**課堂大作業：應用WeIdentity實現一個‘金融貸款購車案例’**

* **WeIdentity大作業（金融貸款憑證----------實現一個‘金融貸款購車案例’）：**

**作業場景描述：**一家車行，在爲客戶辦理購車手續時需要驗證客戶的貸款是否已發放，確定客戶貸款信息的真實性，當前存在的問題是，對于車行而言，缺乏高效即時驗證貸款及客戶信息真實性的手段，對于客戶而言，可以避免在貸款發放過程中二頭跑。基于WeID，用戶身份同電子身份ID一一對應，真實金融貸款信息電子化，幷一同保存在區塊鏈上，保證身份信息真實不可篡改，驗證身份信息快速高效。



**-------客戶進行WeIdentity DID注册**

1.客戶進入用戶代理公司的客戶信息（KYC）注册頁面，提交個人信息

2.用戶代理公司<亦可以是車行>審核申請，成功後授予該客戶Weidentity DID

**------客戶向金融公司或銀行申請購車貸款憑證（Credential）**

1.客戶向金融公司或銀行提交購車貸款憑證（Credential）申請

2. 金融公司或銀行審核申請，成功後授予該客戶購車貸款憑證Credential上鏈

**------客戶提交貸款憑證（Credential），用于驗證**

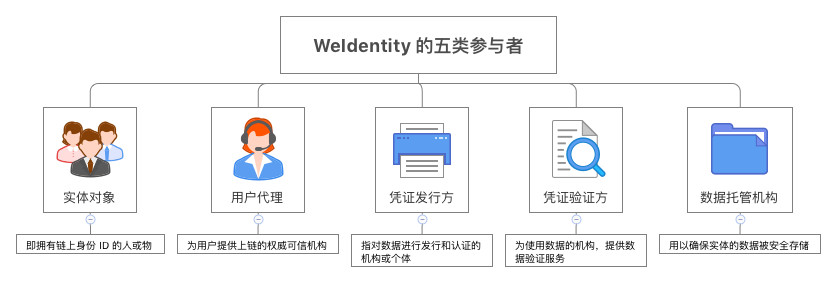
1.客戶向車行提交自己的貸款憑證Credential，車行鏈上驗證真實性後交付汽車

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Final Project：**實現一個‘金融貸款購車案例’** | | | | |
| **No.編號** | **Name名稱** | **Role角色** | **Resp說明** | **現場講解** |
| 1 | Davis | 小組長 | 案例描述及場景說明；角色分配；基本流程測試與設計； | 一至四（概述） |
| 2 | Matthew | 副小組長/金融公司或銀行  (Issuer) | 基本流程測試及跑通；金融公司或銀行(Issuer)方編碼； | 五（概述、金融公司或銀行(Issuer)方） |
| 3 | Nelson | 組員/金融公司或銀行  (Issuer) | 金融公司或銀行(Issuer)方編碼；配合基本流程測試及跑通； | 五（概述、金融公司或銀行(Issuer)方） |
| 4 | Tao | 組員/客戶(User) | 組員/客戶(User) 方編碼；配合基本流程測試及跑通； | 五（概述、客戶(User)） |
| 5 | Henry | 組員/車行公司(Verifier) | 車行公司(Verifier)方編碼；配合基本流程測試及跑通； | 五（概述、車行公司(Verifier)） |
| **備注**：請大家按如下實驗作业練習過程步驟（一至四步驟，參考“3、WeIdentity课堂练习.pptx”及weid-sample及HTTP-Service構建配合完成大作業！**最後我組決定選用HTTP-Service+PostMan方式實現案例。**） | | | | |

