[1引言 8](#_Toc518075628)

[1.1编写目的 8](#_Toc518075629)

[1.2背景 8](#_Toc518075630)

[1.3定义 8](#_Toc518075631)

[1.4参考资料 9](#_Toc518075632)

[2程序系统的结构 9](#_Toc518075633)

[3识别关键信息（2001）设计说明 11](#_Toc518075634)

[3.1程序描述 11](#_Toc518075635)

[3.2功能 11](#_Toc518075636)

[3.3性能 12](#_Toc518075637)

[3.4输入项 12](#_Toc518075638)

[3.5输出项 12](#_Toc518075639)

[3.6算法 12](#_Toc518075640)

[3.7流程逻辑 13](#_Toc518075641)

[3.8接口 13](#_Toc518075642)

[3.9存储分配 13](#_Toc518075643)

[3.10注释设计 13](#_Toc518075644)

[3.11限制条件 13](#_Toc518075645)

[3.12测试计划 13](#_Toc518075646)

[3.13尚未解决的问题 14](#_Toc518075647)

[4保存资产（2002）设计说明 14](#_Toc518075648)

[4.1程序描述 14](#_Toc518075649)

[4.2功能 14](#_Toc518075650)

[4.3性能 14](#_Toc518075651)

[4.4输入项 14](#_Toc518075652)

[4.5输出项 15](#_Toc518075653)

[4.6算法 15](#_Toc518075654)

[4.7流程逻辑 15](#_Toc518075655)

[4.8接口 16](#_Toc518075656)

[4.9存储分配 16](#_Toc518075657)

[4.10注释设计 16](#_Toc518075658)

[4.11限制条件 16](#_Toc518075659)

[4.12测试计划 16](#_Toc518075660)

[4.13尚未解决的问题 16](#_Toc518075661)

[5资产鉴权（2003）设计说明 17](#_Toc518075662)

[5.1程序描述 17](#_Toc518075663)

[5.2功能 17](#_Toc518075664)

[5.3性能 17](#_Toc518075665)

[5.4输入项 17](#_Toc518075666)

[5.5输出项 17](#_Toc518075667)

[5.6算法 18](#_Toc518075668)

[5.7流程逻辑 18](#_Toc518075669)

[5.8接口 18](#_Toc518075670)

[5.9存储分配 18](#_Toc518075671)

[5.10注释设计 19](#_Toc518075672)

[5.11限制条件 19](#_Toc518075673)

[5.12测试计划 19](#_Toc518075674)

[5.13尚未解决的问题 19](#_Toc518075675)

[6查询资产（2004）设计说明 19](#_Toc518075676)

[6.1程序描述 19](#_Toc518075677)

[6.2功能 19](#_Toc518075678)

[6.3性能 19](#_Toc518075679)

[6.4输入项 20](#_Toc518075680)

[6.5输出项 20](#_Toc518075681)

[6.6算法 20](#_Toc518075682)

[6.7流程逻辑 20](#_Toc518075683)

[6.8接口 21](#_Toc518075684)

[6.9存储分配 21](#_Toc518075685)

[6.10注释设计 21](#_Toc518075686)

[6.11限制条件 21](#_Toc518075687)

[6.12测试计划 21](#_Toc518075688)

[6.13尚未解决的问题 22](#_Toc518075689)

[7删除资产（2005）设计说明 22](#_Toc518075690)

[7.1程序描述 22](#_Toc518075691)

[7.2功能 22](#_Toc518075692)

[7.3性能 22](#_Toc518075693)

[7.4输入项 22](#_Toc518075694)

[7.5输出项 22](#_Toc518075695)

[7.6算法 23](#_Toc518075696)

[7.7流程逻辑 23](#_Toc518075697)

[7.8接口 23](#_Toc518075698)

[7.9存储分配 23](#_Toc518075699)

[7.10注释设计 23](#_Toc518075700)

[7.11限制条件 24](#_Toc518075701)

[7.12测试计划 24](#_Toc518075702)

[7.13尚未解决的问题 24](#_Toc518075703)

[8查看资产详情（2006）设计说明 24](#_Toc518075704)

[8.1程序描述 24](#_Toc518075705)

[8.2功能 24](#_Toc518075706)

[8.3性能 24](#_Toc518075707)

[8.4输入项 24](#_Toc518075708)

[8.5输出项 25](#_Toc518075709)

[8.6算法 25](#_Toc518075710)

[8.7流程逻辑 25](#_Toc518075711)

[8.8接口 25](#_Toc518075712)

[8.9存储分配 25](#_Toc518075713)

[8.10注释设计 25](#_Toc518075714)

[8.11限制条件 26](#_Toc518075715)

[8.12测试计划 26](#_Toc518075716)

[8.13尚未解决的问题 26](#_Toc518075717)

[9下载资产原件（2007）设计说明 26](#_Toc518075718)

[9.1程序描述 26](#_Toc518075719)

[9.2功能 26](#_Toc518075720)

[9.3性能 26](#_Toc518075721)

[9.4输入项 26](#_Toc518075722)

[9.5输出项 27](#_Toc518075723)

[9.6算法 27](#_Toc518075724)

[9.7流程逻辑 27](#_Toc518075725)

[9.8接口 27](#_Toc518075726)

[9.9存储分配 28](#_Toc518075727)

[9.10注释设计 28](#_Toc518075728)

[9.11限制条件 28](#_Toc518075729)

[9.12测试计划 28](#_Toc518075730)

[9.13尚未解决的问题 28](#_Toc518075731)

[10上传资产文件（2008）设计说明 28](#_Toc518075732)

[10.1程序描述 28](#_Toc518075733)

[10.2功能 28](#_Toc518075734)

[10.3性能 29](#_Toc518075735)

[10.4输入项 29](#_Toc518075736)

[10.5输出项 29](#_Toc518075737)

[10.6算法 29](#_Toc518075738)

[10.7流程逻辑 29](#_Toc518075739)

[10.8接口 30](#_Toc518075740)

[10.9存储分配 30](#_Toc518075741)

[10.10注释设计 30](#_Toc518075742)

[10.11限制条件 30](#_Toc518075743)

[10.12测试计划 30](#_Toc518075744)

[10.13尚未解决的问题 31](#_Toc518075745)

[11 API交互程序-注册用户（3001）设计说明 31](#_Toc518075746)

[11.1程序描述 31](#_Toc518075747)

[11.2功能 31](#_Toc518075748)

[11.3性能 31](#_Toc518075749)

[11.4输入项 31](#_Toc518075750)

[11.5输出项 31](#_Toc518075751)

[11.6算法 32](#_Toc518075752)

[11.7流程逻辑 32](#_Toc518075753)

[11.8接口 32](#_Toc518075754)

[11.9存储分配 32](#_Toc518075755)

[11.10注释设计 32](#_Toc518075756)

[11.11限制条件 32](#_Toc518075757)

[11.12测试计划 33](#_Toc518075758)

[11.13尚未解决的问题 33](#_Toc518075759)

[12 API交互程序-创建通道（3002）设计说明 33](#_Toc518075760)

[12.1程序描述 33](#_Toc518075761)

[12.2功能 33](#_Toc518075762)

[12.3性能 33](#_Toc518075763)

[12.4输入项 33](#_Toc518075764)

[12.5输出项 33](#_Toc518075765)

[12.6算法 34](#_Toc518075766)

[12.7流程逻辑 34](#_Toc518075767)

[12.8接口 34](#_Toc518075768)

[12.9存储分配 34](#_Toc518075769)

[12.10注释设计 34](#_Toc518075770)

[12.11限制条件 34](#_Toc518075771)

[12.12测试计划 35](#_Toc518075772)

[12.13尚未解决的问题 35](#_Toc518075773)

[13 API交互程序-安装链码（3003）设计说明 35](#_Toc518075774)

[13.1程序描述 35](#_Toc518075775)

[13.2功能 35](#_Toc518075776)

[13.3性能 35](#_Toc518075777)

[13.4输入项 35](#_Toc518075778)

[13.5输出项 36](#_Toc518075779)

[13.6算法 36](#_Toc518075780)

[13.7流程逻辑 36](#_Toc518075781)

[13.8接口 36](#_Toc518075782)

[13.9存储分配 36](#_Toc518075783)

[13.10注释设计 36](#_Toc518075784)

[13.11限制条件 37](#_Toc518075785)

[13.12测试计划 37](#_Toc518075786)

[13.13尚未解决的问题 37](#_Toc518075787)

[14 API交互程序-实例化链码（3004）设计说明 37](#_Toc518075788)

[14.1程序描述 37](#_Toc518075789)

[14.2功能 37](#_Toc518075790)

[14.3性能 37](#_Toc518075791)

[14.4输入项 38](#_Toc518075792)

[14.5输出项 38](#_Toc518075793)

[14.6算法 38](#_Toc518075794)

[14.7流程逻辑 38](#_Toc518075795)

[14.8接口 38](#_Toc518075796)

[14.9存储分配 39](#_Toc518075797)

[14.10注释设计 39](#_Toc518075798)

[14.11限制条件 39](#_Toc518075799)

[14.12测试计划 39](#_Toc518075800)

[14.13尚未解决的问题 39](#_Toc518075801)

[15 API交互程序-调用链码（3005）设计说明 39](#_Toc518075802)

[15.1程序描述 39](#_Toc518075803)

[15.2功能 40](#_Toc518075804)

[15.3性能 40](#_Toc518075805)

[15.4输入项 40](#_Toc518075806)

[15.5输出项 40](#_Toc518075807)

[15.6算法 40](#_Toc518075808)

[15.7流程逻辑 40](#_Toc518075809)

[15.8接口 41](#_Toc518075810)

[15.9存储分配 41](#_Toc518075811)

[15.10注释设计 41](#_Toc518075812)

[15.11限制条件 41](#_Toc518075813)

[15.12测试计划 41](#_Toc518075814)

[15.13尚未解决的问题 42](#_Toc518075815)

[16 API交互程序-查询链码（3006）设计说明 42](#_Toc518075816)

[16.1程序描述 42](#_Toc518075817)

[16.2功能 42](#_Toc518075818)

[16.3性能 42](#_Toc518075819)

[16.4输入项 42](#_Toc518075820)

[16.5输出项 42](#_Toc518075821)

[16.6算法 43](#_Toc518075822)

[16.7流程逻辑 43](#_Toc518075823)

[16.8接口 43](#_Toc518075824)

[16.9存储分配 43](#_Toc518075825)

[16.10注释设计 43](#_Toc518075826)

[16.11限制条件 43](#_Toc518075827)

[16.12测试计划 44](#_Toc518075828)

[16.13尚未解决的问题 44](#_Toc518075829)

[17 API交互程序-重建通道（3007）设计说明 44](#_Toc518075830)

[17.1程序描述 44](#_Toc518075831)

[17.2功能 44](#_Toc518075832)

[17.3性能 44](#_Toc518075833)

[17.4输入项 44](#_Toc518075834)

[17.5输出项 45](#_Toc518075835)

[17.6算法 45](#_Toc518075836)

[17.7流程逻辑 45](#_Toc518075837)

[17.8接口 45](#_Toc518075838)

[17.9存储分配 45](#_Toc518075839)

[17.10注释设计 46](#_Toc518075840)

[17.11限制条件 46](#_Toc518075841)

[17.12测试计划 46](#_Toc518075842)

[17.13尚未解决的问题 46](#_Toc518075843)

[18 API交互程序-升级链码（3008）设计说明 46](#_Toc518075844)

[18.1程序描述 46](#_Toc518075845)

[18.2功能 46](#_Toc518075846)

[18.3性能 46](#_Toc518075847)

[18.4输入项 47](#_Toc518075848)

[18.5输出项 47](#_Toc518075849)

[18.6算法 47](#_Toc518075850)

[18.7流程逻辑 47](#_Toc518075851)

[18.8接口 48](#_Toc518075852)

[18.9存储分配 48](#_Toc518075853)

[18.10注释设计 48](#_Toc518075854)

[18.11限制条件 48](#_Toc518075855)

[18.12测试计划 48](#_Toc518075856)

[18.13尚未解决的问题 49](#_Toc518075857)

[19 API交互程序-查询区块信息（3009）设计说明 49](#_Toc518075858)

[19.1程序描述 49](#_Toc518075859)

[19.2功能 49](#_Toc518075860)

[19.3性能 49](#_Toc518075861)

[19.4输入项 49](#_Toc518075862)

[19.5输出项 49](#_Toc518075863)

[19.6算法 50](#_Toc518075864)

[19.7流程逻辑 50](#_Toc518075865)

