JPassword项目概要设计

# 总体设计

## 概述



主程序（客户端）：客户端采用Chrome内核，页面数据保存在LocalStorage方式，同步数据时调用ipfs接口保存数据。开发语言使用nodejs + vue方式。

浏览器插件：使用chrome plugin技术，基于主程序，依据chorme打包的要求打包到应用市场发布

数据同步：用不同的策略同步客户端与持久层的数据，使两者数据得以同步和一致。

支付接口：目前接入支付宝接口，将支付信息保存在区块链的交易信息中。

IPFS接口：与IPFS服务进行

## 模块功能及相互之前数据接口

本项目分为

1）客户端模块：安装在windows操作系统中。

2）插件端模块：在浏览器插件端使用的，及时记录

3）收费及支付模块：使用支付宝

# 业务流程设计

## 用户注册



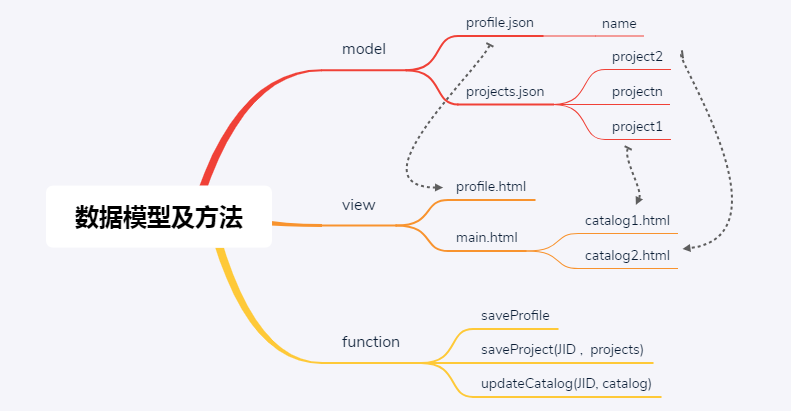
## 支付流程



## 数据加密方式

1. 用用户数据的使用公钥加密
2. 将数据转换为base64字节方式上传到 IPFS
3. 用户交费信息记录在区块链的交易memo中

# 业务数据模型



# 客户端主程序

## 程序结构：

Doucument :jpass项目相关文档存放目录。

Lib: 相关工具类工具文件包

Node\_modues:系统开发工具包

Public:公共静态资源目录（图片）

Src:项目开发资源包

Comment/lang：国际化文件包

Components:项目组件

Img:图片资源目录包

Views:项目页面包

项目核心文件：

App.vue:项目主入口文件

CreatePassword.js:密码服务文件

FileIndeDb.js: 数据库核心文件

Jcc\_ipfs.js: jcc钱包支付接口

Lock.js：项目锁定功能文件

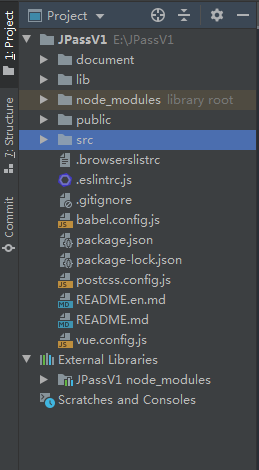
Router:系统路由核心文件

Package.json：定义了这个项目所需要的各种模块，以及项目的配置信息（比如名称、版本、许可证等元数据）

Package-lock:系统依赖包锁定文件。

Vue.config:跨域动态代理配置文件。

Readme.md:项目说明文件。



## 

## 接口部分设计：

##### Write(secret,data,filepath)

Secret:得到用户公钥与私钥，公钥用于计算数据签名，私钥用于加密数据内容。

Data:上传ipfs数据

FilePath：存储在ipfs中的文件路径

#### Read(secret,data,filepath)

Secret:得到用户私钥，私钥用于解密数据内容。

FilePath：存储在ipfs中的文件路径

Address：用户钱包地址

## 与jcc区块链的登录设计：

1. 用户注册时已经把区块链的用户keyStore 文件缓存到流量器中。
2. 用户登录时选择用户名也就是选择了用户的keyStore文件。
3. 输入密码，提交
4. 系统创建井创钱包对象wallet。
5. keyStore json对象解析出钱包地址address。
6. 将keyStore的json对象载入wallet 对象。
7. 通过wallet的getScretwithAddress 方法，以及password和address两个参数解析出用户的secret.