|  |  |  |
| --- | --- | --- |
| **WeIdentity方案** | **VS** | **傳統方案** |
| 電子化貸款憑證成本低，貸款證明文件轉為  Credential，可信存儲，易於管理，可以即時查詢及驗證，客戶不需要二頭跑； | **成本與便利性** | 紙質貸款憑證文件管理成本高，易丟失，不易更新，客戶需要二頭跑； |
| 鏈上客戶身份與現實身份一一對應；密碼學確保真實性和有效性，區塊鏈記錄確保不可篡改性； | **真實性** | 紙質貸款文件有造假可能性，需銀行-車行互相驗真周期長、效率低； |
| 全流程無第三方機構參與，金融公司或銀行可以選擇性向車行公司披露客戶信息；用戶授權後，車行公司方能獲取貸款批核信息，隱私保護性強； | **隱私保護** | 金融公司或銀行需要給車行披露客戶資料，資料傳遞的中間過程可能導致用戶隱私洩漏問題； |
| 方案可擴展性強，增加金融公司或銀行、車行數量，工作量不會相應增加，甚至可以聯合貸款 | **可擴展性** | 方案可擴展性差，每增加一個金融公司或銀行、車行公司，工作量會線性增加 |

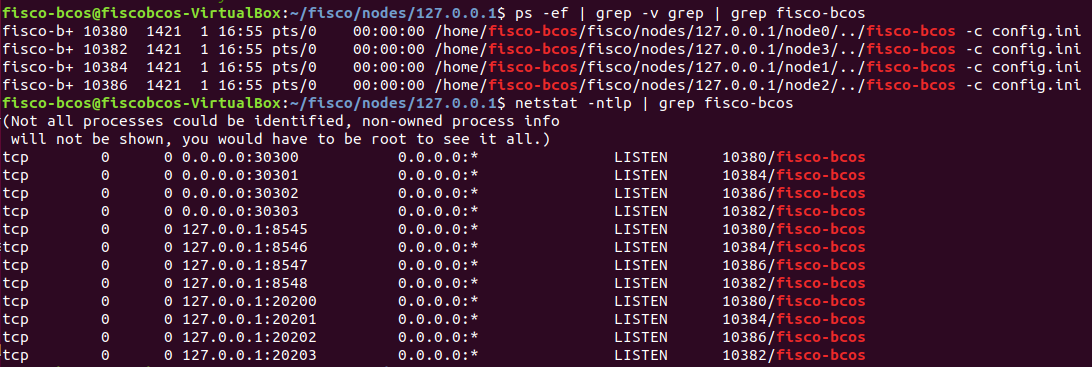
**參考：**以下WeIdentity的五類參與者角色分工如下------------------

實體對象：客戶；用戶代理：用戶代理服務公司<亦可以是車行>；憑證發行方：金融公司或銀行；憑證驗證方：車行公司；數據托管機構（VM模擬器）；

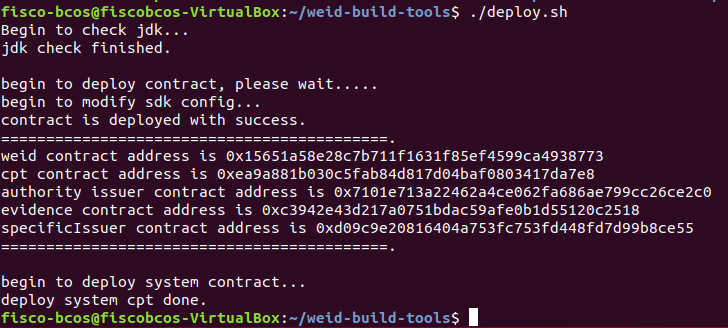
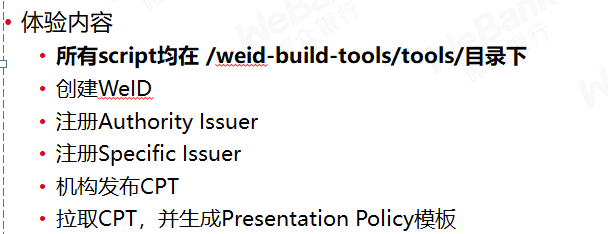


