

起点Origin- 开发文档

基于FISCO BCOS的“起点Origin”族谱平台开发文档

业务背景

业务简介

中国的族谱历史悠久，其起源可以追溯到商周时期。许多大家族的族谱撰写流传达数十年甚至上千年不绝，比如“孔子族谱”。修订族谱，寻根问祖几乎是每个有条件的家庭都会设想甚至投入的事情，但就目前编撰最久的孔子族谱而言，它编撰长达2500多年，但历史可查的仅仅500多年，族谱损失严重，且早期由于嫡长子编撰法使得族员遗漏颇多，损害了它的历史价值。

祭祀祖先，学习族谱中的各世代家族成员姓名，大事记，居住地，产业迭代，祖传家训等对于一个家族自我认同和凝聚力的形成有很大的作用；对历史研究也充满社会意义；对增强家族乃至国家的文化自信颇具意义。

那么编撰一份好的族谱就显得至关重要。

业务痛点

1. 保存困难。传承修订2000多年的孔子族谱，在世的也仅仅500年，这都是吉尼斯世界纪录。
2. 族谱伪造。历代不乏名人伪造祖先以求体面，普通人附会望族求得利益，比如唐朝皇族李式自称是老子的后代，寒族出身的唐朝宰相李义府，自称自己出自赵郡李氏。
3. 粗制滥造。许多当代族谱并没有形成良好的记录规范和记录创新，大量抄袭仿照。
4. 编撰成本高。由于战乱，流失，家族衰败，家族迁徙等原因，族谱传承终有断代。家族中的家庭，一些可能重视修族谱，制作优良；一些可能不重视，制作简陋，甚至不修订了，造成族谱缺失。

Fisco Bcos的优势

1. 数据化的保存手段。由于存储技术的迭代更新，有望长期甚至永久保存记录，解决纸质版族谱易丢失以损害的困难。
2. 可一定程度解决族谱伪造。可以接入公安系统对上链的族谱信息在公安内部系统进行非中心化核验，在区块链上进行中心化的存储；同时可接入社会舆情投票系统
3. 粗制滥造。利用互联网化的手段，可对区块链族谱中的一些字段，或其他字段进行补充修订。
4. 降低家族编撰难度。

- a. 普通家庭成员可利用“区块链客户系统”进行族员录入，降低纸质编撰的难度。
- b. 公安节点可进行简易版的数据录入，以避免家族的重视造成的族员数据缺失。

社会价值

1. 了解家族来源。传递家族的兴衰历史，追求家族的历史足迹，了解家族的奋斗历程。
2. 研究历史的变迁。河南省图书馆的鲁山族谱，就详细记载了康熙七年一支军队出征的情况，包括将官的名录官职、出征前的会议记录、准备工作，甚至具体到每日行军路线、路程、士兵艰苦的生存环境等等，非常有历史研究价值。
3. 教育后人意义。有的族谱会记录家族的家训，教育后人要遵守家法国法、和睦宗族乡里等。

文案术语

1. 起点Origin。它是作者为这个系统取的名字，是一个普通家庭进行族谱存证的平台，以FISCO BCOS作为底层区块链平台。

技术方案设计

业务需求分析

1. 调用者本人族谱信息的录入：addPerson(int version, string name, string motto, address father, address mother)

功能：用户将自己的个人基本信息存入平台(族谱合约SaveRepository)

场景：用户首次想将自己的信息存入“起点Origin”平台时调用

a. 族员防伪地录入族谱数据

- i. 录入父母数据，会经过父母互确认机制，然后在父母列表中分别追加。即如果父母同意子女列表中有对应的子女，那么会在父母的children列表中加入，同时儿子的父母数据会上链成功；否则则虽然录入，但显示父母数据未经确认，直到经过父母或公安节点进行确认后才标记为确认。displayWaitingForConfirmParentList
- ii. 录入子女数据，会经过子女互确认机制，然后在子女列表中追加。即如果子女同意父母列表中有对应的父母，那么会在子女的father 或 mother列表中加入，同时父母的子女数据才会上链成功；否则则虽然录入，但显示子女数据未经确认，直到经过子女或公安节点进行确认后才标记为确认。displayWaitingForConfirmChildrenList

b. 族员可更改部分非关键字段

- i. 每个族员，默认可更改3代内的CURD表中的非关键数据，但不可更改关键数据。部分解决族谱被后代过分修饰，甚至篡改的危险。

2. 调用者添加亲属关系：addRelative(string relation, address id)

功能：用户将自己的个人额外亲属信息存入平台(族谱合约SaveRepository)