[19.8接口 50](#_Toc518075866)

[19.9存储分配 50](#_Toc518075867)

[19.10注释设计 50](#_Toc518075868)

[19.11限制条件 51](#_Toc518075869)

[19.12测试计划 51](#_Toc518075870)

[19.13尚未解决的问题 51](#_Toc518075871)

[20底层区块链部署设计说明 51](#_Toc518075872)

[20.1程序描述 51](#_Toc518075873)

[20.2功能 51](#_Toc518075874)

[20.3性能 51](#_Toc518075875)

[20.4输入项 51](#_Toc518075876)

[20.5输出项 52](#_Toc518075877)

[20.6算法 52](#_Toc518075878)

[20.7流程逻辑 52](#_Toc518075879)

[20.8接口 52](#_Toc518075880)

[20.9存储分配 52](#_Toc518075881)

[20.10注释设计 52](#_Toc518075882)

[20.11限制条件 52](#_Toc518075883)

[20.12测试计划 53](#_Toc518075884)

[20.13尚未解决的问题 53](#_Toc518075885)

**详细设计说明书**

# 1引言

## 1.1编写目的

本文档编写的目的是详细地介绍系统模块的设计，在概要设计的基础上，对功能模块进行过程描述，设计功能模块的内部细节，包括流程控制，算法和数据结构的设计，为开发提供必要的说明。

概要设计解决了软件系统总体的结构设计问题，包括整个软件系统的结构、模块划分、模块功能和模块间的联系等。详细设计解决了如何实现各个模块的内部功能，为系统的更详细的模块设计，对内部实现细节作精确的描述。

预期读者：

对《详细设计说明书》进行评审和确认的相关业务、技术人员

本项目编码阶段的软件开发工程师

本项目测试阶段的软件测试工程师

本项目运行阶段的工程师

其他需要查看本文档的人员

## 1.2背景

说明：

1. 待开发软件系统的名称：可信数字资产存证应用；
2. 项目的任务提出者：新华三技术有限公司
3. 开发者：第七届中国软件杯石家庄铁道大学软件工程系Genesis Block参赛组
4. 用户：所有使用该系统的人员
5. 运行该程序系统的计算中心：新华三技术有限公司虚拟服务器

## 1.3定义

区块链：区块链是用分布式数据库识别、传播和记载信息的智能化对等网络, 也称为价值互联网。

数字资产：是经过二进制编码的任何被授权使用的文本或媒体资源，包括文本内容、图片和多媒体，包括但不仅限于网站及其内容、应用软件、代码、电子文档、图片内容、媒体内容等。

关键信息：根据数字资产的不同有着其不同的关键信息。

资产鉴权：利用特定方式对资产进行鉴权，来判定其是否被篡改。

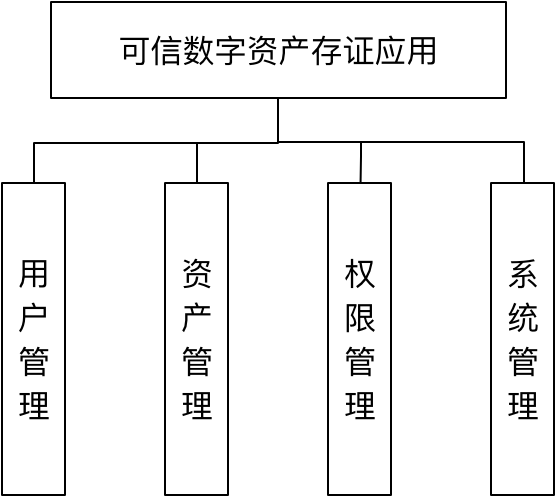
MD5：即Message-Digest Algorithm 5，为计算机安全领域广泛使用的一种散列函数，用以提供消息的完整性保护。

## 1.4参考资料

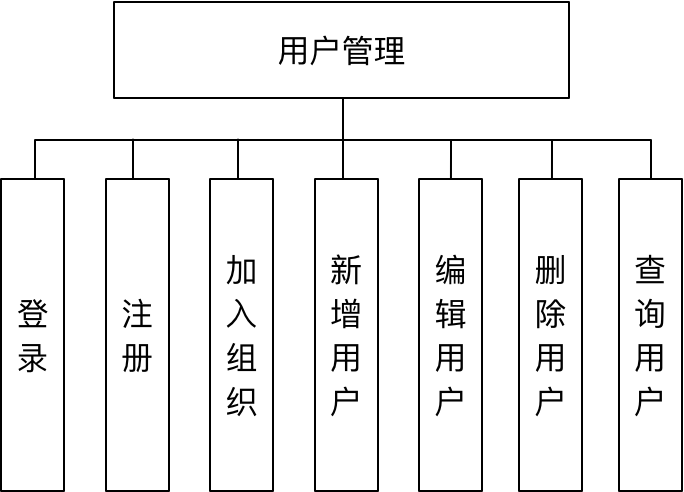
1. 第七届中国软件杯可信数字资产存证应用题目说明
2. 《可信数字资产存证应用需求规格说明书》
3. 《可信数字资产存证应用概要设计说明书》

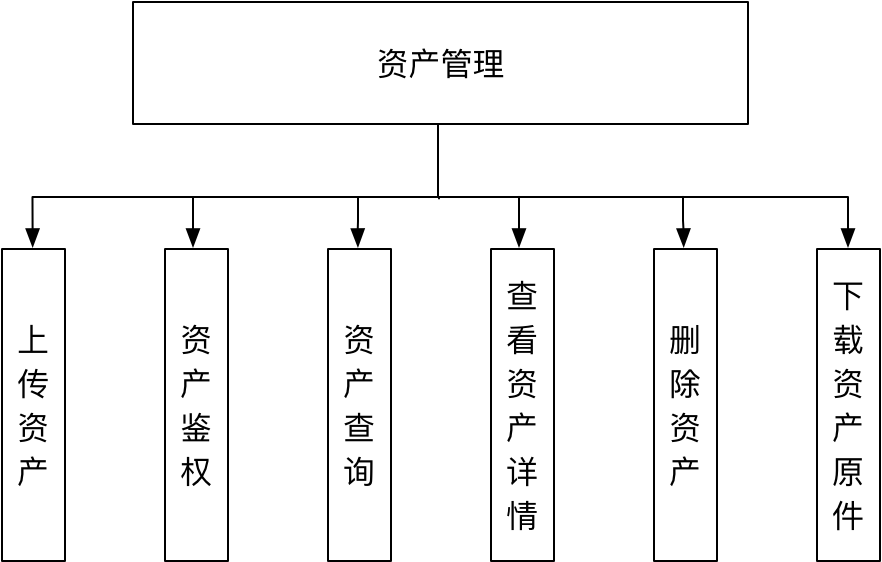
# 2程序系统的结构

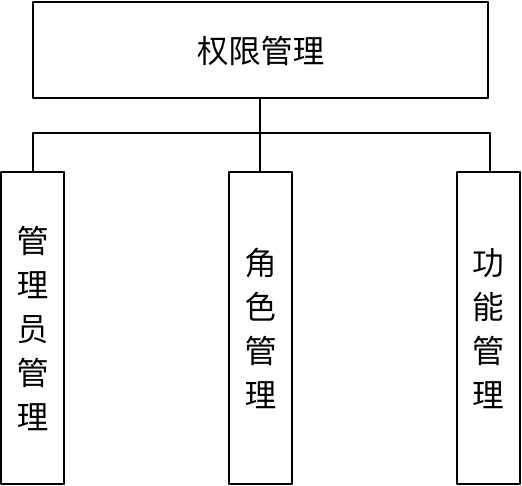
系统结构：

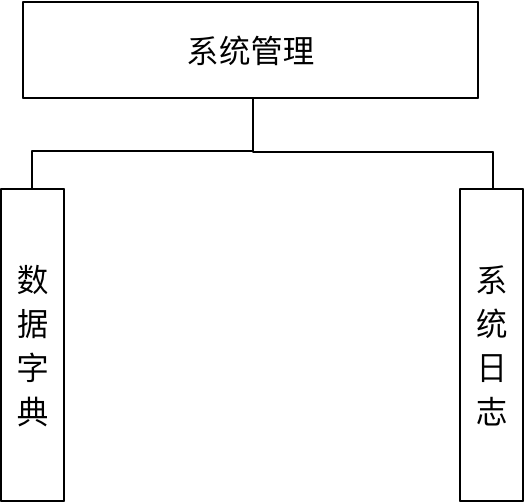


子模块

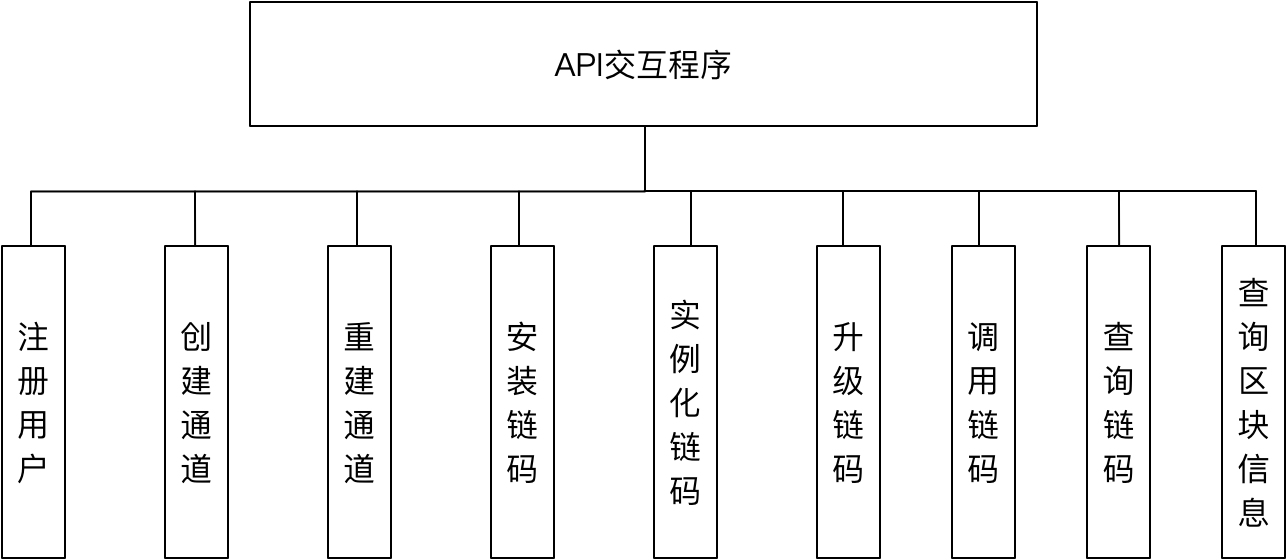








API交互程序：



# 3识别关键信息（2001）设计说明

## 3.1程序描述

用户上传资产和资产鉴权时上传资产文件，需要识别资产的关键信息

## 3.2功能

识别资产文件的关键信息

## 3.3性能

需要在3秒内识别出关键信息对返回给用户

## 3.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| 资产文件 | file | File | 文件选择框选择 |
| 资产类型编码 | code | String | 下拉选择框 |

## 3.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 是否成功 | success | boolean | 请求是否成功 |
| 数据 | data | Object | 返回的数据，为关键信息的JSON格式的字符串 |
| 消息 | message | String | 返回的消息 |
| 状态码 | code | String | 状态码 |