**其它：實驗大作业過程說明如下：**

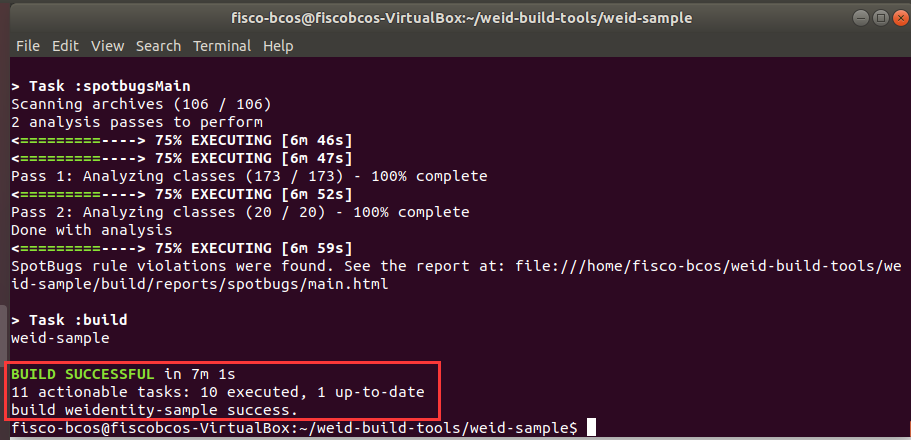
**一、啟動鏈（節點）-------檢查是否正常啟動4個節點及節點使用之端口情况**



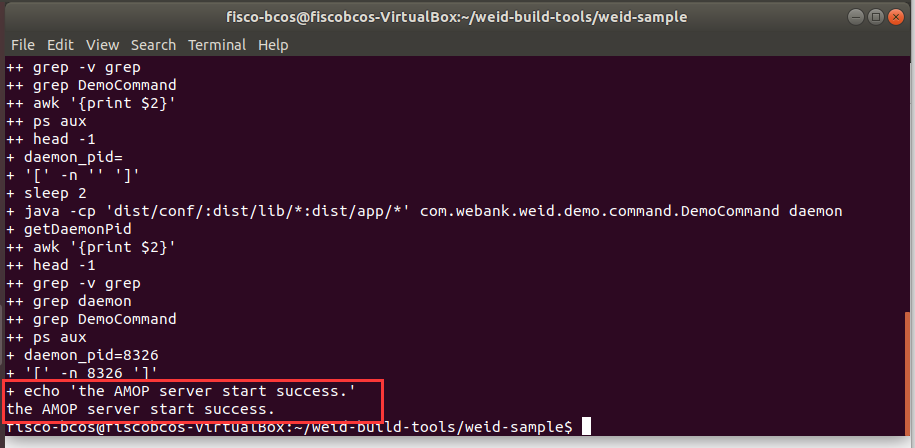
**二、部署WeID----運行脚本deploy.sh進行WeID合約的發布，完成Build-tools布署，通過tools快速體驗WeID的核心功能，如下：**

 <https://weidentity.readthedocs.io/zh_CN/latest/docs/weidentity-quick-tools.html>

**三、體驗weid-sample----Step1.運行脚本build.sh編譯weid-sample成功如下（必須安裝Gradle5.2.1）**



**Step2.啓動AMOP服務：#./command.sh daemon**



**Step3.** **修改User Agent配置：**

**#vi下述三個文件的節點機構名爲organizationA，節點IP用127.0.0.1:20200**

**/weid-sample/src/main/resources/weidentity.properties**

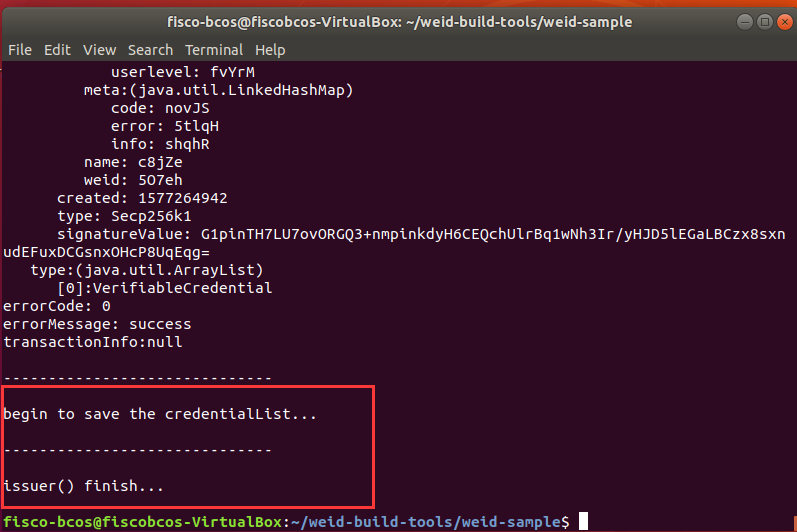
**/weid-sample/build/resources/main/weidentity.properties**

