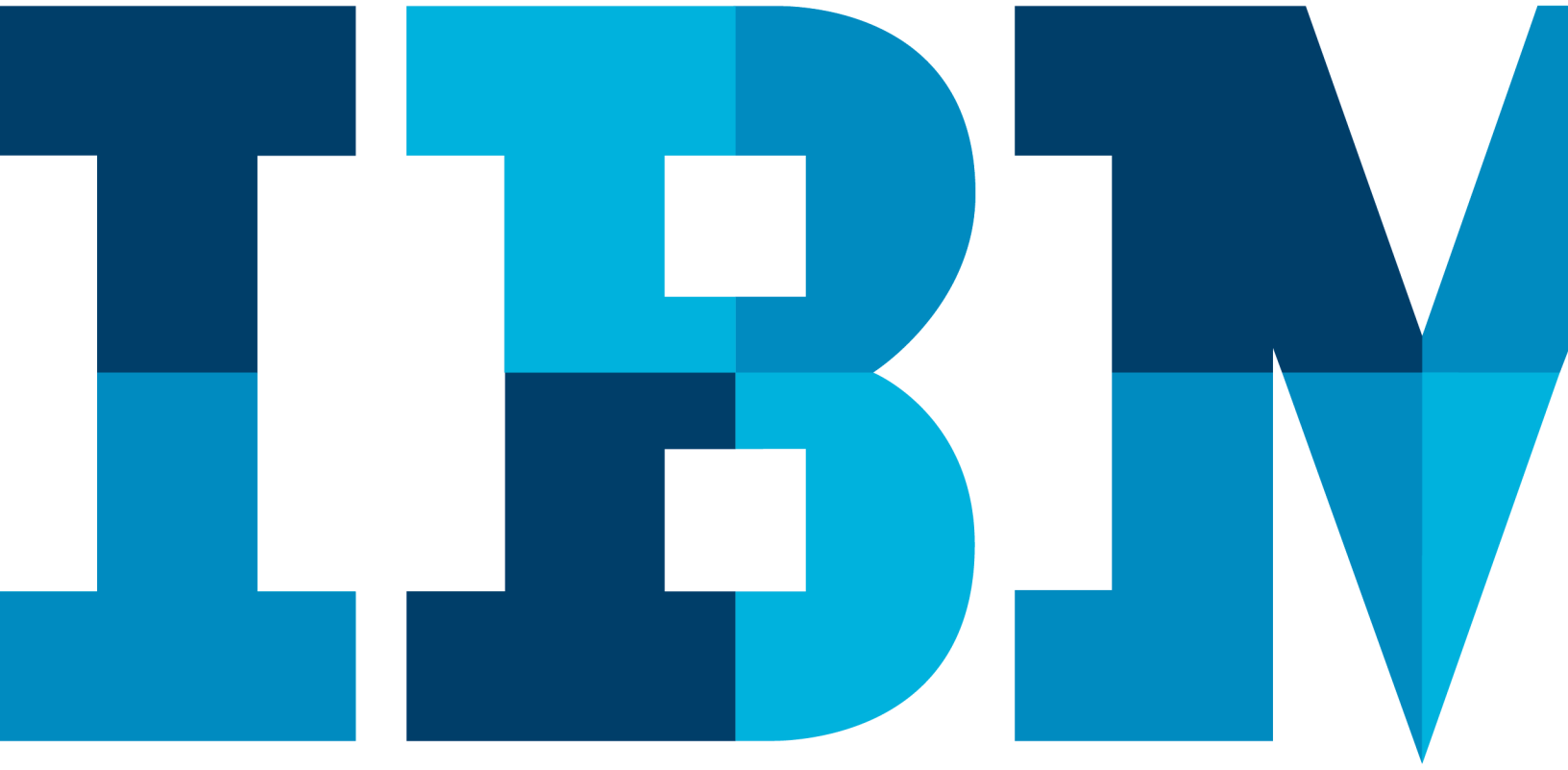
IBM Blockchain Proof of Technology Demo Scenarios Details and Changing the Scenarios

Lab02 Exercises



IBM

Contents

Overview

This lab will start to show the operational aspects of the demo in Lab 01. It will start to modify the demo to explore a few of the operational changes that may be carried out and understand a few of the methods used in Hyperledger

Introduction

Pre Requisites:

2 cores

4GB RAM

VMWare V10+

IBM_HyperLedger_Car_Leasing_Demo_v0.7+

Preferable to have completed 'IBM Blockchain PoT Lab Guide 01 first

Icons

The following symbols appear in this document at places where additional guidance is available.

Icon	Purpose	Explanation
	Important!	This symbol calls attention to a particular step or command. For example, it might alert you to type a command carefully because it is case sensitive.
	Information	This symbol indicates information that might not be necessary to complete a step, but is helpful or good to know.
	Trouble-shooting	This symbol indicates that you can fix a specific problem by completing the associated troubleshooting information.

Section 1. Start or Reset the Demo

If you have run IBM Blockchain PoT Lab 01 on the same machine as you are going to use for this lab, it will be beneficial to reset the demo to transfer the vehicles back to the initial state of the lab to ensure you have vehicles to use in this Lab.

If you have previously started the Demo – Please skip to part _4.

_ 1. Open a **Terminal** window

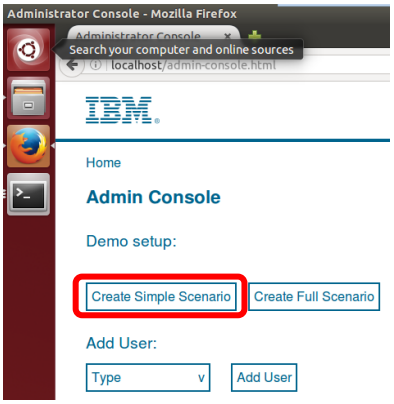
_ 2. Press the '**Up Arrow**' to bring up the command required to start the services:

```
ibm@ubuntu:~$ sh Desktop/startup.sh
```

The command should read '**`sh Desktop/startup.sh`**' if this is not the case, please enter this command

_ 3. Hit '**Enter**' to run this command

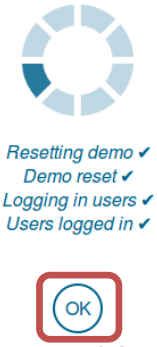
_ 4. In a browser, go to: http://localhost/admin-console.html



_ 5. Click Create Simple Scenario

This will initialise the demo with a blockchain and vehicles to use within the demo
It will take around 1 minute to complete

_ 6. Once complete, press '**OK'**



_ 7. Finally, click Home to go to the main page of the Demo

The demo is now ready to use.