场景：当用户已经存入基本信息后(在已经调用addPerson后)，还想添加 伴侣couple/子女children时调用

```
调用者id: 0x9b5be6d3aab47842a1c6f59957335bd4b2114275
relation: "fation"
id: 0x1efcab424d86784254d153a0c298b30200a4d689
```

3. 查看调用者待确认的亲属关系：selectWaitingList()

功能：查看用户id待确认的 亲属关系列表

优势：亲属之间的互相确认机制。

4. 确认调用者与id之间的亲属关系：voteRelative(string relation, address id)

功能：同意在族谱上(族谱合约SaveRepository) 与亲属(id) 正式建立 一个叫relation的关系

场景：用户想确认自己与id 是一种叫做 relation 的亲属关系时调用

优势：亲属之间的互相确认机制

5. 查看n代系谱图：selectNLineage(uint n)

功能：查看(族谱合约SaveRepository) 合约调用者tx.origin的 n 代族谱信息

场景：用户添加个人信息到(族谱合约SaveRepository)上后，总想看看 这几代的状况吧，这时调用此合约

```
孙一代 - ***
孙二代 - ***
孙三代 - ***
孙四代 - ***
孙五代
```

3. 查看第n带先辈：selectNPerson(uint n)

功能：查看(族谱合约SaveRepository) 合约调用者tx.origin的 n 代族谱信息

场景：用户添加个人信息到(族谱合约SaveRepository)上后，总想看看 祖宗的基本信息吧，

使用：selectNperson(1):查看父亲的基本信息

selectNperson(2): 查看爷爷的基本信息

selectNperson(3): 查看祖祖的基本信息

```
version:1
id:0x9b5be6d3aab47842a1c6f59957335bd4b2114275
name:"孙***"
motto:"我们从**迁徙至***，这辈子主要在经营***，家业不易。后代当更勤勉治学"
```

数据结构

```
1 contract Person{
2     int _version;
3     address _id;
4     string _name;
5     string _motto;
6     address _father;
7     address _mother;
8     address _couple;
9     address _children;
10 }
```

参与者与场景

参与者

参与区块链底层管理的节点有：

1. 公安部门（可不参加）。如果法律允许，可对族员相关信息进行录入。
2. 族谱管理节点（必须参加）。提供族员信息录入的平台，平台只需部署1次，即可使用。这个节点对普通用户来说，就是这个平台系统。
3. 其他节点。协助认证某些族员的信息录入。