Secret 不为空，用户验证成功。

1. 把用户的相关信息载入到object 登录对象中，缓存到用户浏览器 localstorage和sessionstorage 中随后页面跳转到系统主页面。

## 数据同步

### 概念

机器码：本地生成的随机码

版本号：根据时间戳（需要联网取格林威制标准时间）生成的字符串

### 策略

一.版本号变动机制：

1.本地任何数据修改，版本号重新生成。

2.同步和初始化时，以版本号大的为准。

3.如有弹框弹出，以用户选择为准。

二.机器码变动机制（客户端与插件端）：

1.登录（系统自动检测数据）：

（1）本地，客户端都没有数据：初始化数据，本地自动生成机器码。

（2）本地有数据，ipfs 没数据。（机器码不变） -- 无动作

（3）本地与ipfs都有数据

判断机器码是否一致

a. 如果机器码一致，--无动作

b.如果机器码不一致，判断数据版本是否一致

数据版本一致：无动作。（出现情况：用户从ipfs 同步过数据，并且没做任何修改和同步操作）

数据版本不一致：判断本地版本数据是否大于ipfs版本数据

本地版本时间大于ipfs：不做处理。（出现情况：用户从ipfs 同步过数据，并且在本地修改了本地数据，并且数据没有上传）

本地版本时间小于ipfs：出现弹出框,让客户选择同步方式（出现情况：用户在其他机器修改数据，并且同步到ipfs。此时，ipfs 版本肯定大于本地版本数据，并且机器码是不相同的）；同步方式为： 1)本地覆盖到远端ipfs; 2)远端ipfs覆盖到本地服务器。

2. 手动同步：

判断数据版本号。

如果本地版本大于ipfs，本地数据覆盖到ipfs.(同步后，本地版本号，机器码均与ipfs相同)

如果本地版本小于ipfs, ipfs数据覆盖到本地，本地自己生成新机器码。（同步后，本地版本号与ipfs 相同，但是机器码与ipfs 不同）。

# Chrome插件版

1. 把相关代码及资源文件打包成插件支付的包
2. 监听用户的输入框，当有密码相关输入时，自动响应

## 记录网页密码设计：

Jpassword插件修改下载到本地 文件的js或者 html文件，加入一段代码，使其可以自动监听其输入密码的动作，从而保存到IPFS中，下次省去了用户记录代码的过程。

# 支付宝服务器端代码

## 程序框架

采用spring boot方式，Java语言开发，mysql数据库。

## 代码结构



# ipfs接口

## ****1.write 向IPFS写入数据****

**简要描述：**

* 向IPFS写入数据  
  验证用户钱包地址是否是JPass的VIP  
  1.根据辅助运营商钱包地址查找有关用户钱包地址的交易内容  
  2.如果存在有关交易，再判断交易信息中存储的使用时间，是否在使用期）

**请求URL：**

* ` <http://localhost:1337/api/v0/write>

**请求方式：**

* POST

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| data | 是 | string | 上传数据 |
| md5 | 是 | string | 上传数据md5 |
| filePath | 是 | string | 文件路径 |
| size | 是 | int | 文件大小 |
| name | 是 | string | 文件名称 |
| sign | 是 | string | 签名 sign的计算 sign(md5+size+filePath+timestamp,privatekey) |
| timestamp | 是 | int | 时间戳 |
| publickey | 是 | string | 公钥 |

**返回示例**

1. {
2. "result":[
3. {
4. "transaction":"QmaZTHxuAxaMupeTRCLwq6WY1b2MEdjzyHtrLp6UGXsmLC",
5. }
6. ],
7. "status":"success"//分为：success:成功； lackoil ：缺少燃料；error: 未知错误
8. }