Section 2. View the eCert of a Participant

This lab will show you how to find and read the contents of the eCert of any participant within the Demo. eCerts are used to certify each participant's credentials when performing a transaction, however these are not stored on the blockchain as the 'owner' or 'recipient' to prevent other participants who recognise the eCert from knowing the participants of other transactions they should not be able to view.

An eCert is the same as an X.509 cert with the addition of an extension field called 'roles' for showing which role each participant is authorised for

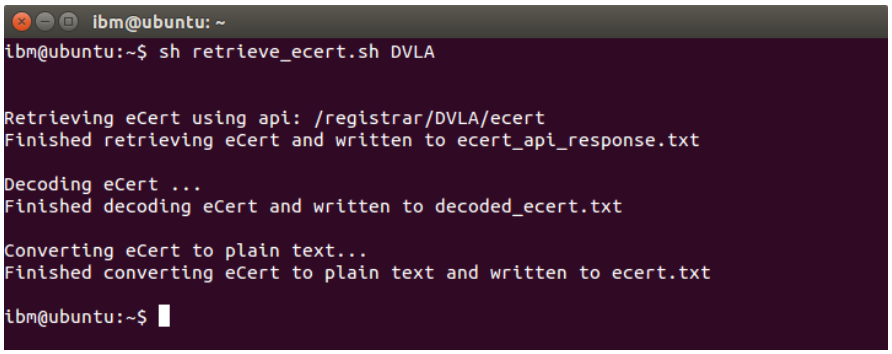
2.1. Viewing the HTML Encrypted eCert

This will show you how to view an eCert from the Certificate Authority. All members of the blockchain are authorised to view any eCert for participants on the same blockchain. The eCert is encoded in HTML encoding. To help with the decoding process we have added a script that will first remove the HTML encoding which will produce a X.509 format, the script will then turn this into plain text and save it the output.

_ 1. Open a new terminal window (**ctrl+alt+t**):

_ 2. In the terminal type: **`sh retrieve_ecert.sh DVLA`** similar to the image below:

To paste into the terminal, to <insert certificate>; right click in the terminal and use 'paste'



Troubleshooting: If this does not output the same as above please check you have completed the entire of Section 1. Particularly Section 1 _ 5 to "Start Demo"

The script uses an HTTP Get request to retrieve the ecert (if you wish to manually perform the request enter localhost:5000/registrar/DVLA/ecert).

At each stage of the operation the script outputs a file. To view this output open the following files.

Response received from calling the api:

`home/ibm/ecert_api_response.txt`

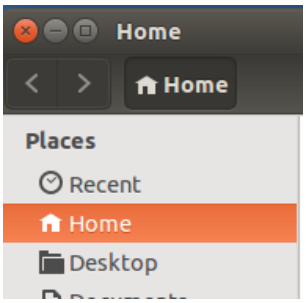
ECert with HTML encoding removed:

`home/ibm/decoded_ecert.txt`

2.2. View the contents of the eCert

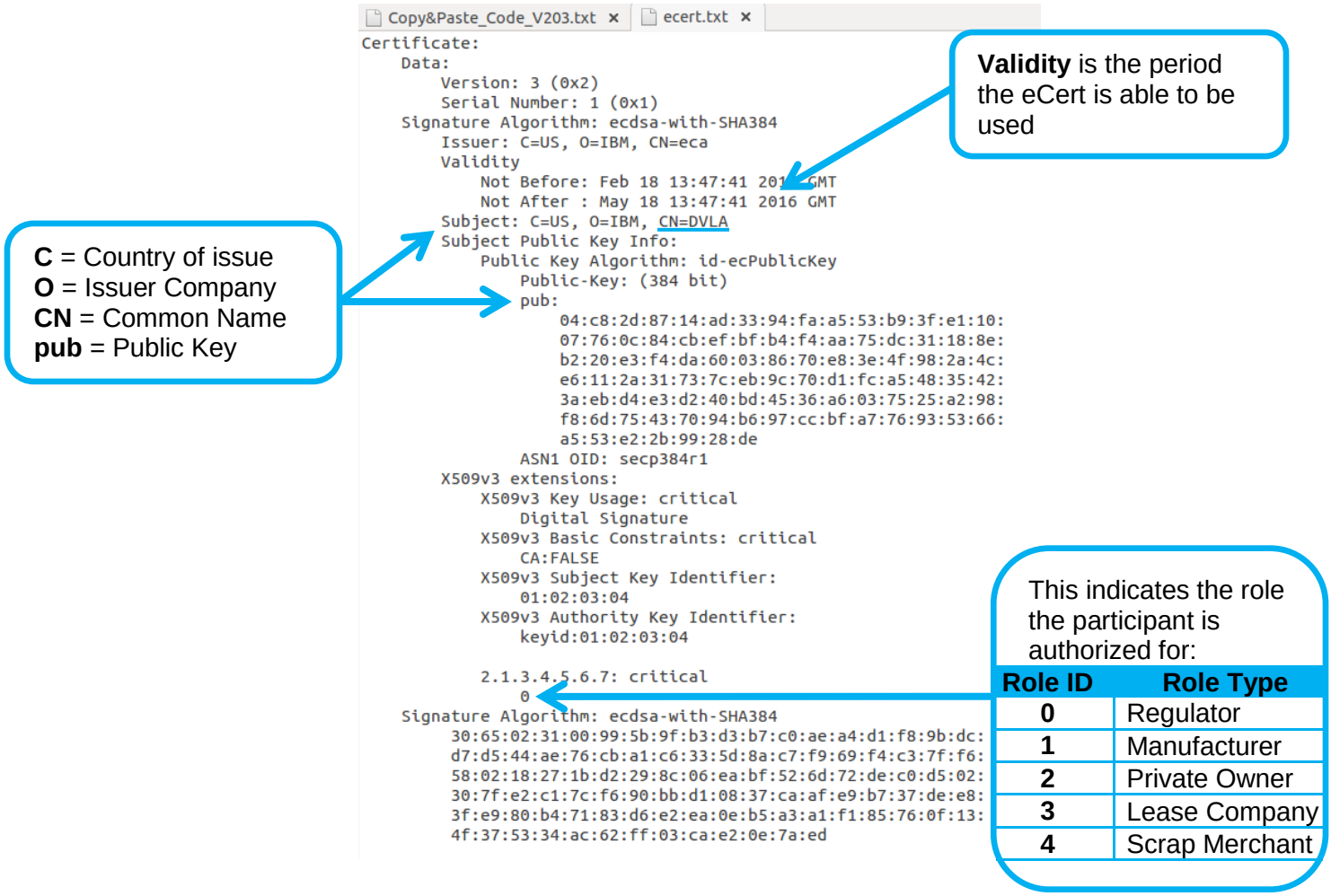
_ 1. Open the files browser

_ 2. Go to **Home**



_ 3. Open the file **ecert.txt**

_ 4. The output will be the eCert in its plane text format



2.3. Repeat for to view other Users

Once you have viewed the eCert for the DVLA, you can go back to 2.1 and change which participant you wish to view the eCert for. To do this change 'DVLA' in the sh command to any of the participants listed below:

_ 1. sh retrieve_ecert.sh DVLA

'**DVLA**' can be changed to any one of the login names e.g:

`Toyota` (A Manufacturer)
`Beechvale_Group` (A Dealership)
`LeaseCan` (A Lease Company)
`Joe_Payne` (A Leasee)
`Cray_Bros_(London)_Ltd` (A Scrap Merchant)

_ 2. Continue to follow 1.2 & 1.3 in the same method to view the certificates of the participants you have chosen and note their differing roles, keys and signatures.

Section 3. View the Contents of a Block

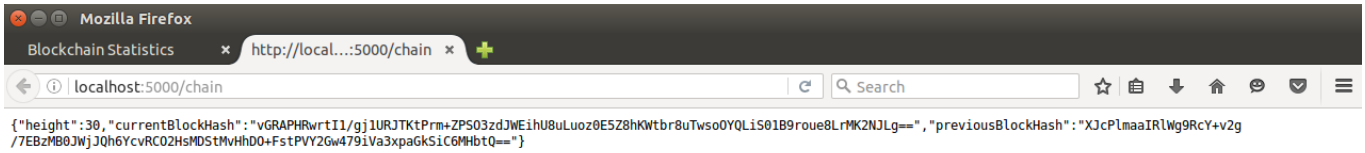
The blocks in Hyperledger contain information about the originator, recipient and the transaction but this is not visible to every participant. This lab shows how to view the contents of a block and starts to show the reasons why any participant cannot view the contents of every block.

3.1. Find how many blocks are in the chain

_ 1. Open a browser or a new tab

_ 2. Go to localhost:5000/chain

_ 3.



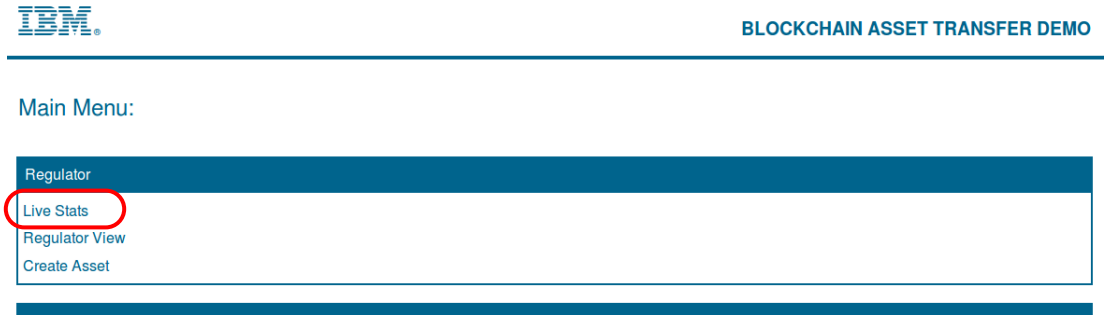
Height is the number of blocks in the chain

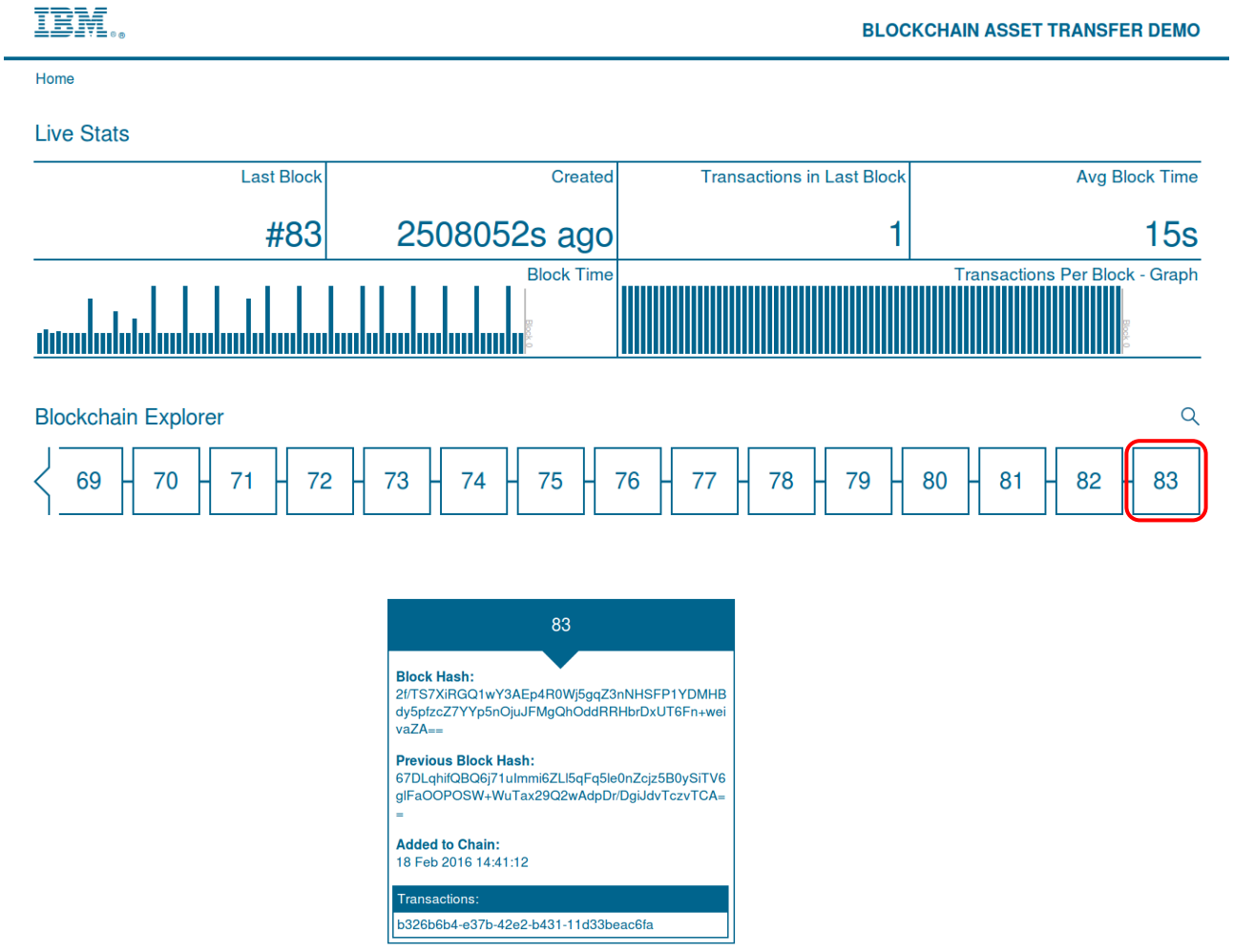
Blocks start at 'Block 0' so the latest block number will be called number *height* – 1. In this example the last block added is 'Block 82'