**/weid-sample/dist/conf/weidentity.properties**

**#/weid-build-tools/dist/conf/weidentity.properties 之下的節點機構名爲organizationB，節點IP用127.0.0.1:20201**

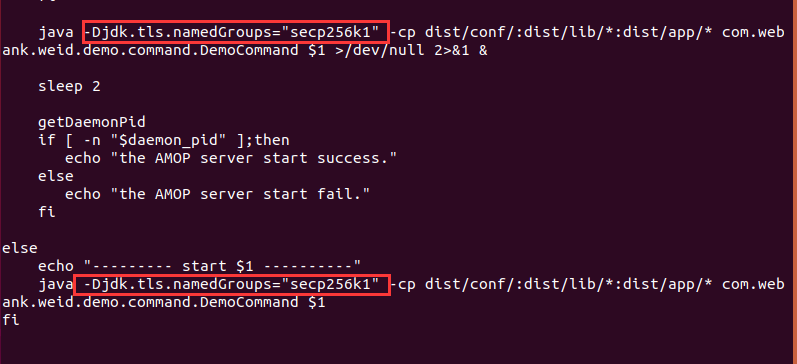
**Step4.執行情况如下：**

**1、Issuer視角如下：**

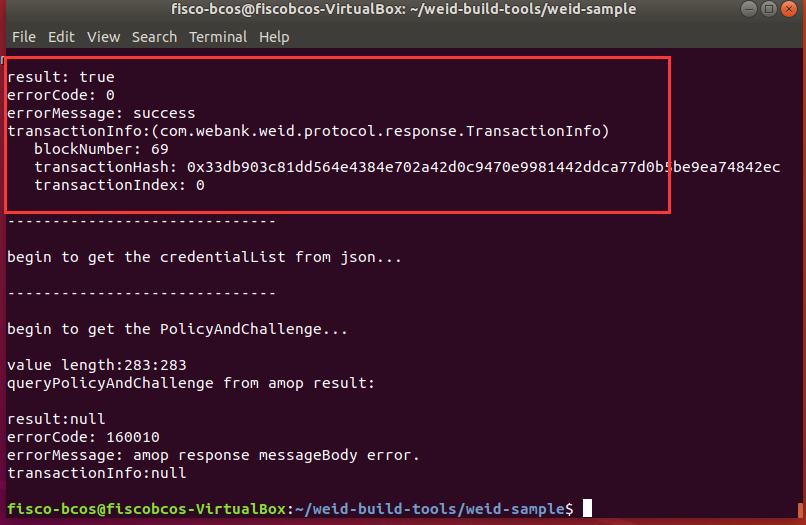


創建Credentiallist成功！

**另外注意：**需要修改/weid-sample/.commond.sh，找到如下內容，並添加-Djdk.tls.namedGroups="secp256k1"，這是因為JDK的版本太高引起的兼容性問題。

