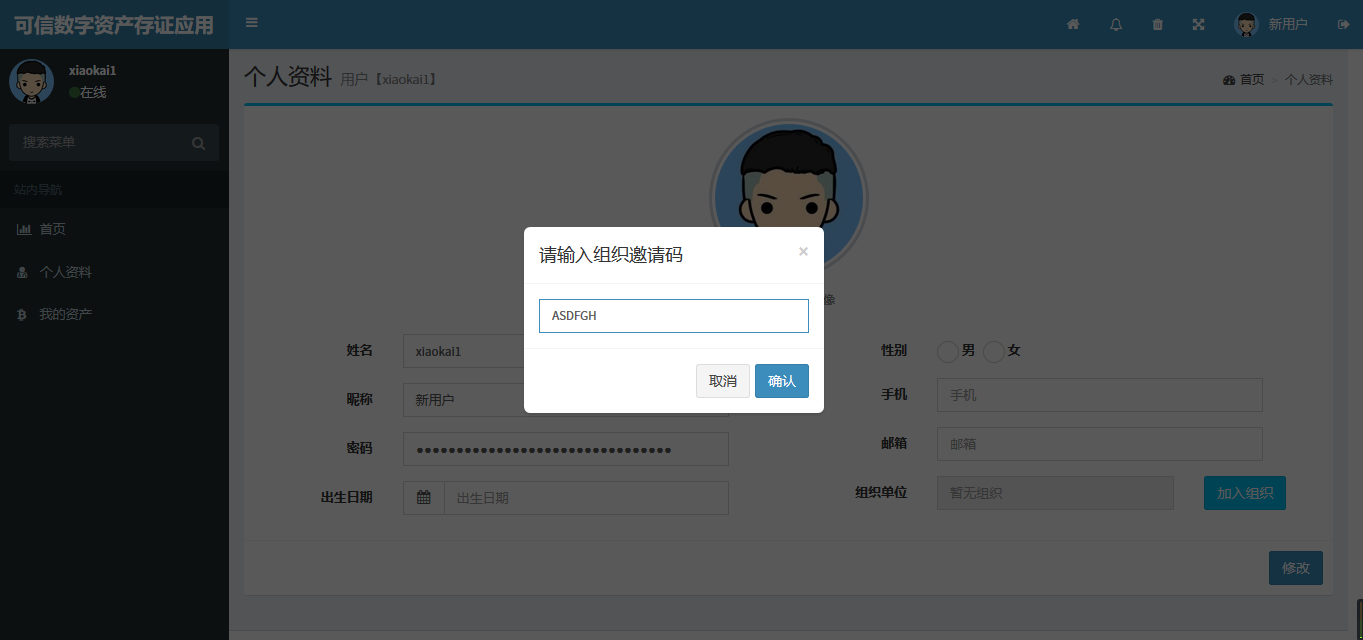
3.查看原件



4.校验原件MD5值



1. 加入组织



1. 鉴权组织下某一用户的资产



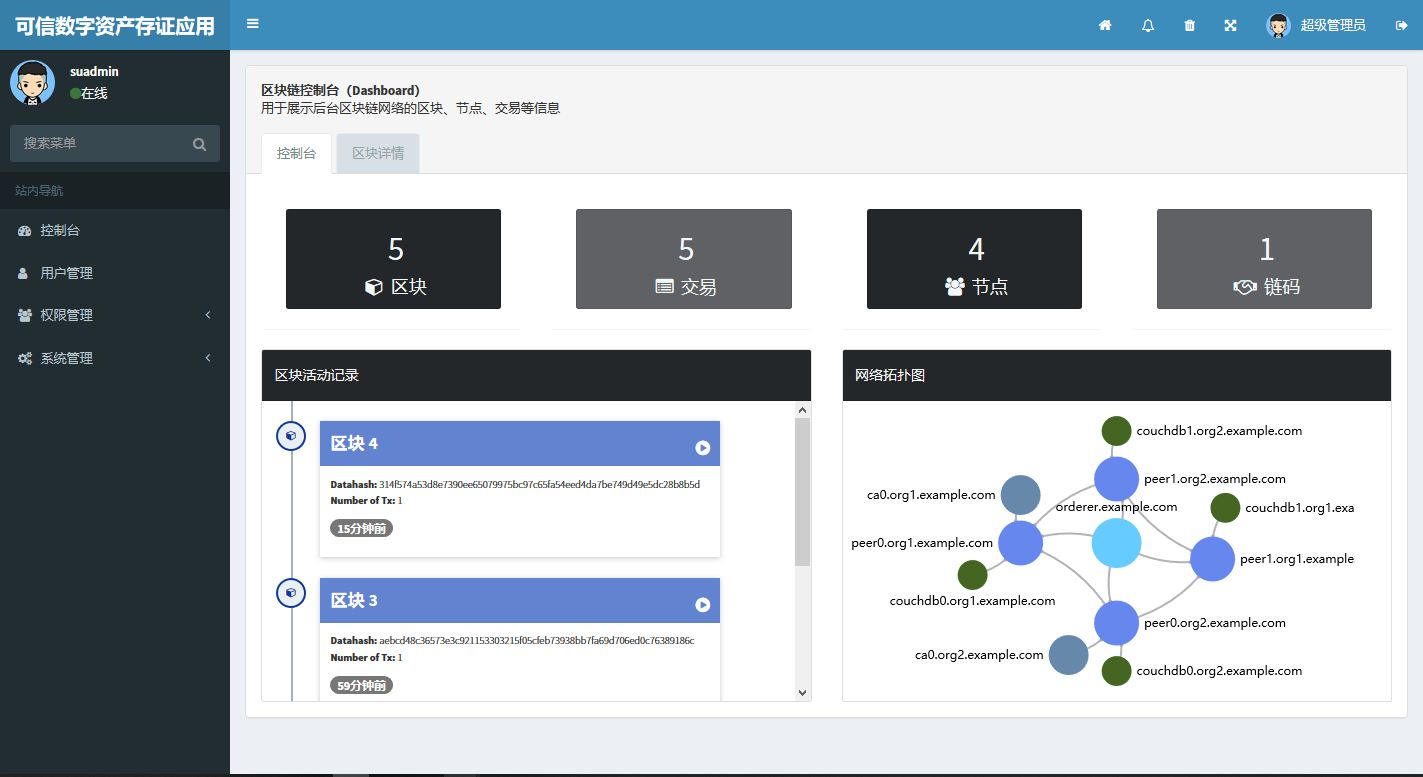
点击鉴权，下面过程与上述鉴权操作相同

1. 查看组织内资产与鉴权统计详情

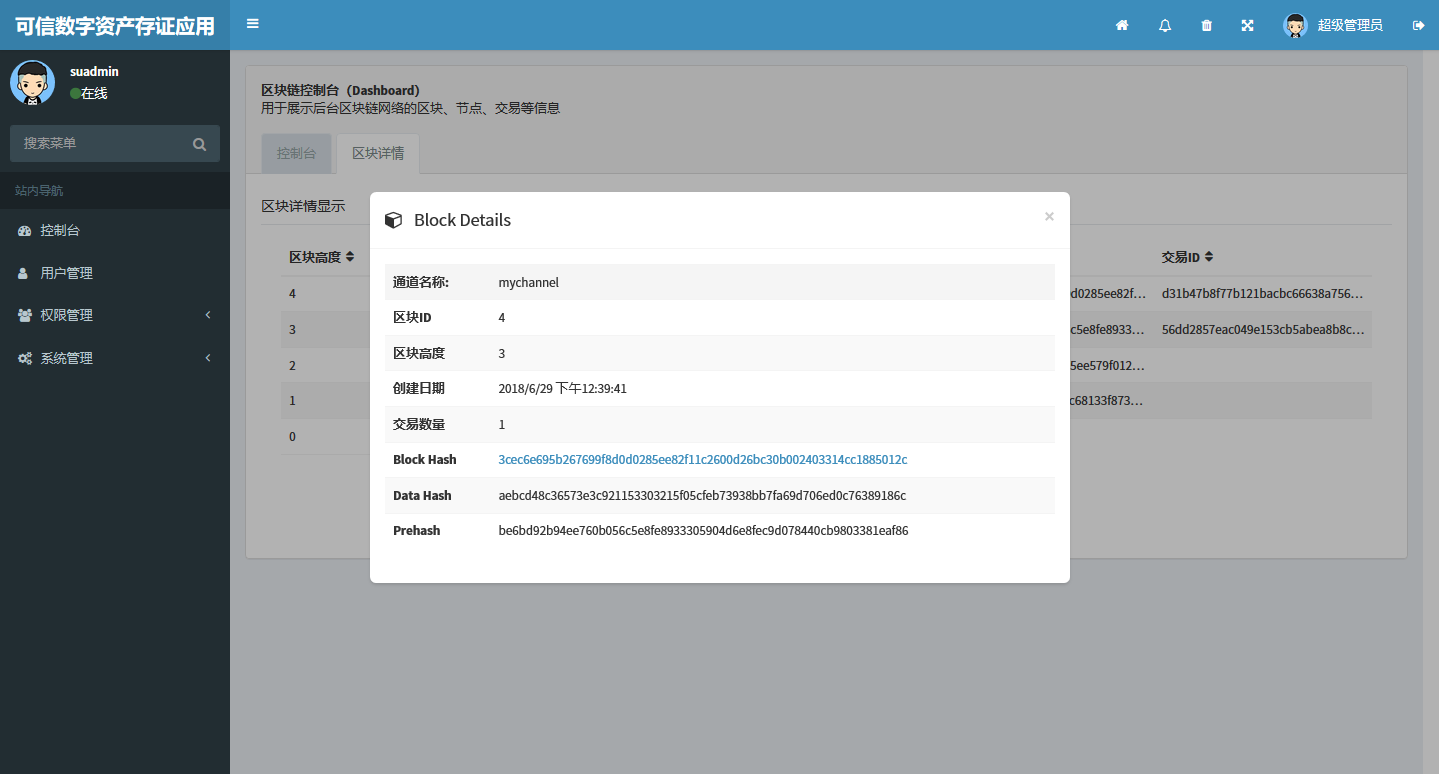


1. 查看区块链相关信息

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

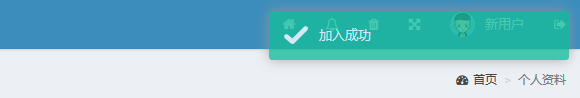
显示资产详细信息

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图像识别程序，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。