This view also shows the **currentBlockHash** and the **previousBlockHash**, showing how the blocks are linked together.

3.2. View contents of a specific block

_ 1. Go to your browser and navigate to the “Live Stats” page:





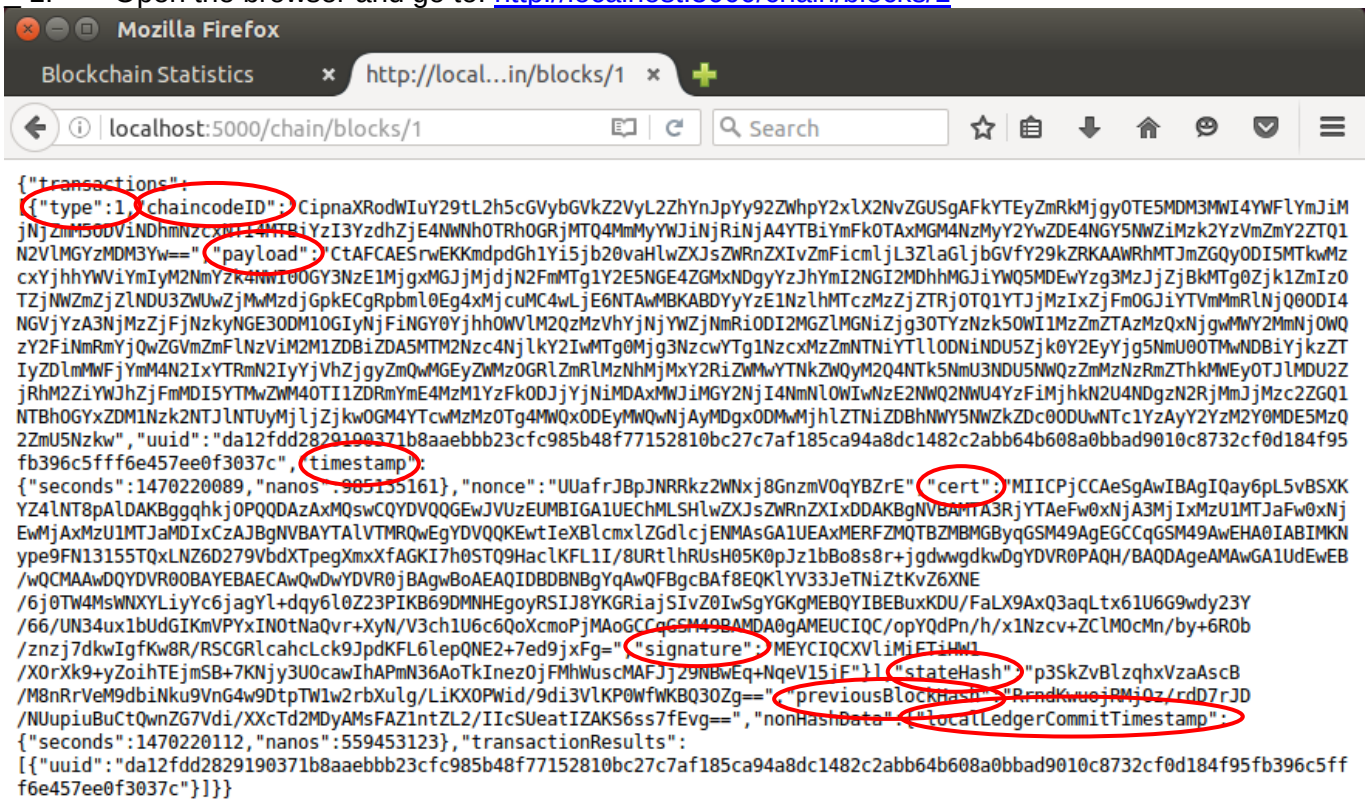
_ 2. Using the block explorer click on block 83:

_ 3. This will show you some of the blocks contents:

This visual display of a block shows the hash of the current block and the hash of the previous block (Note: The hash of the current block is not stored in the block itself. We retrieve it by looking at the next block in the chain). It also stores the date when the block was added to the chain on the local peer and a table of the transactions in the block. The transactions are identified by their uuid.