## 3.6算法

资产关键信息识别：

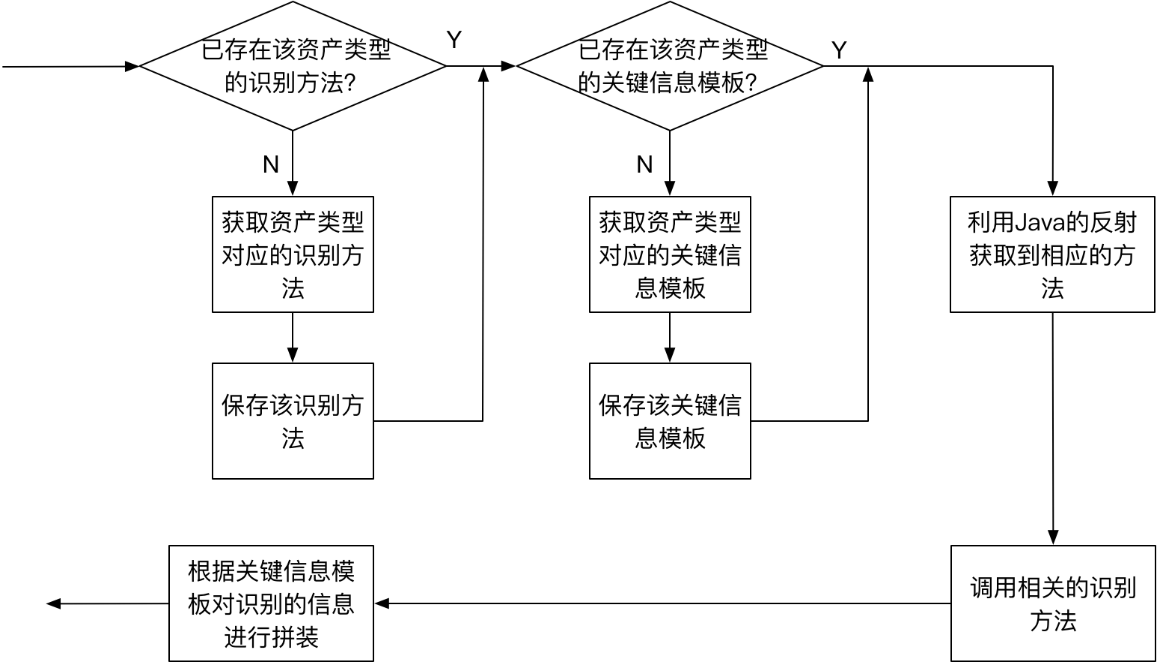
图片文件：采用百度OCR提供的文字识别服务

PDF文件：采用Apache的pdfBox开源工具进行提取

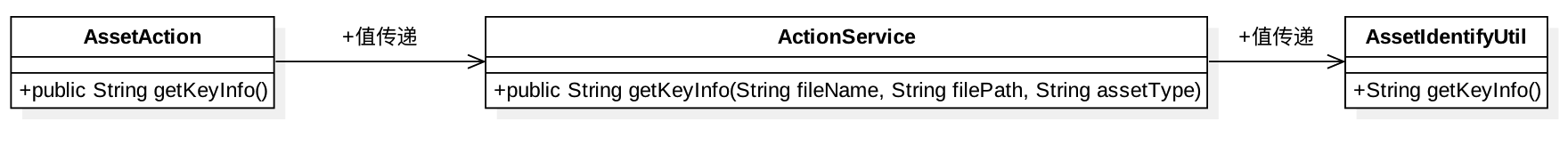
关键信息拼装：

采用HashMap数据结构对关键信息进行拼装，将关键信息模板进行分割，然后依次遍历提取的关键信息，从关键信息中查找出该项关键信息然后保存到HashMap对象中。遍历完成即可成功拼装该类资产所有的关键信息。

## 3.7流程逻辑



## 3.8接口



## 3.9存储分配

无

## 3.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 3.11限制条件

需要连接网络

## 3.12测试计划

技术要求：需要进行单元测试

输入数据：资产文件为身份证图片文件，资产类型为身份证

预期结果：返回该身份证的关键信息，包括姓名、性别、民族、出生、公民身份号码等

进度安排：开发完成后即进行测试

## 3.13尚未解决的问题

无

# 4保存资产（2002）设计说明

## 4.1程序描述

用户登录后希望上传资产，则进行资产上传

## 4.2功能

保存用户填写的资产信息到数据库和区块链。

## 4.3性能

需要在7秒内完成所有的保存操作，其中能够在1秒内完成保存到数据库的操作，能够在6秒内完成保存到区块链的操作。

## 4.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| 资产类型编码 | Code | String | 下拉选择框 |
| 标签 | Tag | String | 文本输入框 |
| 关键信息 | keyInfo | String | Js函数获取 |
| 文件名称 | fileName | String | Js函数获取 |
| 文件路径 | filePath | String | Js函数获取 |
| 文件类型 | fileType | String | Js函数获取 |
| 文件通用类型 | commonType | String | Js函数获取 |

## 4.5输出项

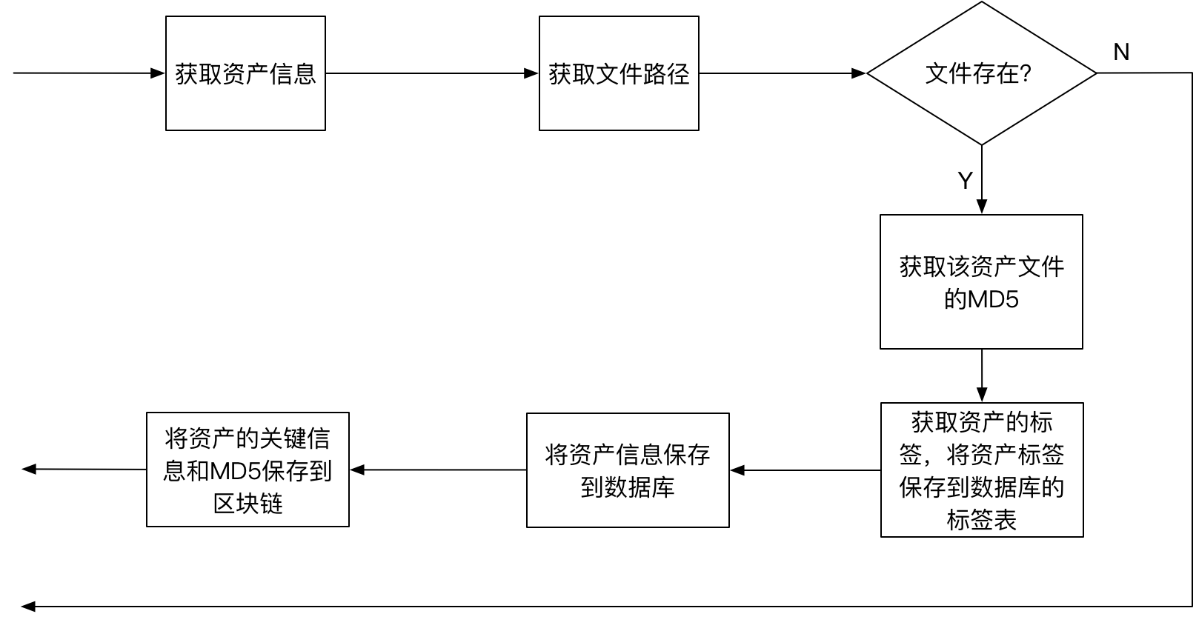
|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 是否成功 | success | boolean | 请求是否成功 |
| 数据 | data | Object | 返回的数据，如果为scucess为保存成功，否则保存失败 |
| 消息 | message | String | 返回的消息 |
| 状态码 | code | String | 状态码 |

## 4.6算法

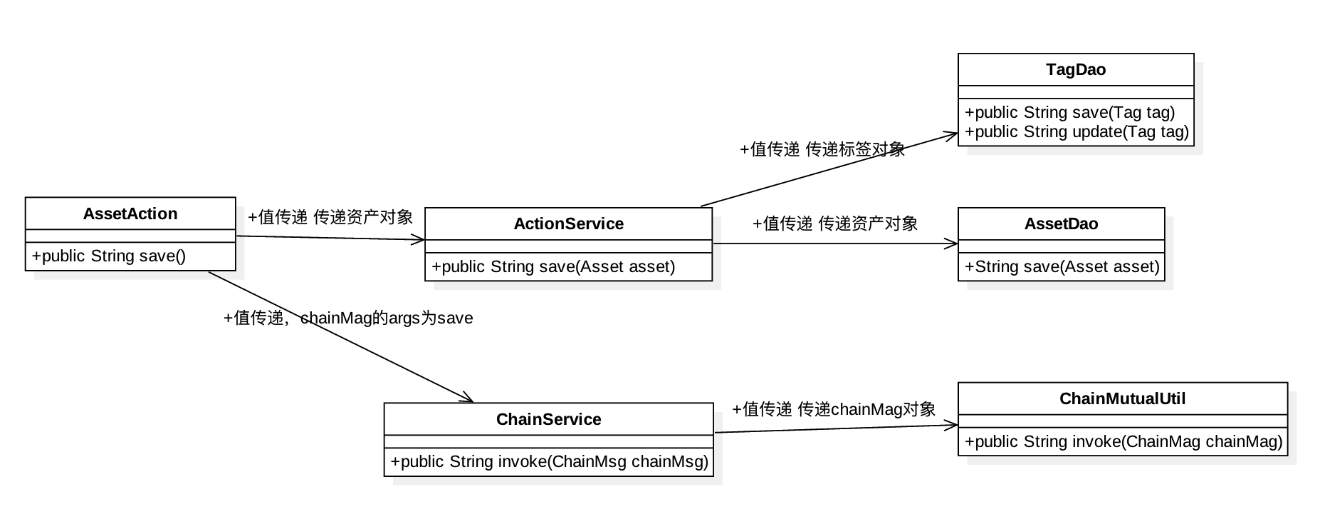
处理标签:

获取到用户填写的标签字符串，然后根据空格分割为字符串数组，然后遍历该数组，依次查询标签是否存在，如果存在则将标签数量加一，否则插入一个新的标签到数据库。

## 4.7流程逻辑



## 4.8接口



## 4.9存储分配

无

## 4.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 4.11限制条件

需要数据库开启和区块链网络开启,并且可用

## 4.12测试计划

技术要求：需要进行单元测试

输入数据：资产文件为身份证图片文件，资产类型为身份证，进行上传，然后会返回资产关键信息、文件名称、文件路径、文件类型、文件普通类型，资产名称，资产标签，资产类型。

预期结果：返回保存成功

进度安排：开发完成后即进行测试

## 4.13尚未解决的问题

无

# 5资产鉴权（2003）设计说明

## 5.1程序描述

用户登录后进行查询资产，查看资产详细信息，点击鉴权，进行资产鉴权

## 5.2功能

对用户的资产进行鉴权，对用户上传的鉴权文件进行关键信息提取然后从区块链中读取原始资产信息进行比对，然后返回比对结果。

## 5.3性能

需要在2秒内完成识别鉴权资产关键信息操作，其中能够在2秒内完成从区块链中读取资产关键信息的操作，能够在1秒内完成资产关键信息比对操作。

## 5.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| 文件名称 | fileName | String | Js函数获取 |
| 文件路径 | filePath | String | Js函数获取 |
| 资产id | Id | String | 隐藏输入框 |

## 5.5输出项

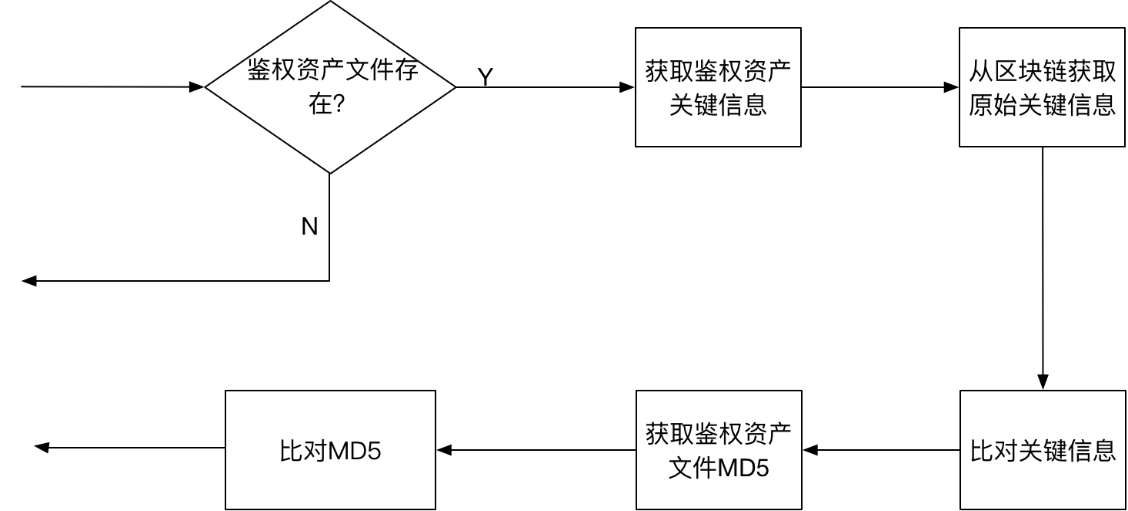
|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 是否成功 | success | boolean | 请求是否成功 |
| 数据 | data | Object | 返回的数据 |
| 消息 | message | String | 返回的消息 |
| 状态码 | code | String | 状态码 |
| 数据子项 | Data. nKeyInfo | String | 鉴权资产文件的关键信息JSON字符串 |
| 数据子项 | Data. chainAsset | String | 资产原件信息JSON字符串 |
| 数据子项 | Data.compareResult | String | 关键信息比对结果JSON字符创 |
| 数据子项 | Data.nMd5 | String | 鉴权资产文件的MD5 |
| 数据子项 | Data. md5CompareResult | String | Md5比对结果 |