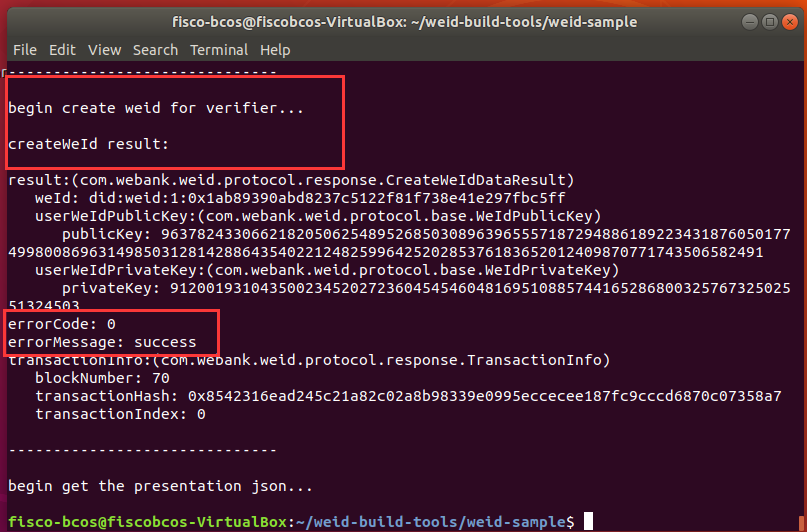


**2、User Agent視角如下：**



創建DID成功！

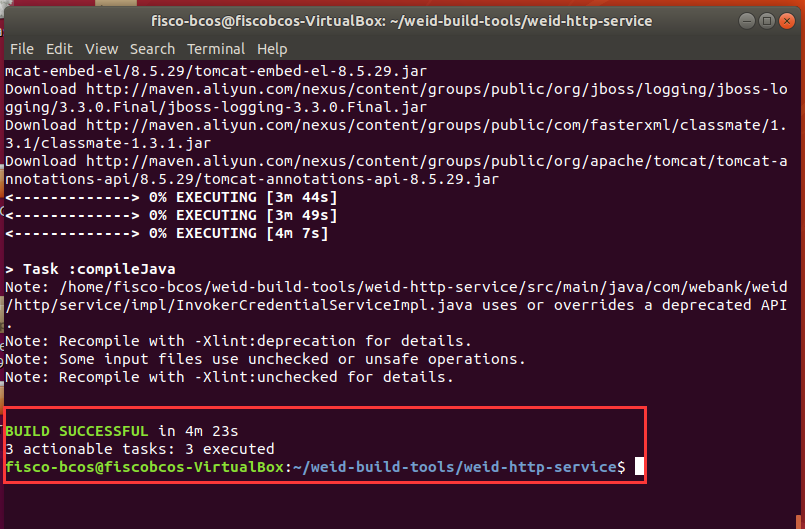
**3、Verifier視角如下：**



Verifier成功！

**四、體驗HTTP-Service-----****Step1.運行#gradle build -x test編譯weid-http-service成功如下**

**（必須安裝Gradle5.2.1）**



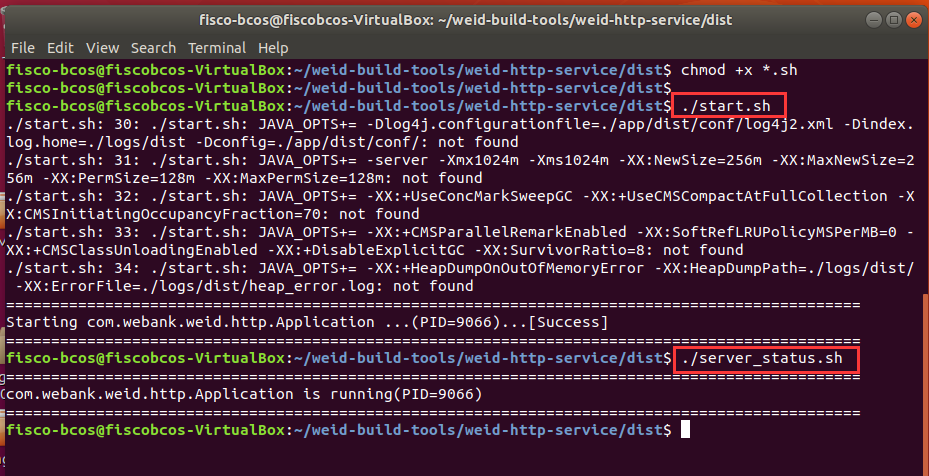
**Step2.** **修改配置文件：**

1.确认 WeIdentity 合约已部署;