3.2.1. View the block as it is returned by the API

_ 1. Open the browser and go to: http://localhost:5000/chain/blocks/1



{"transactions":
[{"type":1,"chaincodeID":CipnaXRodWIuY29tL2h5cGVybGVkZ2VyL2ZhYnJpYy92ZWhpY2xlX2NvZGUSgAFkYTEyZmRkMjgyOTE5MDM3MWI4YWFlYmJiM
jNjZmM5ODViNDhmNzcxMjI4NTBiYzI3YzdhZjE4NWNhOTRhOGRjMTQ4MmMyYWJiNjRiNjA4YTBiYmFkOTAxMGM4NzMyY2YwZDE4NGY5NWZiMzk2YzVmZmY2ZTQ1
N2VlMGYzMDM3Yw==","payload":CtAFCAESrwEKKmdpdGh1Yi5jb20vaHlwZXJsZWRnZXIvZmFicmljL3ZlaGljbGVfY29kZRKAAWRhMTJmZGQyODI5MTkwMz
cxYjhhYWViYmIyM2NmYzk4NWI0OGY3NzE1MjgxMGJjMjdjN2FmMTg1Y2E5NGE4ZGMxNDgyYzJhYmI2NGI2MDhhMGJiYWQ5MDEwYzg3MzJjZjBkMTg0Zjk1ZmIzO
TZjNWZmZjZlNDU3ZWUwZjMwMzdjGpkECgRpbml0Eg4xMjcuMC4wLjE6NTAwMBKABDYyYzE1NzlhMTczMzZjZTRjOTQ1YTJjMzIxZjFmOGJiYTVmMmRlNjQ0ODI4
NGVjYzA3NjMzZjFjNzkyNGE3ODM1OGIyNjFiNGY0YjhhOWVlM2QzMzVhYjNjYWZjNmRiODI2MGZlMGNiZjg3OTYzNzk5OWI1MzZmZTAzMzQxNjgwMWY2MmNjOWQ
zY2FiNmRmYjQwZGVmZmFlNzViM2M1ZDBiZDA5MTM2Nzc4NjlkY2IwMTg0Mjg3NzcwYTg1NzcxMzZmNTNiYTllODNiNDU5Zjk0Y2EyYjg5NmU0OTMwNDBiYjkzZT
IyZDlmMWFjYmM4N2IxYTRmN2IyYjVhZjgyZmQwMGEyZWMzOGRlZmRlMzNhMjMxY2RiZWMwYTNkZWQyM2Q4NTk5NmU3NDU5NWQzZmMzNzRmZThkMWEyOTJlMDU2Z
jRhM2ZiYWJhZjFmMDI5YTMwZWM4OTI1ZDRmYmE4MzM1YzFkODJjYjNiMDAxMWJiMGY2NjI4NmNlOWIwNzE2NWQ2NWU4YzFiMjhkN2U4NDgzN2RjMmJjMzc2ZGQ1
NTBhOGYxZDM1Nzk2NTJlNTUyMjljZjkwOGM4YTcwMzMzOTg4MWQxODEyMWQwNjAyMDgxODMwMjhlZTNiZDBhNWY5NWZkZDc0ODUwNTc1YzAyY2YzM2Y0MDE5MzQ
2ZmU5Nzkw","uuid":"da12fdd2829190371b8aaebbb23cfc985b48f77152810bc27c7af185ca94a8dc1482c2abb64b608a0bbad9010c8732cf0d184f95
fb396c5fff6e457ee0f3037c",timestamp":
{"seconds":1470220089,"nanos":985135161},"nonce":"UUafrJBpJNRRkz2WNxj8GnzmVOqYBZrE","cert":"MIICPjCCAeSgAwIBAgIQay6pL5vBSXK
YZ4lNT8pAlDAKBggqhkjOPQQDAzAxMQswCQYDVQQGEwJVUzEUMBIGA1UEChMLSHlwZXJsZWRnZXIxDDAKBgNVBAMTA3RjYTAeFw0xNjA3MjIxMzU1MTJaFw0xNj
EwMjAxMzU1MTJaMDIxCzAJBgNVBAYTAlVTMRQwEgYDVQQKEwtIeXBlcmxlZGdlcjENMAsGA1UEAxMERFZMQTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABIMKN
ype9FN13155TQxLNZ6D279VbdXTpegXmxXfAGKI7h0STQ9HaclKFL1I/8URtlhRUsH05K0pJz1bBo8s8r+jgdwwgdkwDgYDVR0PAQH/BAQDAgeAMAwGA1UdEwEB
/wQCMAAwDQYDVR0OBAYEBAECAwQwDwYDVR0jBAgwBoAEAQIDBDBNBgYqAwQFBgcBAf8EQKlYV33JeTNiZtKvZ6XNE
/6j0TW4MsWNXYLiyYc6jagYl+dqy6l0Z23PIKB69DMNHEgoyRSIJ8YKGRiajSIvZ0IwSgYGKgMEBQYIBEBuxKDU/FaLX9AxQ3aqLtx61U6G9wdy23Y
/66/UN34ux1bUdGIKmVPYxINOtNaQvr+XyN/V3ch1U6c6QoXcmoPjMAoGCCqGSM49BAMDA0gAMEUCIQC/opYQdPn/h/x1Nzcv+ZClMOcMn/by+6ROb
/znzj7dkwIgfKw8R/RSCGRlcahcLck9JpdKFL6lepQNE2+7ed9jxFg=","signature":MEYCIQCXVliMiETiHW1
/XOrXk9+yZoihTEjmSB+7KNjy3UOcawIhAPmN36AoTkInezOjFMhWuscMAFJj29NBwEq+NqeV15jF"}],"stateHash":"p3SkZvBlzqhxVzaAscB
/M8nRrVeM9dbiNku9VnG4w9DtpTW1w2rbXulg/LiKXOPWid/9di3VlKP0WfWKBQ3OZg==","previousBlockHash":"RrndKwuejRMjOz/rdD7rJD
/NUupiuBuCtQwnZG7Vdi/XXcTd2MDyAMsFAZ1ntZL2/IIcSUeatIZAKS6ss7fEvg==","nonHashData":{"localLedgerCommitTimestamp":
{"seconds":1470220112,"nanos":559453123},"transactionResults":
[{"uuid":"da12fdd2829190371b8aaebbb23cfc985b48f77152810bc27c7af185ca94a8dc1482c2abb64b608a0bbad9010c8732cf0d184f95fb396c5ff
f6e457ee0f3037c"}]}}