## 5.6算法

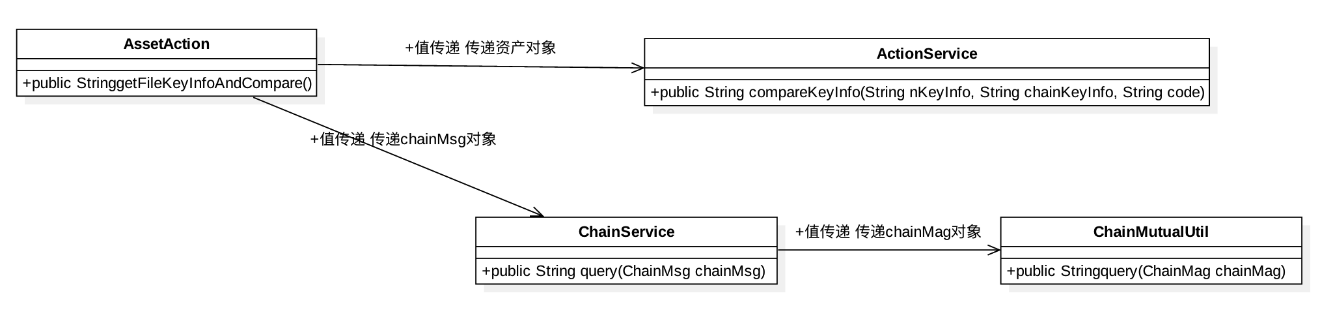
比对关键信息：

获取到关键信息模板，将关键信息模板进行分割，得到关键信息字符串数组，然后遍历该数组，然后依次从鉴权资产的关键信息中获取和从区块链中获取的资产原始关键信息，然后进行对比，然后将比对结果存储到HashMap对象中。

## 5.7流程逻辑



## 5.8接口



## 5.9存储分配

无

## 5.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 5.11限制条件

需要数据库开启和区块链网络开启,并且可用

## 5.12测试计划

技术要求：需要进行单元测试

输入数据：对系统用户xiaoxi的身份证资产进行鉴权，上传在原始身份证图片基础上经过修改公民身份号码和姓名的图片，然后进行鉴权。

预期结果：返回鉴权图片的关键信息，显示姓名和公民身份号码不一致，鉴权资产被篡改。

进度安排：开发完成后即进行测试

## 5.13尚未解决的问题

无

# 6查询资产（2004）设计说明

## 6.1程序描述

用户登录后查看个人资产，点击个人资产进行查询资产，也可以点击资产类型属性菜单进行查询资产信息，或通过输入资产名称进行查询，或通过点击标签进行查询。

## 6.2功能

根据条件对用户的资产进行查询

## 6.3性能

在2秒内完成查询操作，并将结果在页面显示

## 6.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| 资产文件类型 | commonType | String | 属性菜单点击 |
| 标签 | tag | String | 点击 |
| 资产名称 | name | String | 文本输入框 |
| 页码 | page | int | 从HTML标签获取，初始值为0 |

## 6.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 是否成功 | success | boolean | 请求是否成功 |
| 数据 | data | Object | 返回的资产列表的JSON字符串 |
| 消息 | message | String | 返回的消息 |
| 状态码 | code | String | 状态码 |

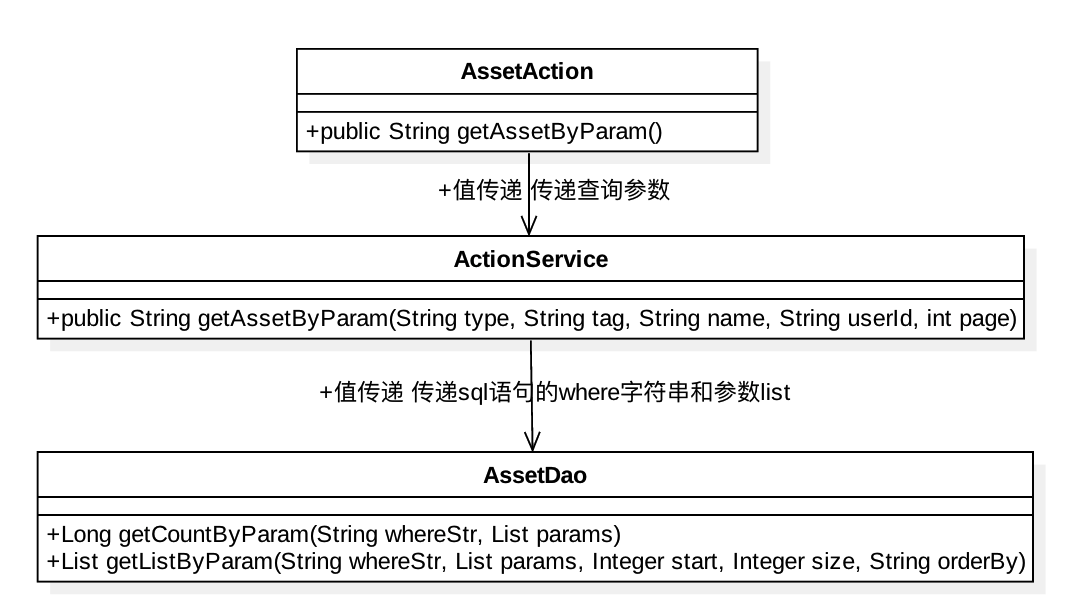
## 6.6算法

不涉及具体算法，主要为对sql语句进行拼装

## 6.7流程逻辑



## 6.8接口



## 6.9存储分配

无

## 6.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 6.11限制条件

需要用户登录，数据库开启，并且可用

## 6.12测试计划

技术要求：需要进行单元测试

输入数据：无

预期结果：显示该用户所有的资产信息

输入数据：点击树形菜单-图片

预期结果：显示该用户所有的图片资产信息

输入数据：点击证件标签

预期结果：显示该用户所有的标签包含证件的资产信息

进度安排：开发完成后即进行测试

## 6.13尚未解决的问题

无

# 7删除资产（2005）设计说明

## 7.1程序描述

用户登录后点击个人资产，然后再点击资产，显示资产的详细信息，然后点击删除，提示是否删除，点击确定，删除资产

## 7.2功能

将数据库和区块链中指定id的资产进行删除，将其资产文件删除，然后将相关的标签的数量减一。

## 7.3性能

其中能够在1秒内完成数据库的删除操作，能够在3秒内完成从区块链删除的操作。

## 7.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| 资产id | id | String | Js设置id参数 |

## 7.5输出项

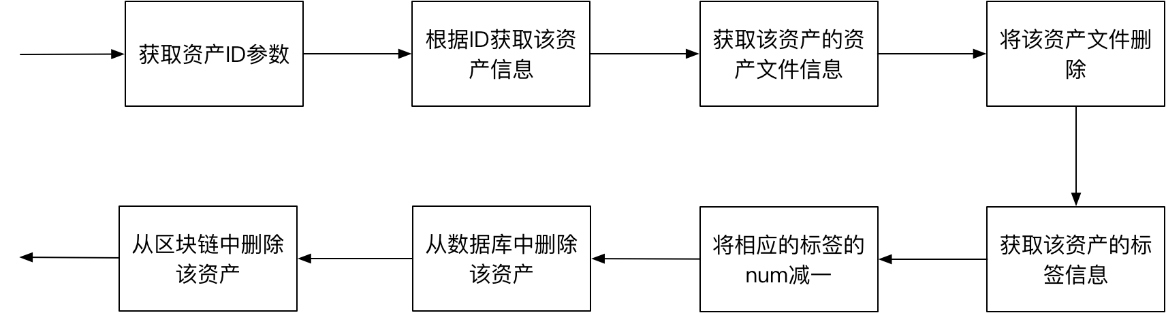
|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 是否成功 | success | boolean | 请求是否成功 |
| 数据 | data | Object | 返回的数据，如果为scucess为删除成功，否则删除失败 |
| 消息 | message | String | 返回的消息 |
| 状态码 | code | String | 状态码 |

## 7.6算法

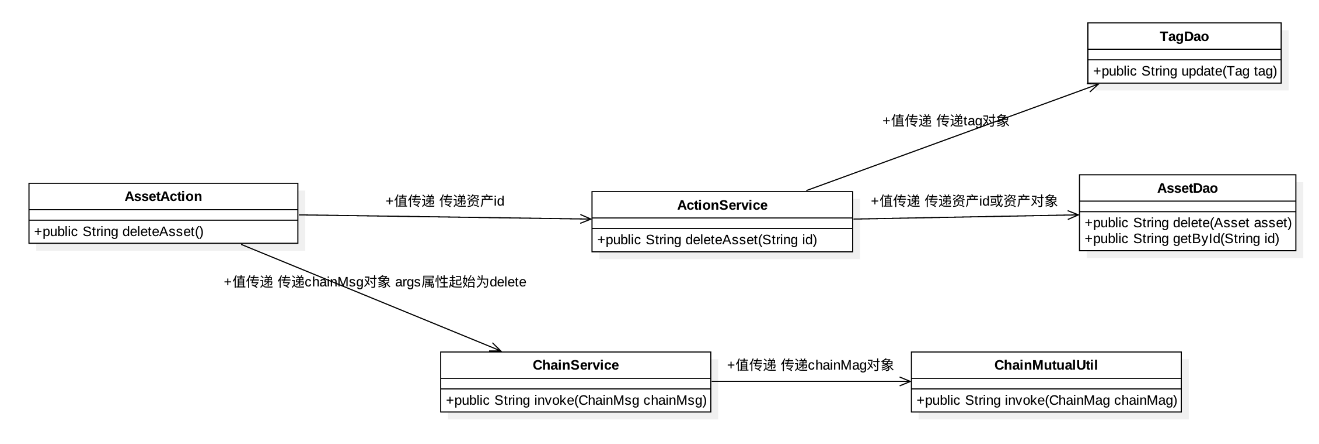
不涉及具体的算法，主要为处理标签:

先查询到该资产所有的标签，然后更新标签数据表，将标签的num属性减一。

## 7.7流程逻辑



## 7.8接口



## 7.9存储分配

无

## 7.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 7.11限制条件

需要数据库开启和区块链网络开启,并且可用

## 7.12测试计划

技术要求：需要进行单元测试

输入数据：使用xiaoxi用户进行登录，然后点击我的身份证资产，点击删除按钮

预期结果：返回删除成功

进度安排：开发完成后即进行测试

## 7.13尚未解决的问题

无

# 8查看资产详情（2006）设计说明

## 8.1程序描述

用户登录后进入个人资产页面，选择一个资产，然后进入资产详情页面显示资产详情。

## 8.2功能

获取指定资产id的资产的详细信息。

## 8.3性能

能够在1秒内查询出资产信息，然后再页面进行显示

## 8.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| 资产id | id | String | Js获取的参数 |

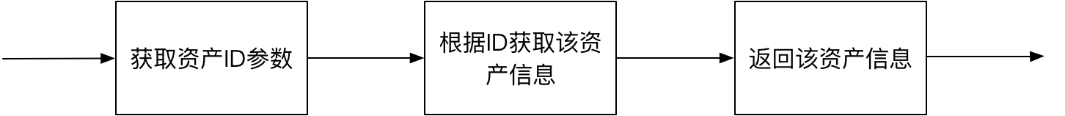
## 8.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 是否成功 | success | boolean | 请求是否成功 |
| 数据 | data | Object | 返回的数据，资产详细信息的JSON字符串 |
| 消息 | message | String | 返回的消息 |
| 状态码 | code | String | 状态码 |

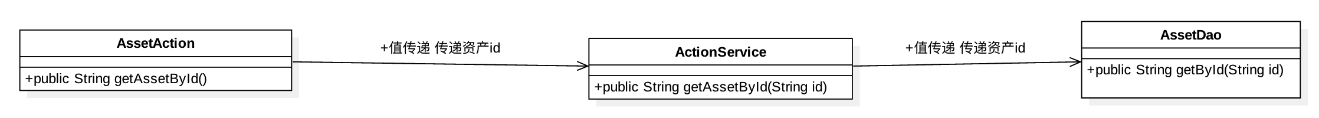
## 8.6算法

无

## 8.7流程逻辑



## 8.8接口



## 8.9存储分配

无

## 8.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 8.11限制条件

需要数据库开启,并且可用

## 8.12测试计划

技术要求：需要进行单元测试

输入数据：使用系统用户xiaoxi用户登录系统，然后进入个人资产页面，点击我的身份证资产。

预期结果：显示该资产的详细信息

进度安排：开发完成后即进行测试

## 8.13尚未解决的问题

无

# 9下载资产原件（2007）设计说明

## 9.1程序描述

用户登录后进入个人资产页面，选择一个资产，然后进入资产详情页面显示资产详情，点击下载原件进行资产原件的下载。

## 9.2功能

下载指定资产id的资产原始文件。

## 9.3性能

无

## 9.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| 文件路径 | FilePath | String | Get url参数 |
| 文件名称 | fileName | String | Get url参数 |