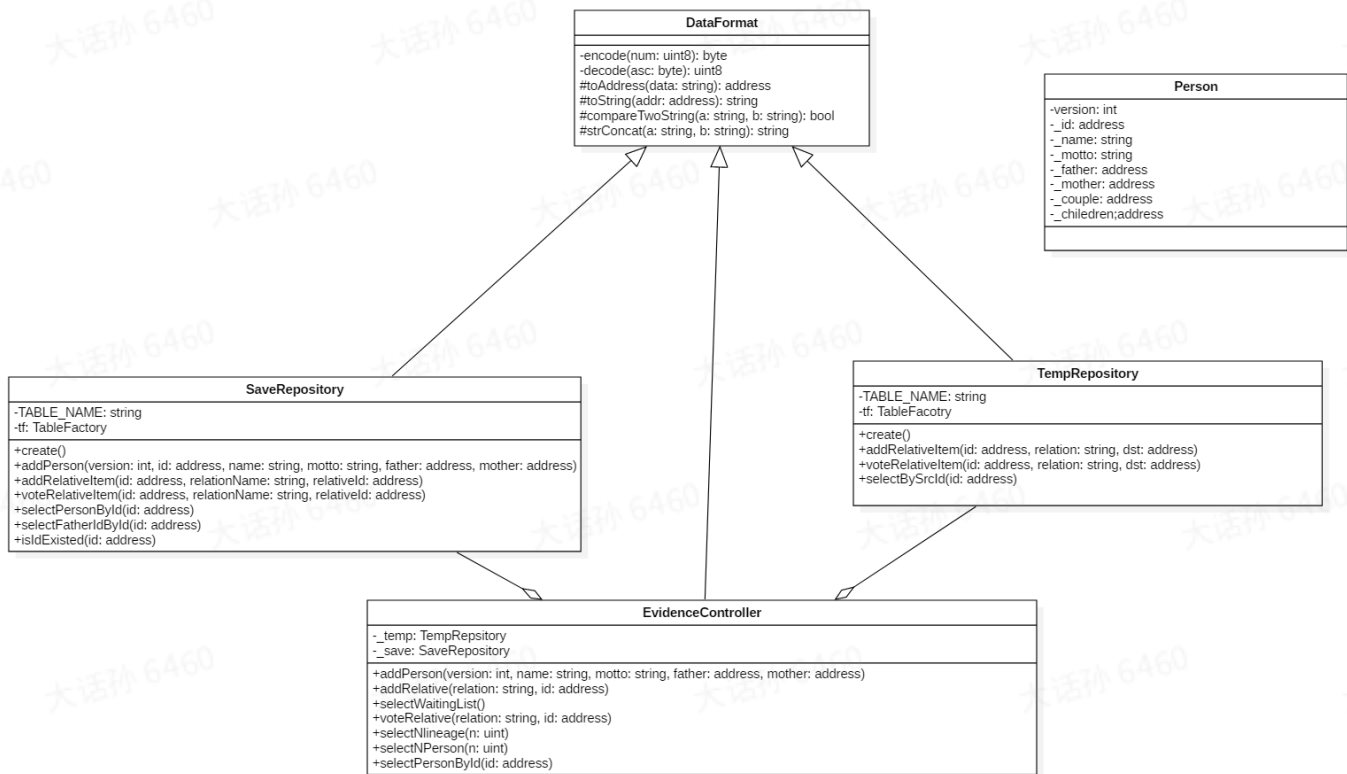
场景

主要提供族谱记录服务。对有需要将录入家族成员姓名，大事记，居住地，产业迭代，祖传家训等信息，并期待长久保存的家族提供族谱记录服务

概要设计方案

UML类图

为了对设计方案展示的更全面，帮助读者理解。特意用智能合约的类图表示各个contract之间的关系



智能合约设计与说明

1. 族谱合约SaveRepository：本系统的主要族谱信息是通过SaveRepository合约操作一个名叫“t_SaveRepository”的Table完成的，它相当于Repository，数据层。
2. 族谱待确认缓存合约TempRepository：本系统用户待确认的的亲属关系，通过这个合约进行存储，操控着一个名叫“t_TempRepository”的Table，也相当于Repository数据层
3. 控制合约EvidenceController：本系统的用户接口，普通用户通过调用这里的接口与平台进行交互。
4. 数据格式合约DataFormat：本系统通用的数据格式转换合约，起到辅助作用，不具有系统实际功能。
5. 人物合约Person：本系统通过这个合约定义任务的基本结构。

使用说明

实际场景介绍

上手指南和使用手册

部署指南

- console：形如下图可以部署

```
[group:1]> deploy TempRepository
contract address: 0xaa21f4f4b4b67fa6dc4f32456f77a2a71fd8cc90

[group:1]> deploy SaveRepository
contract address: 0x7cd12c3c244eadee5ed2f6fab4671607b4bbf360

[group:1]> deploy EvidenceController "0xaa21f4f4b4b67fa6dc4f32456f77a2a71fd8cc90" "0x7cd12c3c244eadee5ed2f6fab4671607b4bbf360"
contract address: 0x7fbfc0f21cf922643dcf2764d53c9c1f8acee79c
```

- WeBASE: 依次TempRepository、SaveRepository,EvidenceController进行部署，以下是EvidenceController部署图。

选择用户

用户:

sunyd

参数:

temp	<div>jc1b328cf4ab8b8e43</div>
save	<div>0xa45cac032df9d32fa!</div>

! 如果参数类型是数组，请用逗号分隔，不需要加上引号，例如：arry1,arry2。string等其他类型也不用加上引号。

取消

确定

调用指南

因为WeBase调用简洁，以下采用WeBASE 调用。

sunyd 账户地址（儿子sunyd）

0x9b5be6d3aab47842a1c6f59957335bd4b2114275

console-pri 账户地址（父亲console-pri）

0x16c9f89c71a17043a8d63af85a342aa7e8d1a595

1. 用户 sunyd 调用 addPerson(), 将自己的基本信息上链

addPerson(1, "qa", "motto str", ...)

注意：1个用户，只能执行1次addPerson(), 一旦执行用户自己的信息就已经上链了。

以后只能够执行addRelative()来更改亲属关系，默认基本信息（id,name）不可以更改，但father,mother,couple,children都可以更新。


```
selectWaitingList()
```

[illegible]

```
voteRelative("father", 0x16c9f89c71a17043a8d63af85a342aa7e8d1a595)
```

```
voteRelative("father", 0x16c9f89c71a17043a8d63af85a342aa7e8d1a595)
```


还原

还原