Block Contents Description:

transaction: An array of transactions stored in the block. This array is made up of transaction objects which contain:

type - Transaction type that is taking place. See table for reference:

Type ID	Action taken
0	Undefined
1	Deployment
2	Update
3	Invoke
4	Query
5	Terminate

chaincodeID – the ID of the chaincode that was called on

payload – Contains all the processes being made in the transaction such as functions, assets names and arguments
uuid - The unique identifier of this transaction
timestamp - The time at which the block or transaction order was proposed
cert – Certificate of the participant submitting the transaction
signature - Signature of the participant submitting the transaction
stateHash - The hash of all the world state
previousBlockHash - The hash of the previous block in the chain
consensusMetadata - Consensus modules may optionally store any additional metadata in this field
nonHashData - Data stored with the block, but not included in the blocks hash. This allows this data to be different per peer or discarded without impacting the blockchain
localLedgerCommitTimestamp - The time at which the block was added to the ledger on the local peer

Section 4. Add a New Participant

In this section we will add you as a leasee into the demo. This will involve four stages in this demo

4.1. Add your log in credentials with the Certificate Authority
4.2. Change the config file listing possible Leasee recipients
4.3. Make the browser open the new participants Transfer Assets page
4.4. Testing transferring a car to your new user

4.1. Add your log in credentials with the Certificate Authority

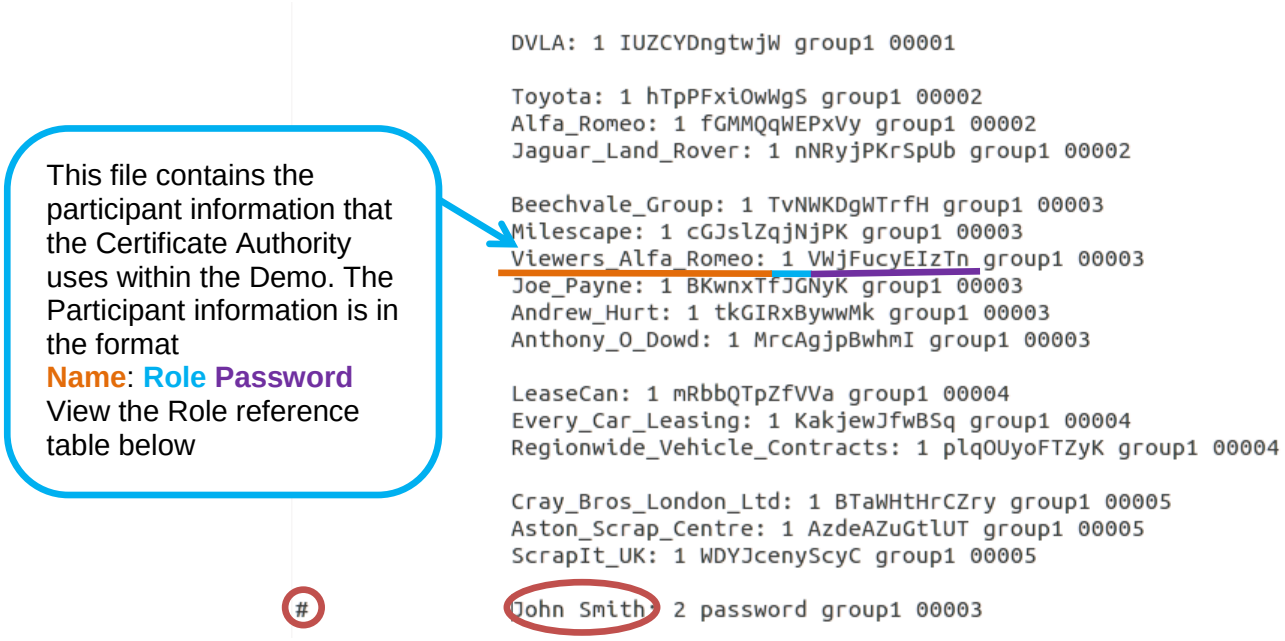
In this demo, a real Certificate Authority has not been used to enable the Demo to run offline and use multiple users concurrently. The OBC CA uses the obcca.yaml to lookup user credentials. To enable the addition of new users for this demo, this file is in plain text showing usernames, roles and passwords. Outside of this demo, IBM Blockchain would use a trusted Certificate Authority and encryption to store user details.

4.1.1. Add your credentials to the .yaml file

_ 1. Open the file browser

_ 2. Go to: Documents > IBM_Blockchain > Projects > src > github.com > hyperledger > fabric > membersrvc

_ 3. Open the file **membersrvc.yaml**



_ 4. To add yourself into this list, remove the '#' comment indicator (circled in red above) to enable the line of code and choose a name to replace '**John Smith**'

The **roles** referenced above are:

Role ID	Role Type
0	Regulator
1	Manufacturer
2	Private Owner
3	Lease Company
4	Scrap Merchant

In this demo, a 'Leasee' and 'Dealership' both viewed as a 'Private Owner' from the viewpoint of the CA. A 'Private Owner' is someone who can only transfer a vehicle and not create/define it or scrap it.

_ 5. **Save** the file

4.1.2. Restart the Peer

To log the user with the CA, the Certificate Authority & Peer need to be restarted. Starting with the peer:

_ 1. Open a new Terminal (**ctrl+alt+t**)

_ 2. Change the directory by entering: **`cd $GOPATH/src/github.com/hyperledger/fabric/devenv`**

The terminal window to the left of the flashing cursor should now show the new directory location that you entered above. If an error ending "No Such File Directory" is shown, the command used was incorrect

_ 3. Enter: **vagrant ssh**

_ 4. **Wait** for Vagrant to load

_ 5. Enter: **killall peer**

_ 6. Enter: **cd $GOPATH/src/github.com/hyperledger/fabric**

_ 7. Enter: **peer node start**

_ 8. **Wait** for peer to load

_ 9. **Minimise** this terminal

4.1.3. Restart the Certificate Authority

_ 1. Open a **new Terminal** (ctrl+alt+t)