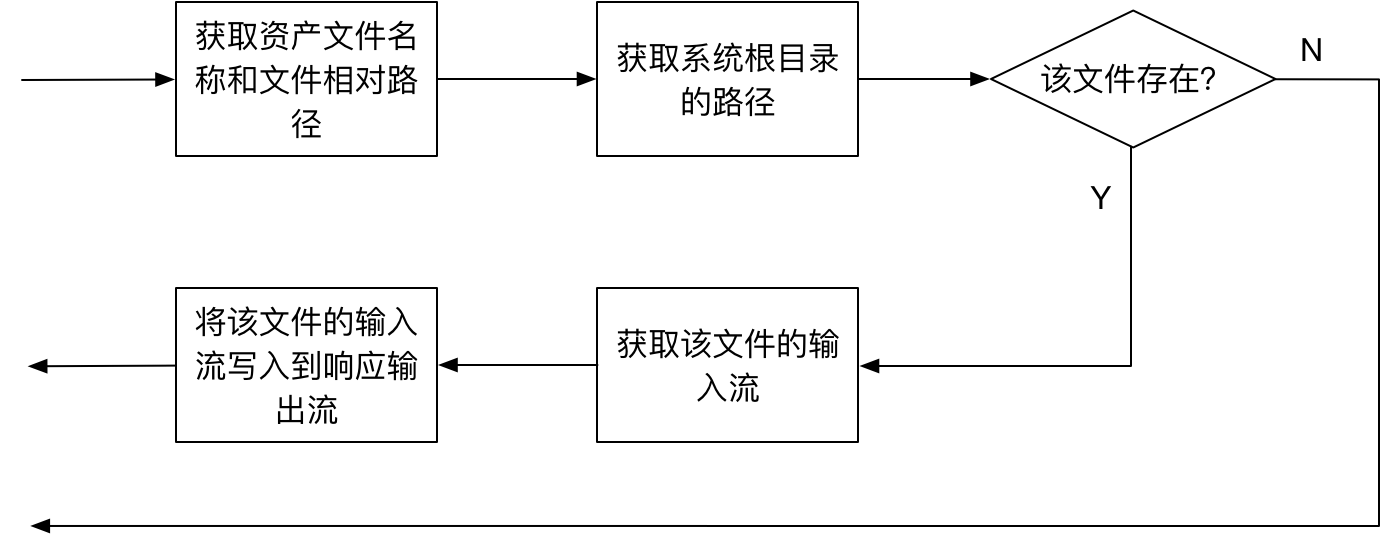
## 9.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 输入流 | 无 | InputStream | 下载文件的输入流 |

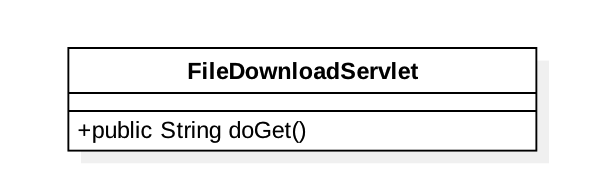
## 9.6算法

无

## 9.7流程逻辑



## 9.8接口



## 9.9存储分配

无

## 9.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 9.11限制条件

需要数据库开启,并且可用

## 9.12测试计划

技术要求：需要进行单元测试

输入数据：使用系统用户xiaoxi用户登录系统，然后进入个人资产页面，点击我的身份证资产，在资产详细信息最下面点击下载原件按钮

预期结果：开始下载该资产原件

进度安排：开发完成后即进行测试

## 9.13尚未解决的问题

无

# 10上传资产文件（2008）设计说明

## 10.1程序描述

用户登录后进行上传资产，或者资产鉴权时将资产文件进行上传

## 10.2功能

上传资产文件

## 10.3性能

无具体要求，要求能够在用户可接受范围内（5秒内）上传成功

## 10.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| 资产文件 | File | File | 页面文件选择框 |

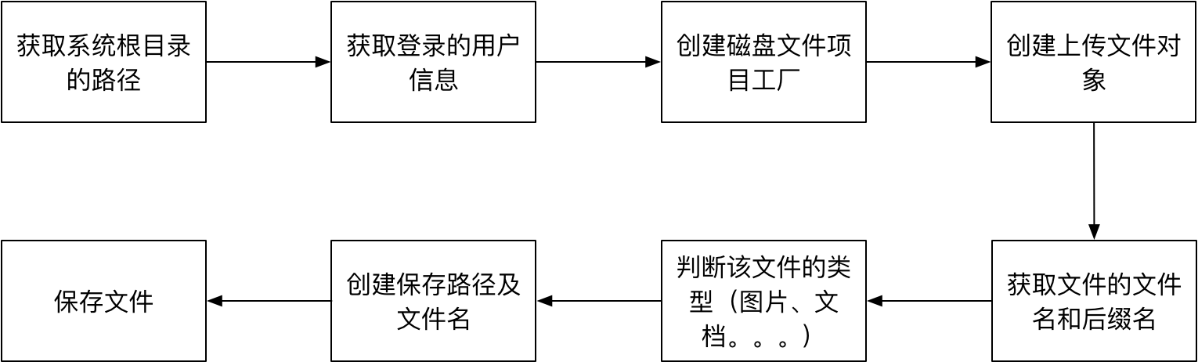
## 10.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 文件名称 | fileName | String | 上传的文件的保存的文件名称 |
| 文件路径 | filePath | String | 上传的文件的相对于web系统的保存路径 |
| 文件类型 | fileType | String | 该上传的文件的类型，例如png,pdf |
| 文件通用类型 | commonType | String | 该上传的文件的通用类型，例如图片、文档 |

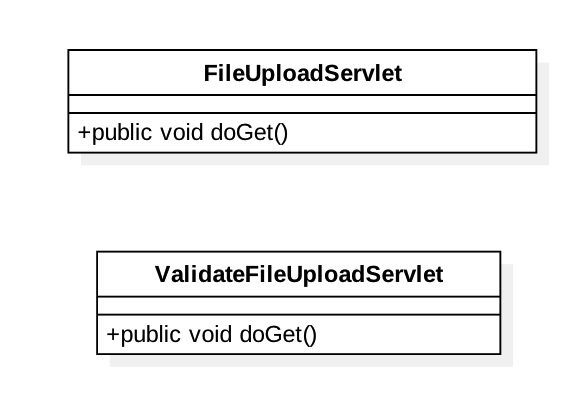
## 10.6算法

没有具体算法要求，采用Apache的fileupload开源工具进行接收文件上传

## 10.7流程逻辑



## 10.8接口



## 10.9存储分配

无

## 10.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 10.11限制条件

无

## 10.12测试计划

技术要求：需要进行单元测试

输入数据：使用系统用户xiaoxi用户登陆系统，点击上传资产，选择资产文件，点击开始上传

预期结果：文件上传成功

进度安排：开发完成后即进行测试

## 10.13尚未解决的问题

无

# 11 API交互程序-注册用户（3001）设计说明

## 11.1程序描述

为了实现WEB系统与区块链网络进行交互，使用fabric的Java SDK编写与区块链的交互程序，实现与区块链的交互，每次交互需要设置用户上下文，故需要注册用户

## 11.2功能

注册用于操作区块链的管理员用户

## 11.3性能

无

## 11.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| 用户名 | Username | String | POST请求参数 |

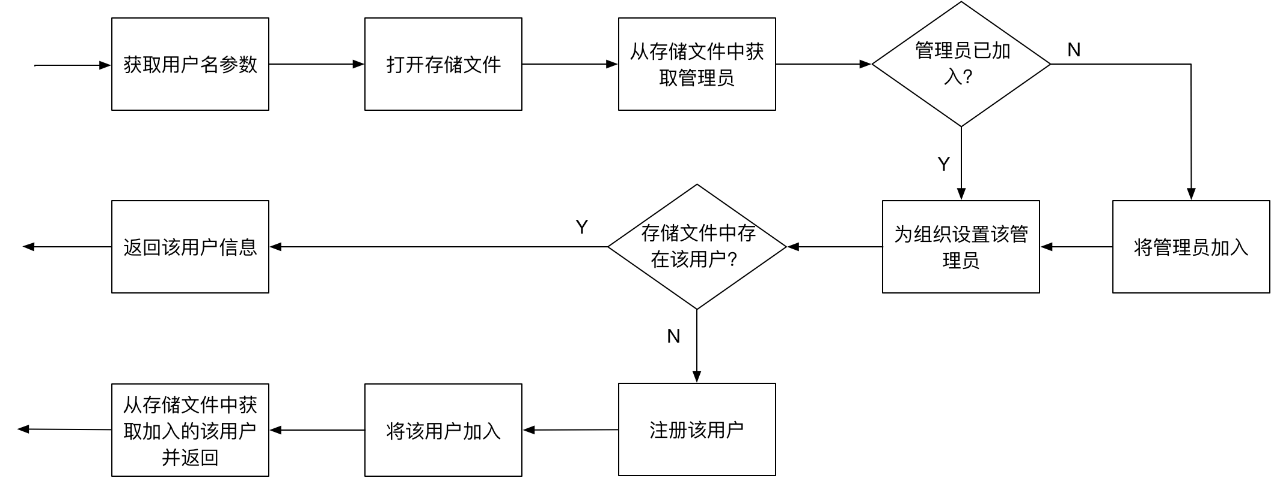
## 11.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 操作结果及Token秘钥 | result | String | 注册成功则返回成功标识以及jwt Token秘钥 |

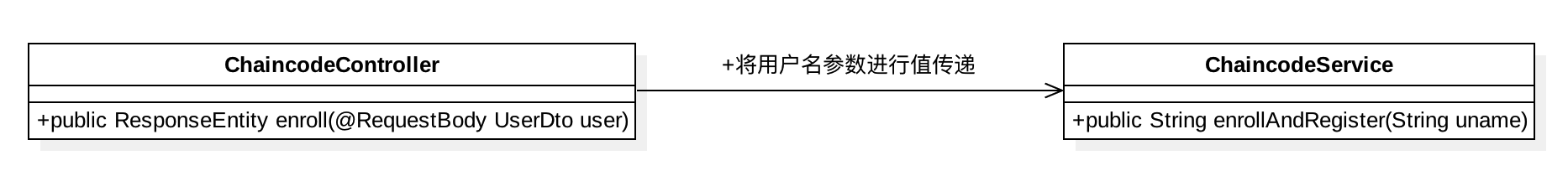
## 11.6算法

无具体算法，采用fabric 1.0 Java SDK

## 11.7流程逻辑



## 11.8接口



## 11.9存储分配

读写HyperledgerEnroll.properties文件

## 11.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 11.11限制条件

区块链正常运行

## 11.12测试计划

技术要求：需要进行单元测试

输入数据：使用curl工具进行调用，传递参数username:test

预期结果：返回User test Enrolled Successfuly jwt:具体的token秘钥字符串

进度安排：开发完成后即进行测试

## 11.13尚未解决的问题

无

# 12 API交互程序-创建通道（3002）设计说明

## 12.1程序描述

区块链需要创建通道，为了方便对区块链网络进行管理，在API交互程序中书写创建通道的接口

## 12.2功能

对区块链网络进行创建通道操作

## 12.3性能

无

## 12.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| Token秘钥 | Authorization | String | POST请求header |

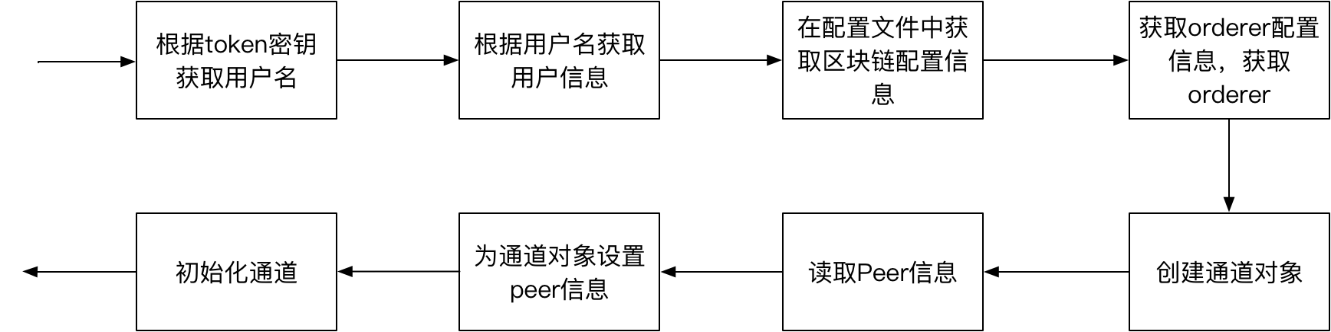
## 12.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 操作结果 | result | String | 创建成功则返回创建成功否则返回发生错误 |