## ****2.read 根据文件路径和钱包地址,读取IPFS数据****

**简要描述：**

* 根据文件路径和钱包地址,读取IPFS数据

**请求URL：**

* http://localhost:1337/api/v0/read

**请求方式：**

* GET,POST

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| filePath | 是 | string | 文件路径 |
| address | 是 | string | 钱包地址 |

**返回示例**

1. {
2. "status":"success",
3. "message":"",//出错才有
4. "file":{
5. "data":"base64",
6. "md5":"123231",
7. "size":6,
8. "timestamp":1586399040066,
9. "path":{
10. "root":"/",
11. "dir":"/二级目录",
12. "base":"asdasd.txt",
13. "ext":".txt",
14. "name":"asdasd",
15. "filePath":"/二级目录/asdasd.txt"
16. }
17. }
18. }

**返回参数说明**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| data | 是 | string | 上传数据 |
| md5 | 是 | string | 上传数据md5 |
| size | 是 | int | 文件大小 |
| timestamp | 是 | int | 时间戳 |
| filePath | 是 | int | 文件路径 |

**备注**

* GET请求直接返回data的内容

## ****3.remove 根据文件路径和钱包地址,删除IPFS数据****

**简要描述：**

* 根据文件路径和钱包地址,删除IPFS数据

**请求URL：**

* ` <http://localhost:1337/api/v0/remove>

**请求方式：**

* POST

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| filePath | 是 | string | 文件路径 |
| address | 是 | string | 钱包地址 |

**返回示例**

1. {
2. "status":"success",//分为：success:成功； lackoil ：缺少燃料；error: 未知错误
3. "message":""//出错才有
4. }

## ****4.list 根据钱包地址和路径查询IPFS数据****

**简要描述：**

* 根据钱包地址和路径查询IPFS数据

**请求URL：**

* http://localhost:1337/api/v0/list

**请求方式：**

* POST

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| path | 否 | string | 目录路径,默认根目录 |
| address | 是 | string | 钱包地址 |

**返回示例**

1. {
2. "status":"success",
3. "message":"",//出错才有
4. "files":[
5. {
6. "path":{
7. "root":"/",
8. "dir":"/",
9. "base":"ZS",
10. "ext":"",
11. "name":"ZS",
12. "filePath":"/ZS"
13. },
14. "size":268,
15. "isDir":false
16. },
17. {
18. "path":{
19. "root":"/",
20. "dir":"/",
21. "base":"ZS1",
22. "ext":"",
23. "name":"ZS1",
24. "filePath":"/ZS1"
25. },
26. "size":270,
27. "isDir":false
28. },
29. {
30. "path":{
31. "root":"/",
32. "dir":"/",
33. "base":"二级目录",
34. "ext":"",
35. "name":"二级目录",
36. "filePath":"/二级目录/"
37. },
38. "size":0,
39. "isDir":true
40. }
41. ]
42. }

## ****5.charge 转账,标记某个钱包地址为JPass的VIP****

**简要描述：**

* 1.根据交易hash找到用户向运营商转账的交易详情
* 2.根据详情得到(1)用户钱包地址（2）运营商钱包地址（3）交易时间（4）交易金额
* 3.判断运营商钱包地址与金额是否符合要求
* 4.用运营商钱包向辅助运营商地址转账一定数量
* 5.交易备注里要存入（1）使用时间信息（交易时间+1年）

**请求URL：**

* http://localhost:1337/api/v0/charge

**请求方式：**

* POST

**参数：**

| **参数名** | **必选** | **类型** | **说明** |
| --- | --- | --- | --- |
| hash | 是 | string | 交易hash |

**返回示例**

1. {
2. status:'success'//失败:error
3. }

# 附：代码仓库及文档

JPassword 客户端程序：https://gitee.com/ahzs\_1/JPassV1.git

Jpassword 插件端程序：https://github.com/zCaviare/jpassword\_extension.git

JPassword 支付宝端程序：https://github.com/zCaviare/JpassPay.git

IPFS接口服务:https://github.com/JCCDex/jcc\_ipfs.git

----------------------------------------------

IPFS接口整理文档：<https://www.showdoc.cc/ipfs?page_id=4160536662548180>