_ 2. Enter: **`cd $GOPATH/src/github.com/hyperledger/fabric/devenv`**

_ 3. Enter: **vagrant ssh**

_ 4. **Wait** for Vagrant to load

_ 5. Enter: **killall mambersrvc**

_ 6. Enter: **cd $GOPATH/src/github.com/hyperledger/fabric/membersrvc**

_ 7. Enter: **make membersrvc**

```
vagrant@hyperledger-devenv:v0.0.10-
3e0e80a:/opt/gopath/src/github.com/hyperledger/fabric $
make membersrvc
INFO: 2016/02/26 12:57:53 CA Server (0.1)
INFO: 2016/02/26 12:57:54 ECA started.
INFO: 2016/02/26 12:57:54 TCA started.
INFO: 2016/02/26 12:57:54 TLSCA started.
```

_ 8. **Minimise** this terminal

4.1.4. Register the user

The user has been logged as a user with the Certificate Authority, however the user must now register to enable them. To do this we used a Post Request:

_ 1. Open a **new terminal**

_ 2. Enter: **curl -H "Content-Type: application/json" -X POST -d '{"enrollId":"<username>","enrollSecret":"<password>"}' http://localhost:5000/registrar**

_ 3. Change the **<username>** & **<password>** to match your new participant using "_" instead of spaces

_ 4. A successful login message will be shown:

```
ibm@ubunntu :~$ curl -H "Content-Type: application/json" -X POST -d
'{"enrollId":"John_Smith","enrollSecret":"ibm4you2"}'
http://localhost:5000/registrar"
{"OK": "Login successful for user 'John_Smith'."}ibm@ubuntu:~$
```

If this does not succeed; check The peer (5.1.2) & CA (5.1.3) have loaded properly, The credentials (5.1.1) inserted match the credentials used here and use a "_" instead of any spaces and finally ensure that the demo is running properly by checking the "Live Stats" page opens.

4.1.5. Test the Participant has been added into the Certificate Authority

Testing the Certificate Authority recognised that the participant is authorised can be done by viewing the certificate of the participant

_ 1. In a browser, go to **http://localhost:5000/registrar/<username>/ecert**

If successful, a certificate will be shown.

4.2. Change the config file listing possible Leasee recipients

_ 1. This file is part of the Demo web page and not part of Hyperledger. It is used to find which participants will appear to be chosen to transfer an asset to.

_ 2. Open the **File Browser**

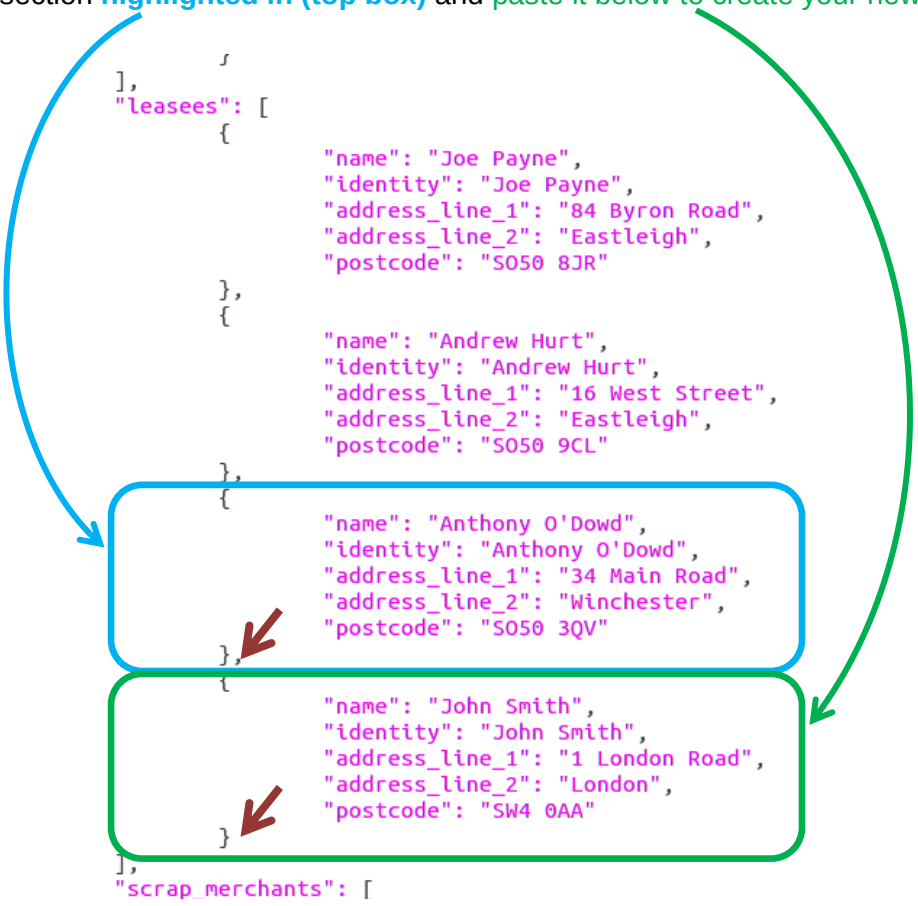
_ 3. Go to: **Home > Documents > car-lease-demo-1 > Server_Side > blockchain > participants**

_ 4. Open file **participants.json**

This file shows all of the details for each participant which is used

_ 5. Scroll down the `"leasees":` section

_ 6. Copy the section **highlighted in (top box)** and paste it below to create your new identity.