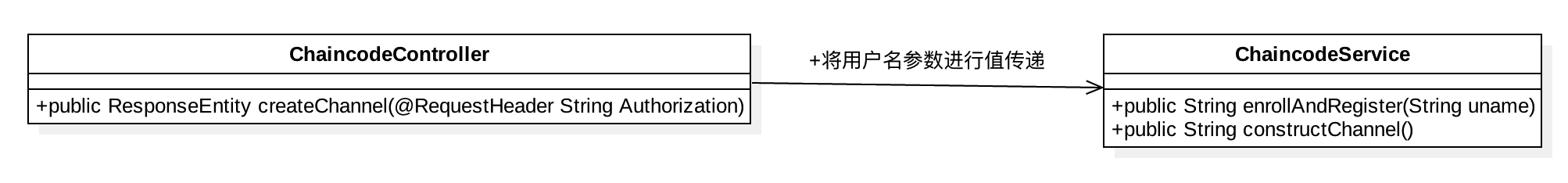
## 12.6算法

无具体算法，采用fabric 1.0 Java SDK

## 12.7流程逻辑



## 12.8接口



## 12.9存储分配

读写HyperledgerEnroll.properties文件

## 12.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 12.11限制条件

区块链正常运行，token秘钥正确

## 12.12测试计划

技术要求：需要进行单元测试

输入数据：使用curl工具进行调用

预期结果：返回channel created successfully

进度安排：开发完成后即进行测试

## 12.13尚未解决的问题

无

# 13 API交互程序-安装链码（3003）设计说明

## 13.1程序描述

区块链需要安装链码，为了方便对区块链网络进行管理，在API交互程序中书写创建链码的接口

## 13.2功能

对区块链网络进行安装链码操作

## 13.3性能

无

## 13.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| Token秘钥 | Authorization | String | POST请求header |
| 链码名称 | myChaincode | String | POST请求参数 |

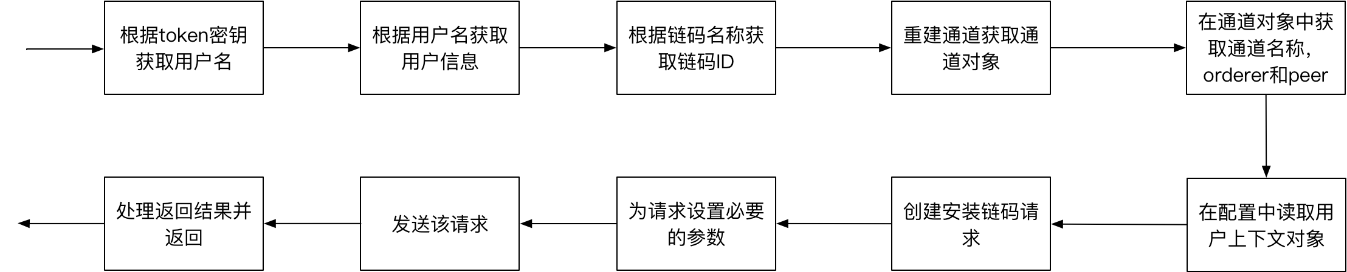
## 13.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 操作结果 | result | String | 安装成功则返回安装成功否则返回发生错误 |

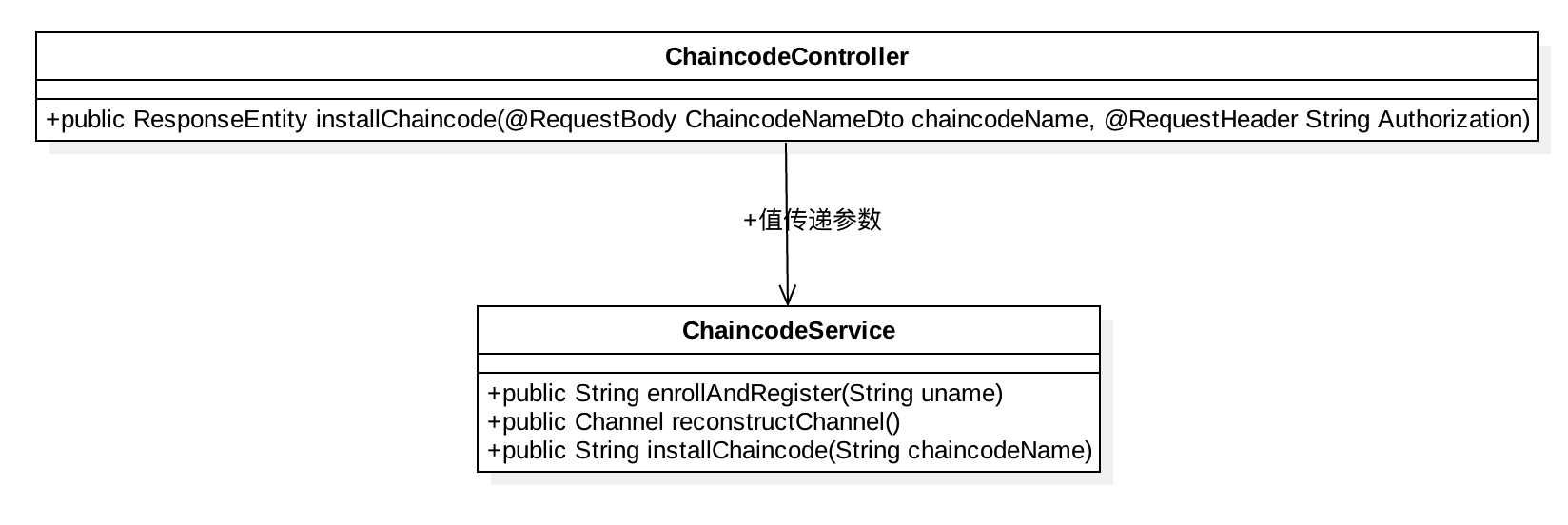
## 13.6算法

无具体算法，采用fabric 1.0 Java SDK

## 13.7流程逻辑



## 13.8接口



## 13.9存储分配

读写HyperledgerEnroll.properties文件

## 13.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 13.11限制条件

区块链正常运行，token秘钥正确，通道已创建

## 13.12测试计划

技术要求：需要进行单元测试

输入数据：使用curl工具进行调用 参数为："chaincodeName": "myChaincode"

预期结果：返回Chaincode installed successfully

进度安排：开发完成后即进行测试

## 13.13尚未解决的问题

无

# 14 API交互程序-实例化链码（3004）设计说明

## 14.1程序描述

区块链安装链码后需要实例化，为了方便对区块链网络进行管理，在API交互程序中书写实例化链码的接口

## 14.2功能

对区块链网络进行实例化链码操作

## 14.3性能

无

## 14.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| Token秘钥 | Authorization | String | POST请求header |
| 实例化方法 | function | String | POST请求参数 |
| 链码名称 | chaincodeName | String | POST请求参数 |
| 方法参数 | args | String[] | POST请求参数 |

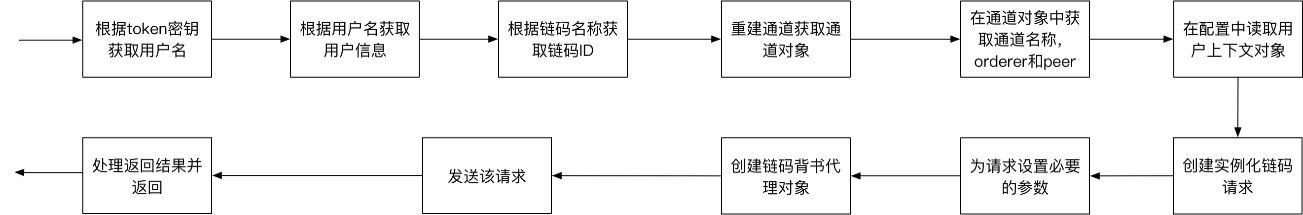
## 14.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 操作结果 | result | String | 实例化成功则返回实例化成功否则返回发生错误 |

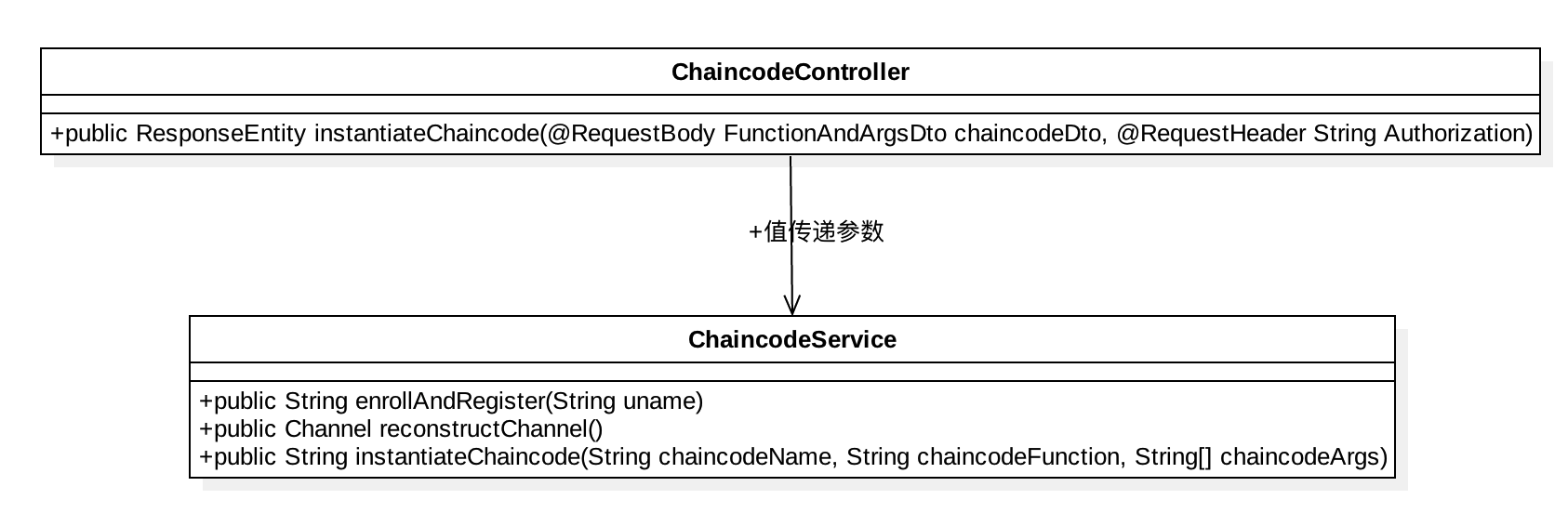
## 14.6算法

无具体算法，采用fabric 1.0 Java SDK

## 14.7流程逻辑



## 14.8接口



## 14.9存储分配

读写HyperledgerEnroll.properties文件

## 14.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 14.11限制条件

区块链正常运行，token秘钥正确，链码已安装

## 14.12测试计划

技术要求：需要进行单元测试

输入数据：使用curl工具进行调用，参数为

"args": [],

"chaincodeName": "myChaincode",

"function": "init"

预期结果：返回Chaincode instantiated Successfully

进度安排：开发完成后即进行测试

## 14.13尚未解决的问题

无

# 15 API交互程序-调用链码（3005）设计说明

## 15.1程序描述

区块链需要调用链码来进行保存、删除以及更新等操作，为了方便对区块链网络进行管理，在API交互程序中书协创建通道的接口

## 15.2功能

对区块链网络进行调用链码进行保存、删除以及更新等操作

## 15.3性能

无

## 15.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| Token秘钥 | Authorization | String | POST请求header |
| 调用的方法 | function | String | POST请求参数 |
| 链码名称 | chaincodeName | String | POST请求参数 |
| 方法参数 | args | String[] | POST请求参数 |