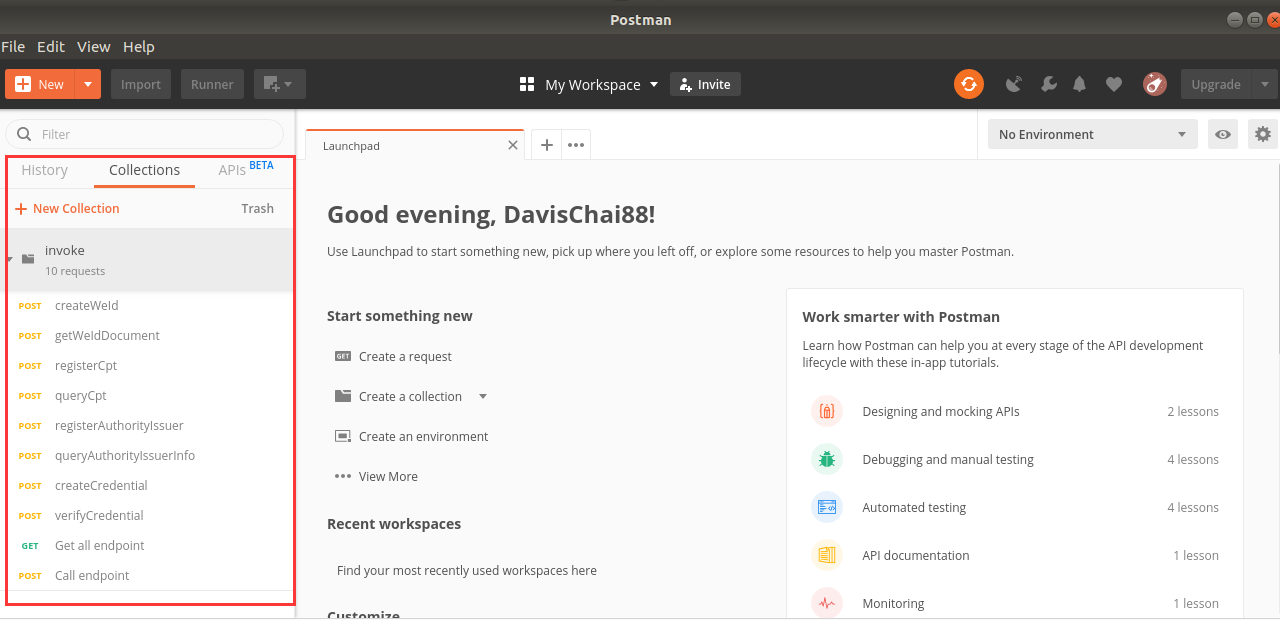
2.修改合约地址（部署工具或者源码部署）及区块链节点信息修改；

參考：<https://weidentity.readthedocs.io/zh_CN/latest/docs/weidentity-rest-deploy.html>