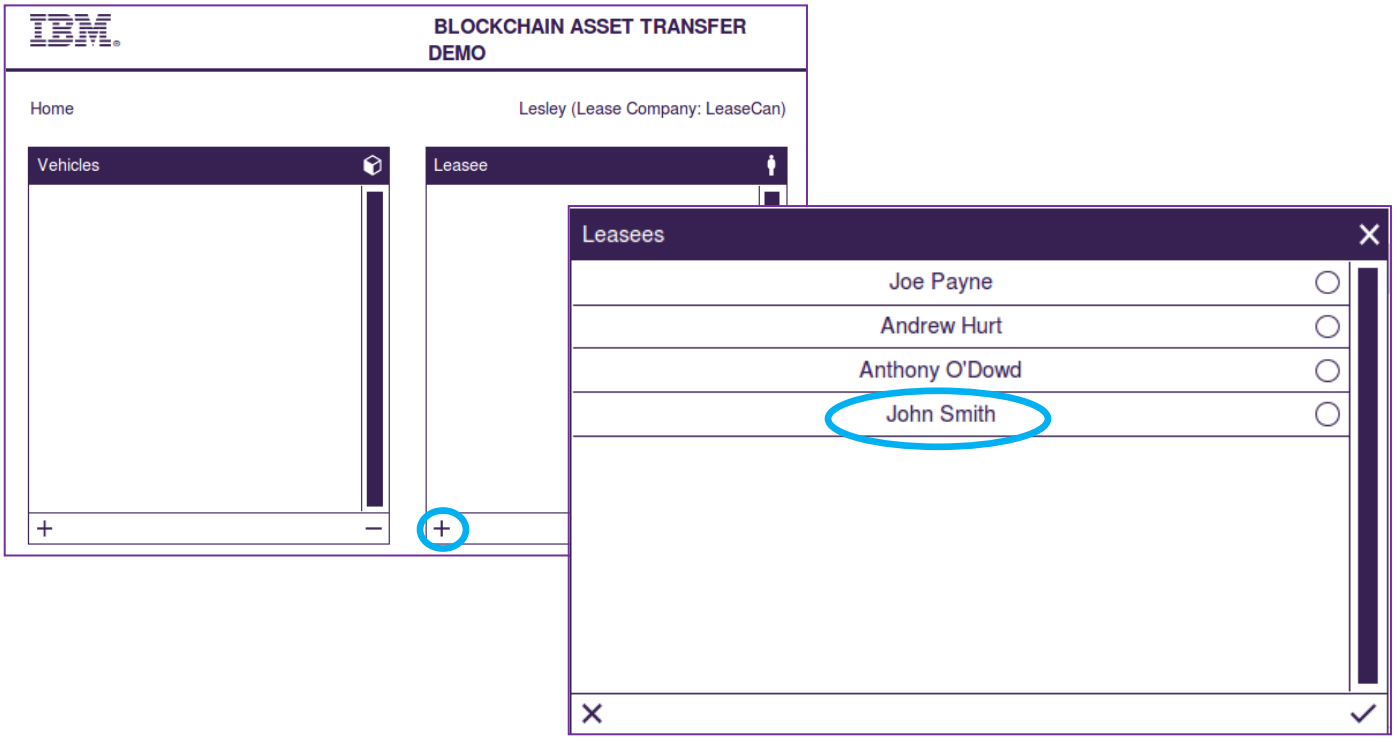
_ 7. **Change the 'name' & 'identity' fields** to match what you entered for the Certificate Authority.

Important! Do not forget to add the comma in place after the second last participant but not after the last participant

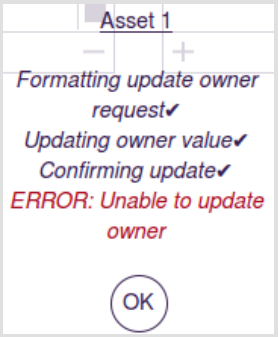
_ 8. **Save** the file

4.2.1. Check the Config file change worked

_ 1. To check this has worked properly, go to the Lease Company → Leasee page and view the possible leasees that can be transferred to:



Troubleshooting: If you get an error when transferring that looks like this, you have not added your participant to the Certificate Authority properly in 4.1



4.3. Change the browser settings to open your participants Transfer Assets page

The web browser reads from the config file to know which participants and companies should be available when transfer asset page is opened. By editing this file we can change the users that appear in the list when you select to switch user.

_ 1. Open the **File Browser**

_ 2. Go to: **Home > Documents > Demo > Client_Side > JavaScript > config**

_ 3. Open the **config.js** file

This file provides which companies & participants the web page will offer up for display when opening the transfer asset pages. This enables us to easily change which page is opened to view their assets.

_ 4. Scroll down to the `//Leasees` section.

Copy the section highlighted in (top box) and **paste it below to create your new identity**.

```
// Leasees
config.participants.users.leasees[0] = {}
config.participants.users.leasees[0].company = 'Joe Payne'
config.participants.users.leasees[0].type = 'Leasee'
config.participants.users.leasees[0].user = 'Joe'
config.participants.users.leasees[0].password = 'ibm4you2'

config.participants.users.leasees[1] = {}
config.participants.users.leasees[1].company = 'Andrew Hurt'
config.participants.users.leasees[1].type = 'Leasee'
config.participants.users.leasees[1].user = 'Andrew'
config.participants.users.leasees[1].password = 'ibm4you2'

config.participants.users.leasees[2] = {}
config.participants.users.leasees[2].company = 'Anthony O\'Dowd'
config.participants.users.leasees[2].type = 'Leasee'
config.participants.users.leasees[2].user = 'Anthony'
config.participants.users.leasees[2].password = 'ibm4you2'

config.participants.users.leasees[3] = {}
config.participants.users.leasees[3].company = 'John Smith'
config.participants.users.leasees[3].type = 'Leasee'
config.participants.users.leasees[3].user = 'John'
config.participants.users.leasees[3].password = 'ibm4you2'
```

Important! Do not forget to change the number in the square brackets to be 3.

_ 5. Save the **config.js** file

4.4. Try out transferring assets to and from the different participant and company

_ 1. Go back to the browser

_ 2. Go to the home page of the demo site

_ 3. Open the '**Lease Company -> Leasee**' transfer asset page.

_ 4. Select a vehicle to transfer, select your new participant to receive the vehicle and transfer the vehicle

Troubleshooting: If the there is no option to transfer assets to your new participant then 4.2 was not completed correctly

_ 5. Go to the home page of the demo site

_ 6. Open up the transfer asset page for the role of your participant

_ 7. Use the three lines to bring up a list of leasees and select your user from there.

_ 8. Transfer the asset to a Scrap Merchant to show you can transfer out from your new participant

Troubleshooting:
If the Transfer Assets page has the incorrect name, then 4.3 was not completed correctly

Appendix A. Troubleshooting

Appendix B. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have

been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

Appendix C. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
System z	*Tivoli*	*WebSphere*	*Workplace*	*System p*	

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

NOTES

NOTES

Please Recycle

IBM Software