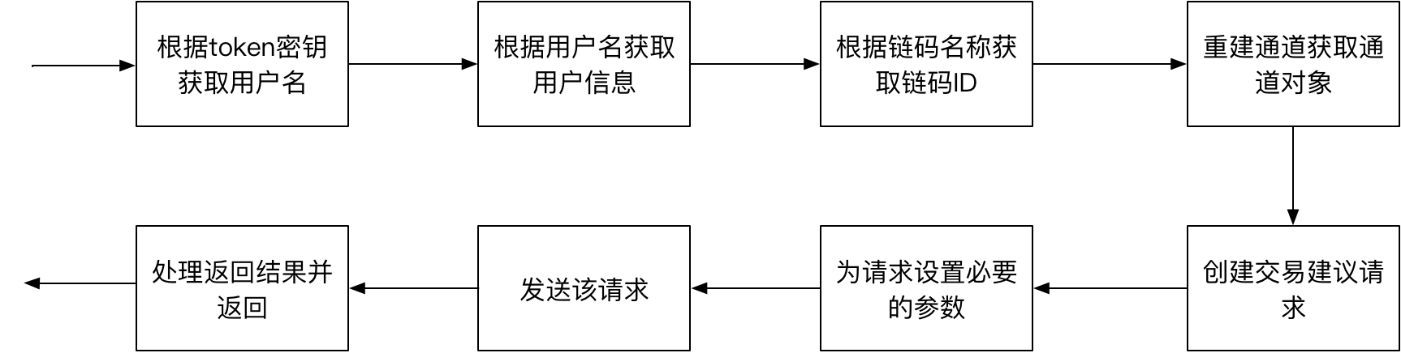
## 15.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 操作结果 | result | String | 调用成功则返回调用成功否则返回发生错误 |

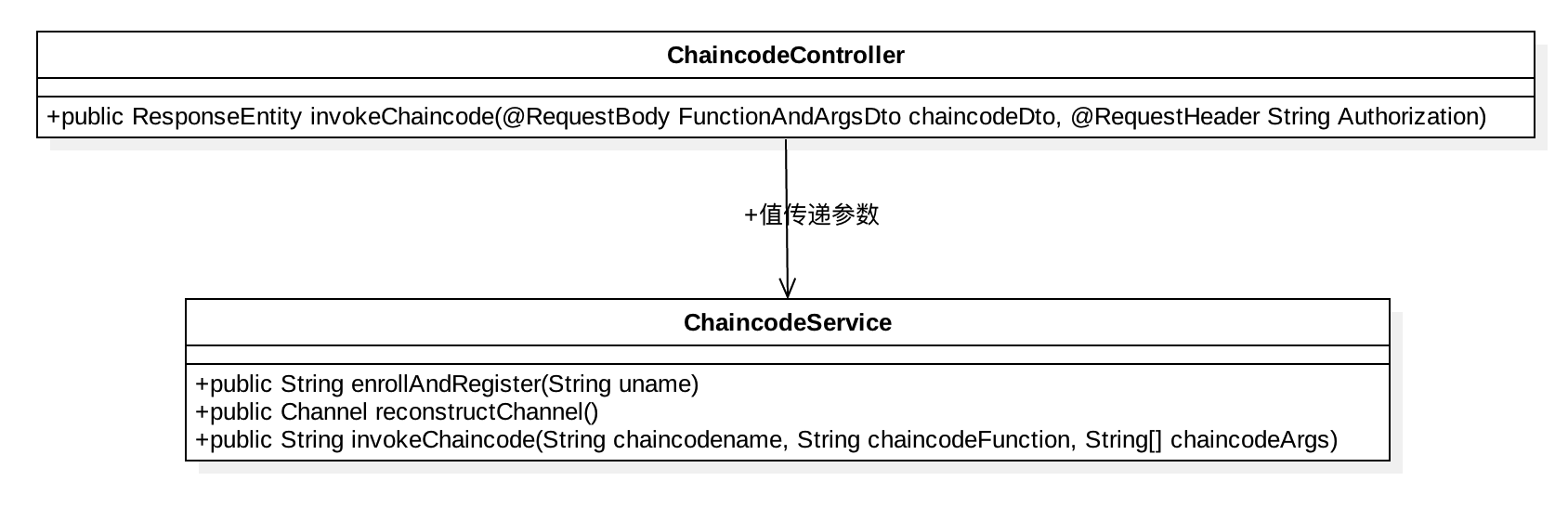
## 15.6算法

无具体算法，采用fabric 1.0 Java SDK

## 15.7流程逻辑



## 15.8接口



## 15.9存储分配

读写HyperledgerEnroll.properties文件

## 15.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 15.11限制条件

区块链正常运行，token秘钥正确

## 15.12测试计划

技术要求：需要进行单元测试

输入数据：使用curl工具进行调用参数为：

"args": [

"save","a","b","c","d","e","f","g","h"

],

"chaincodeName": "myChaincode",

"function": "invoke"

预期结果：返回Transaction invoked successfully

进度安排：开发完成后即进行测试

## 15.13尚未解决的问题

无

# 16 API交互程序-查询链码（3006）设计说明

## 16.1程序描述

区块链需要调用链码来进行查询，为了方便对区块链网络进行管理，在API交互程序中书协创建通道的接口

## 16.2功能

对区块链网络进行调用链码来进行查询操作

## 16.3性能

无

## 16.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| Token秘钥 | Authorization | String | GET请求header |
| 调用的方法 | function | String | GET请求参数 |
| 链码名称 | chaincodeName | String | GET请求参数 |
| 方法参数 | args | String | GET请求参数 |

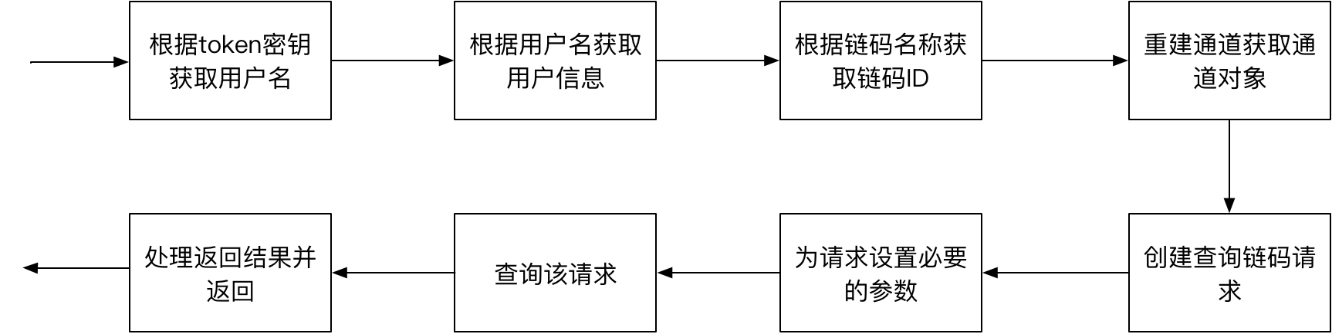
## 16.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 查询结果 | result | String | 查询成功则返回查询结果JSON字符串，否则返回发生错误 |

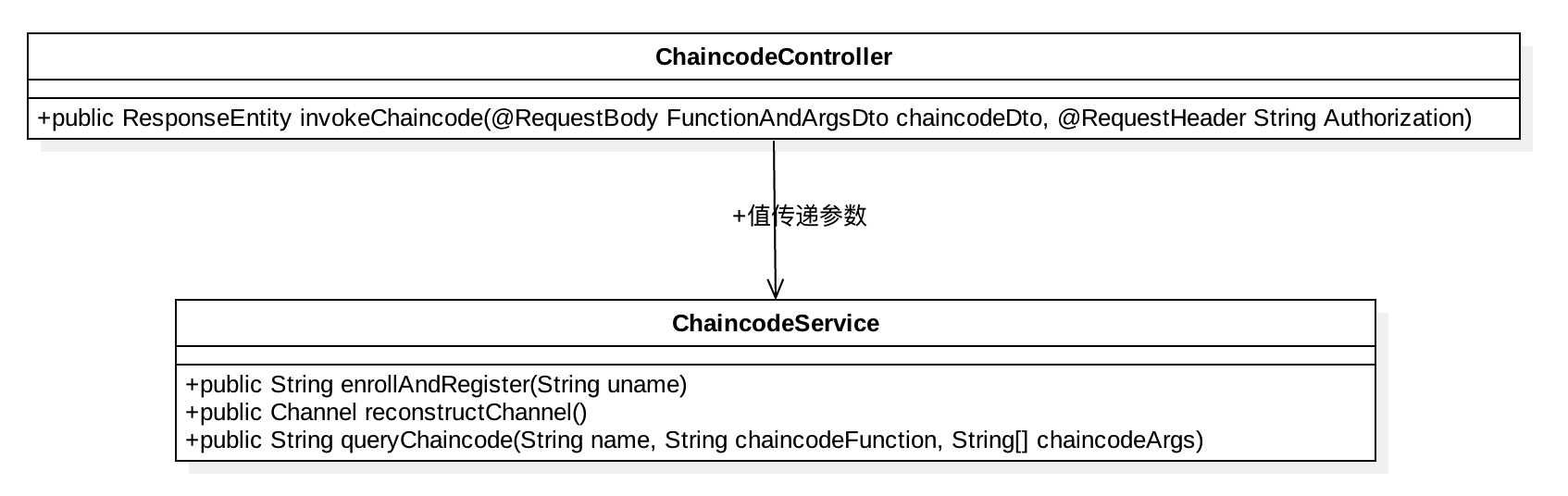
## 16.6算法

无具体算法，采用fabric 1.0 Java SDK

## 16.7流程逻辑



## 16.8接口



## 16.9存储分配

读写HyperledgerEnroll.properties文件

## 16.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 16.11限制条件

区块链正常运行，token秘钥正确，链码已实例化

## 16.12测试计划

技术要求：需要进行单元测试

输入数据：使用curl工具进行调用参数为：

ChaincodeName=myChaincode&function=invoke&args=query%2Ca

预期结果：返回id为a的信息

进度安排：开发完成后即进行测试

## 16.13尚未解决的问题

无

# 17 API交互程序-重建通道（3007）设计说明

## 17.1程序描述

区块链需要重建通道，为了方便对区块链网络进行管理，在API交互程序中书写重建通道的接口

## 17.2功能

对区块链网络进行重建通道的操作

## 17.3性能

无

## 17.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| Token秘钥 | Authorization | String | POST请求header |

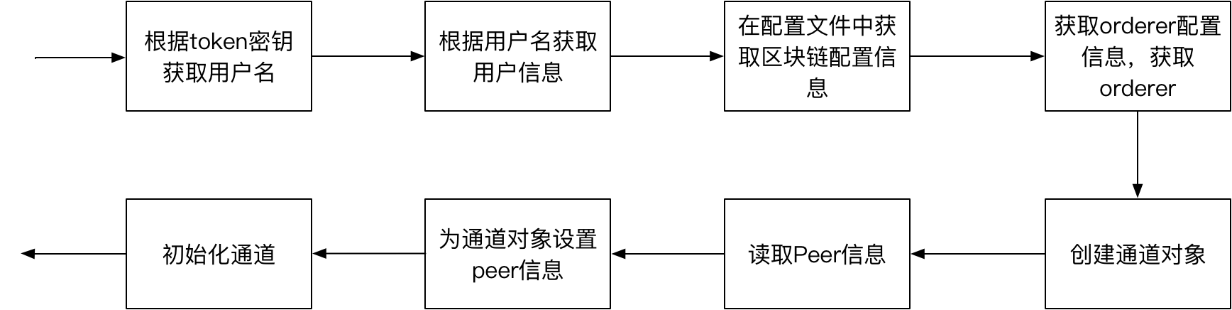
## 17.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 重建结果 | result | String | 重建成功则重建成功，否则返回发生错误 |