**Step3.執行情况如下：啟動HTTP-Service及觀察應用狀態**



**Step4.** **安裝Postman及Invoke測試**



然後按Invoke裏面不同的方法做測試即可。

**五、（第1組）選用HTTP-Service+Postman方式構建實現大作業------一個‘金融貸款購車案例’，具體過程如下：**



**-------客戶進行WeIdentity DID注册**

~~1.客戶進入用戶代理公司的客戶信息（KYC）注册頁面，提交個人信息~~

**客戶進入代理公司無需提供個人信息，保證客戶個人私隱（代理公司只需認錢不需認人）**

~~2.用戶代理公司<亦可以是車行>審核申請，成功後授予該客戶Weidentity DID~~

~~同上~~

**------- STEP1. 用戶代理為客戶註冊購車意向合同聲明**

1. 為註冊代理公司申請DID並註冊成為Issuer

2. 由代理公司往鏈上發布購買意向聲明**（客戶無需向代理服務公司提供任何個人資料/或僅提供基本資料，以保障個人隱私）**

**------STEP2. 客戶向金融公司或銀行申請購車貸款憑證（Credential）**

1. 客戶向金融公司或銀行提交購車貸款憑證（Credential）申請

2. 金融公司或銀行審核申請，成功後授予該客戶購車貸款憑證Credential上鏈（**STEP3**）

**------STEP4客戶提交貸款憑證（Credential），用于驗證**

1.客戶向車行提交自己的貸款憑證Credential，車行驗證真實性後交付汽車

**Prepare01. 註冊業務實體**

* **技術實現**

- 通过WeIdService的createWeId()生成一个WeIdentity DID并注册到链上；

- 通过WeIdService的getWeIdDocument()獲取對應DID的相關public key；

- 通過WeIdService的registerAuthorityIssuer()註冊有關機構成為AuthorityIssuer

* **為代理服務公司註冊 DID：**

|  |  |
| --- | --- |
| **代理公司** | **functionName : CreateWeId, getWeIdDocument (public key)** |
| 車行A | **did:weid:1:0xe71a5c41d749cd60a94574d797cf2db41acd6362**  11155932033748773519505914719313752212213348045073475523740549015135636699414065490680829863281888604325934923295416875762229721153897776684053144781465494 |
| 車行B | **did:weid:1:0x8a76b3c83dc508d704b77d2b3da81174e92a0823**  9404446263392011469416841937203143821909499703691587888030022312834603529982710728388654397923988744478691721007011776686676154642118015491133058571184495 |

* **為貸款機構註冊 DID：**

|  |  |
| --- | --- |
| **貸款機構** | **functionName : CreateWeId, getWeIdDocument (public key)** |
| 銀行A | **did:weid:1:0xabba807ed4408b2a705cf50fe153351e341142cd**  6801105129696240739616695658401489095517671600782623003065594152758122299844447215935468351306337427417793616085979604586715724896530005609815370730374034 |
| 借貸公司B | **did:weid:1:0xde318bf343e3a07dd250adec6699519e31f1c60c**  5507875379932321264906146818377171990368867956603657290445514810082965511112080100889435176561146746437114519602104925651990900526273229345733915546730068 |

* **為代理服務公司註冊 及 貸款機構 註冊成為授权机构(AuthorityIssuer)**
* 通過WeIdService的registerAuthorityIssuer()實現，並使用queryAuthorityIssuer進行驗證
* **其他測試用途DID**

**did:weid:1:0x71045fbfde4424eac2b098832373e6072881192d**

**Prepare02. 註冊CPT （CAPO）**

* **技術實現**

- 通过WeIdService的registercpt () 註冊cpt；



**Prepare03. 註冊CPT （CALOAN）**

* **技術實現**

- 通过WeIdService的registercpt () 註冊cpt；





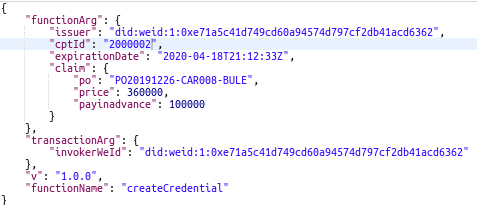
**Car Purchasing（Step1）. 購車貸款申請（車行A代理發怖邏輯）**

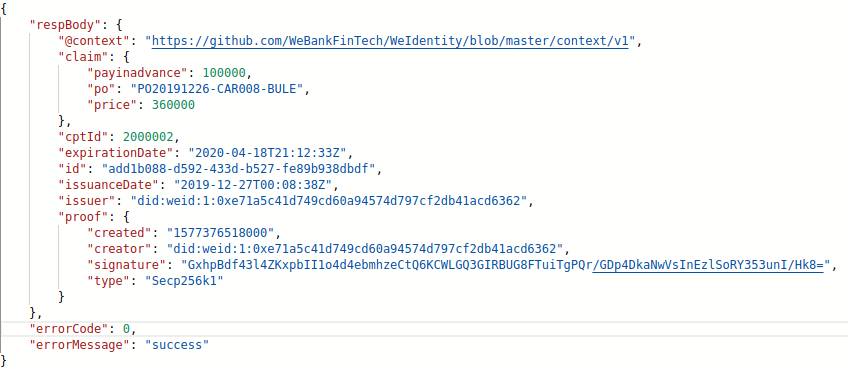
* **技術實現**

- 客戶與汽車銷售公司洽商銷售價格，達成購買相關產品意向；

- 針對需要進行貸款購買的客戶，將其採購信息上鏈，以便有關貸款機構進行審批核實

(通過WeIdService的createCredential ()實現)





**Loan Request（Step2）. 貸款申請**

* **技術實現**

- 客戶向指定銀行或貸款機構發出申請貸款請求，並出示有關CAPO credential；

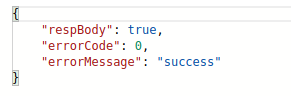
- 銀行或貸款機構驗證CAPO credential的真實性及有效性

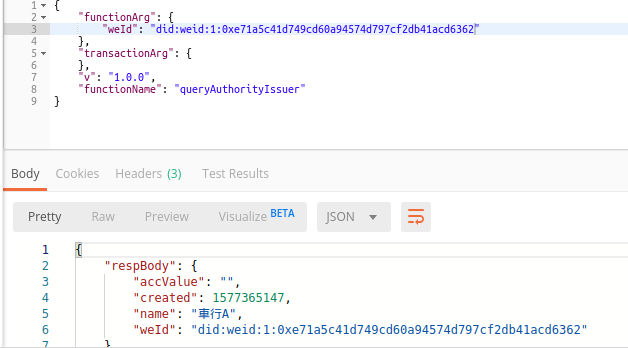
(通過WeIdService的verifyCredential ()實現)

- 銀行或貸款機構驗證issuer信息，確保其銷售業務符合該公司的貸款政策

(通過WeIdService的queryAuthorityIssuer ()實現)





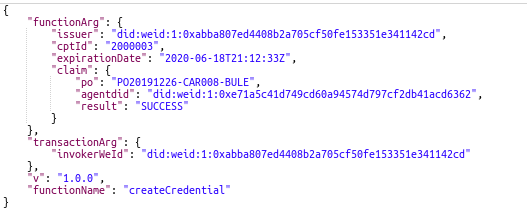


**Loan Request（Step3）. 發放貸款 （銀行A）**

* **技術實現**

- 銀行或貸款機構審批客戶貸款請求，並及時將其審批結果上鏈；

(通過WeIdService的createCredential ()實現)





**GotCar（Step4）. 提車 （客戶）**

* **技術實現**

- 客戶向車行展示貸款結果

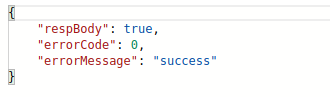
- 車行驗證CALOAN credential的真實性及有效性

(通過WeIdService的verifyCredential ()實現)

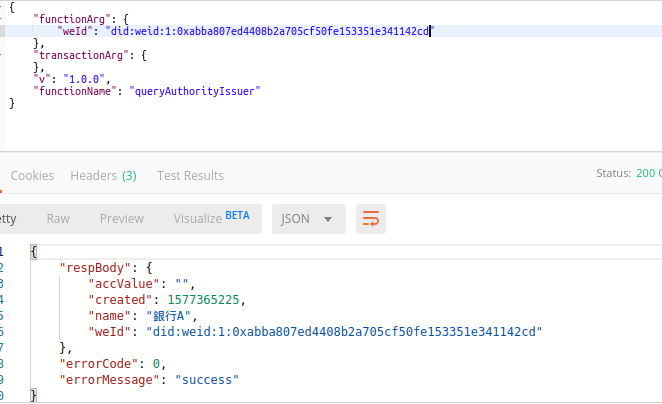
- 車行驗證CALOAN credential發自於銀行機構

- 車行根據CALOAN所描述的PO及AgentDID，確保有關的貸款發放對象及購買合約正確後，給予客戶提車權利，同時保留憑證，及時向有關銀行收取有關款項

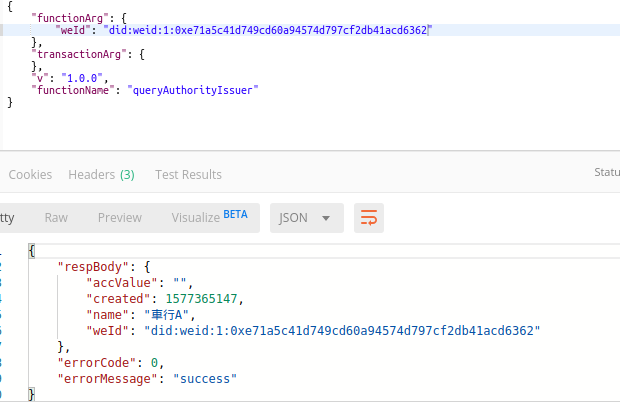




**檢查Issuer是否銀行**



**檢查agentdid是否自己**



**核對PO及提車人身份後發放車輛**