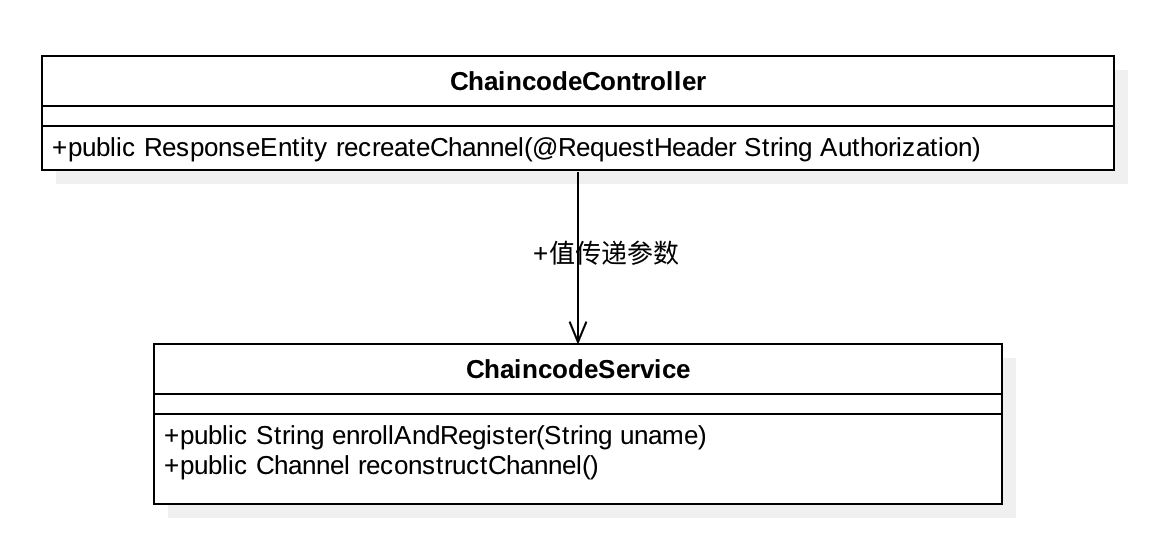
## 17.6算法

无具体算法，采用fabric 1.0 Java SDK

## 17.7流程逻辑



## 17.8接口



## 17.9存储分配

读写HyperledgerEnroll.properties文件

## 17.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 17.11限制条件

区块链正常运行，token秘钥正确，链码已实例化

## 17.12测试计划

技术要求：需要进行单元测试

输入数据：使用curl工具进行调用

预期结果：返回channel recreated successfully

进度安排：开发完成后即进行测试

## 17.13尚未解决的问题

无

# 18 API交互程序-升级链码（3008）设计说明

## 18.1程序描述

区块链需要升级链码，为了方便对区块链网络进行管理，在API交互程序中书写升级链码的接口

## 18.2功能

对区块链网络进行升级链码的操作

## 18.3性能

无

## 18.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| Token秘钥 | Authorization | String | POST请求header |
| 实例化方法 | function | String | POST请求参数 |
| 链码名称 | chaincodeName | String | POST请求参数 |
| 方法参数 | args | String[] | POST请求参数 |

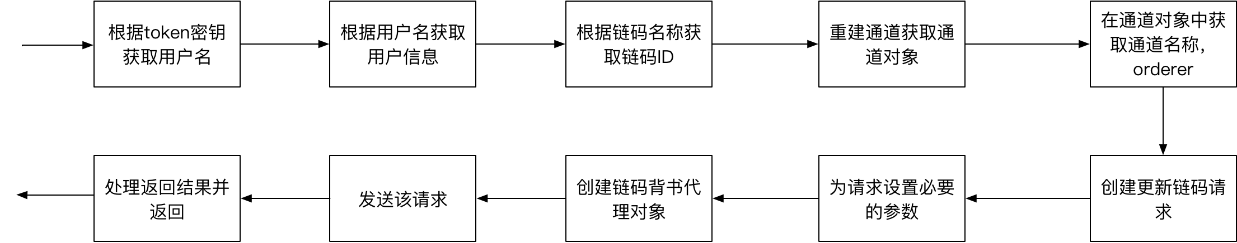
## 18.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 重建结果 | result | String | 升级成功则返回升级成功，否则返回发生错误 |

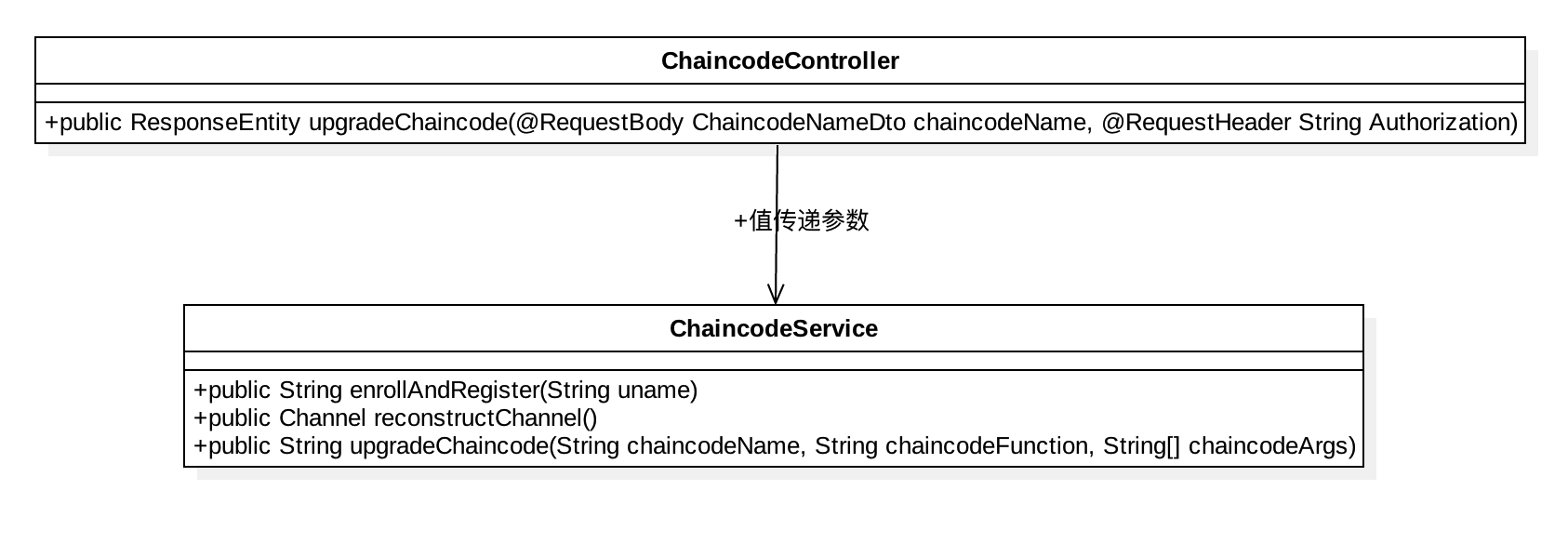
## 18.6算法

无具体算法，采用fabric 1.0 Java SDK

## 18.7流程逻辑



## 18.8接口



## 18.9存储分配

读写HyperledgerEnroll.properties文件

## 18.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 18.11限制条件

区块链正常运行，token秘钥正确，已创建通道

## 18.12测试计划

技术要求：需要进行单元测试

输入数据：使用curl工具进行调用，参数为：

"args": [

],

"chaincodeName": "myChaincode",

"function": "init"

预期结果：返回升级链码成功

进度安排：开发完成后即进行测试

## 18.13尚未解决的问题

无

# 19 API交互程序-查询区块信息（3009）设计说明

## 19.1程序描述

Web系统需要查询区块信息，在API交互程序中书写查询区块信息的接口

## 19.2功能

查询区块信息

## 19.3性能

无

## 19.4输入项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 输入方式 |
| Token秘钥 | Authorization | String | GET请求header |

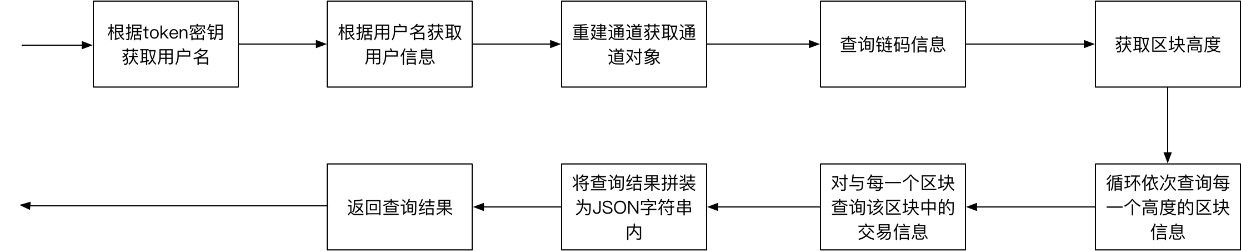
## 19.5输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据格式 | 备注 |
| 查询结果 | result | String | 查询成功则返回区块信息的JSON字符串，否则返回发生错误 |

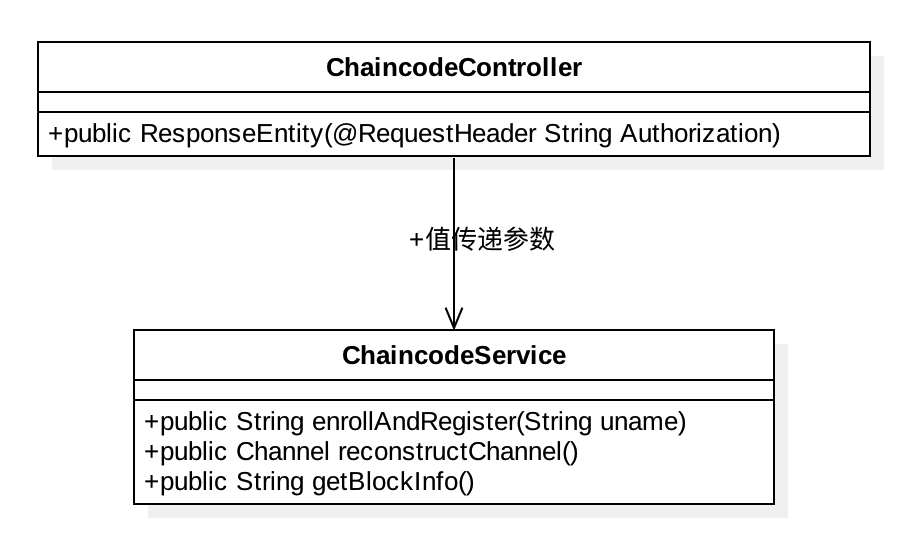
## 19.6算法

无具体算法，采用fabric 1.0 Java SDK

## 19.7流程逻辑



## 19.8接口



## 19.9存储分配

读写HyperledgerEnroll.properties文件

## 19.10注释设计

1. 在方法首部注释功能说明，需要的参数，创建日期和修改日期，作者等
2. 在各个处理流程开始和结束位置添加说明注释
3. 对需要用到的变量的用处，作用范围进行注释

## 19.11限制条件

区块链正常运行，token秘钥正确，链码已实例化

## 19.12测试计划

技术要求：需要进行单元测试

输入数据：使用curl工具进行调用

预期结果：返回区块信息JSON字符串

进度安排：开发完成后即进行测试

## 19.13尚未解决的问题

无

# 20底层区块链部署设计说明

## 20.1程序描述

底层区块链网络采用4 Peer + 1 Orderer + 2 CA + 4 couchDB 来搭建区块链整体网络,其中Orderer 节点单独在一台主机上，每个Peer节点都配有一个couchDB节点数据库，其中Peer1节点与Peer2节点组成组织1，Peer3与Peer4节点组成组织2，每个组织配备一个CA节点用于证书的发放。

## 20.2功能

启动Fabric多机部署。

## 20.3性能

无。

## 20.4输入项

Shell 命令： （1）./network up cdb

（2）docker exec –it cli bash

（3）bash scripts/script.sh mychannel

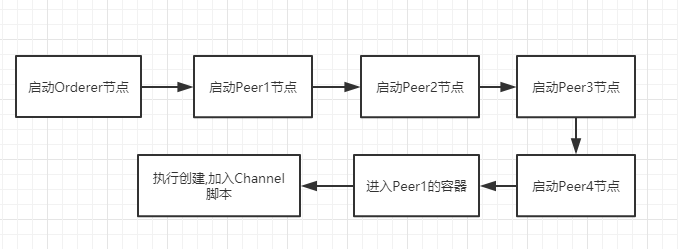
## 20.5输出项

启动日志

## 20.6算法

无具体算法，底层区块链环境采用的是Fabric 1.0

## 20.7流程逻辑



## 20.8接口

无接口设计

## 20.9存储分配

通过docker-compose的方式启动docker 容器，将链码文件映射到docker容器中。

## 20.10注释设计

在配置文件中都要相应的注释说明，包括配置参数，以及参数值的作用。

对启动脚本中的命令都进行了注释。

## 20.11限制条件

Docker 环境干净，之前的网络数据已经清理

## 20.12测试计划

技术要求：需要测试执行脚本

输入数据：使用命令和脚本

预期效果：节点容器创建成功

速度安排：启动成功后可进行测试

## 20.13尚未解决的问题

CouchDB数据的持久化存储