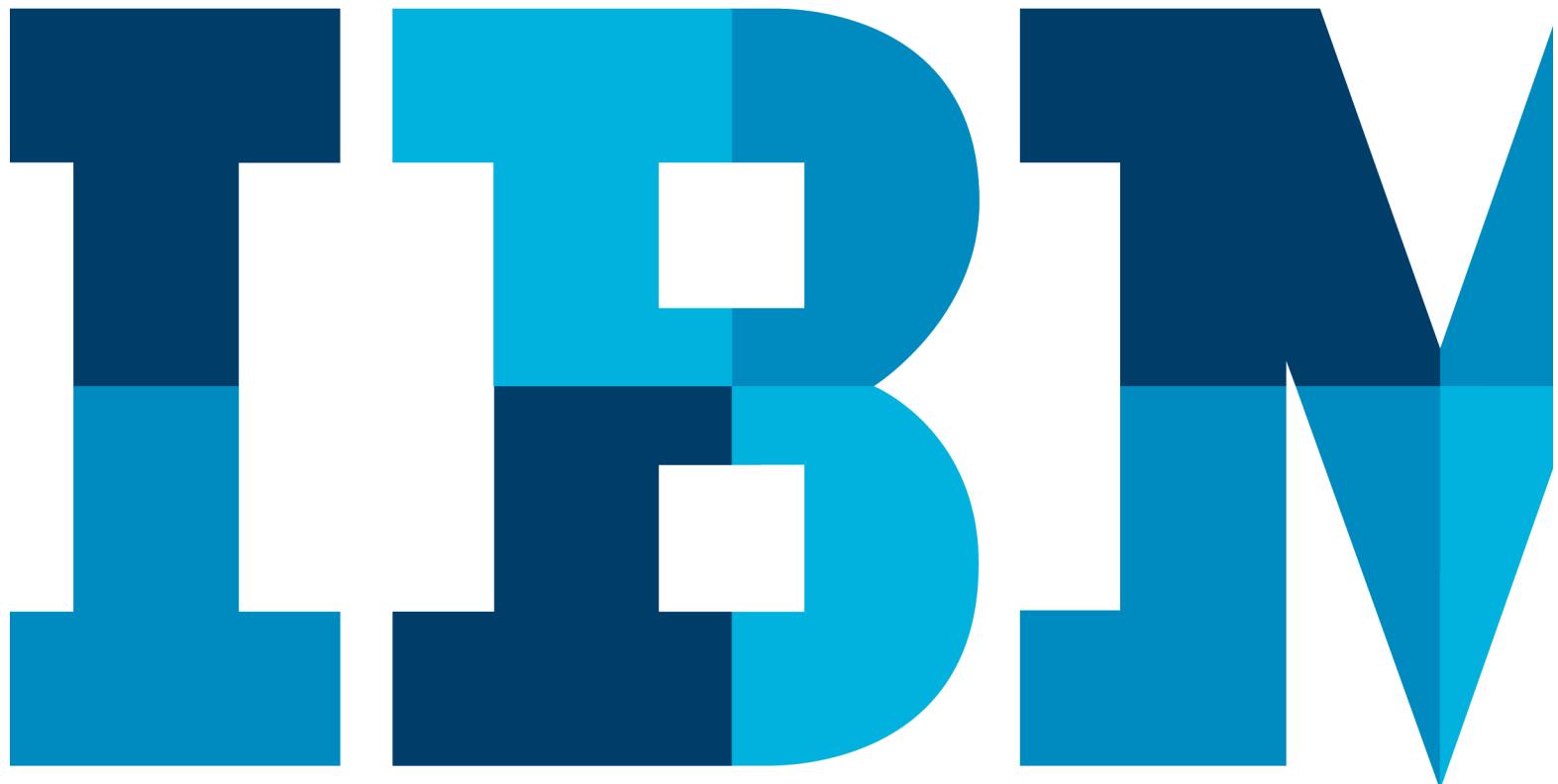


# IBM Blockchain Hands-On Blockchain Explained

*Lab One – Vehicle Lifecycle Demo VM – Exercises*



---

## Contents

<b>RUNNING THE DEMO.....</b>	<b>4</b>
1.1.    ORDER THE NEW CAR .....	4
1.2.    MANUFACTURING PROCESS .....	6
1.3.    INSURE THE CAR.....	9
1.4.    INTERNET OF THINGS INTEGRATION .....	11
1.5.    REGULATOR ANALYTICS .....	15
<b>APPENDIX A. STARTING THE SAMPLE APPLICATION.....</b>	<b>17</b>
<b>APPENDIX B. KEYBOARD LANGUAGE CHANGE.....</b>	<b>18</b>
<b>APPENDIX C. NOTICES.....</b>	<b>20</b>
<b>APPENDIX D. TRADEMARKS AND COPYRIGHTS .....</b>	<b>22</b>

---

## Overview

The purpose of this lab is to introduce you to the concepts of a blockchain by showing you how a blockchain transfers assets between participants in a business network. We will use car leasing as the scenario for the demo.

## Introduction

The VM requires:

- 2 cores
- 4 GB RAM (8GB recommended)
- 40 GB Hard Drive

The virtual machine is based on Linux Ubuntu 14.04 and contains Composer Playground, Node-Red, Loopback, Yoman, Docker, Docker-Compose, Visual Studio Code with Hyperledger Composer extension.

There are no additional files or software that is proprietary to the lab in the virtual machine. This means that the lab may be run on a machine without a virtual machine if Hyperledger Fabric and the other prerequisites have been installed.

Once the setup has been completed (done for you on the VM, follow appendix A. if this has not been completed) will open the following pages in your browser:

- Customers Car Order App: <http://localhost:8100/>
- Arium Manufacturer App: <http://localhost:6002/>
- VDA (Regulator) App: <http://localhost:6001/>
- Prince Insurer App: <http://localhost:4200/overview>
- Node Red Flows: <http://localhost:1880/>
- REST APIs: <http://localhost:3000/explorer/>
- Hyperledger Composer: <http://localhost:8080/>

## Running the demo

In this demo you will be playing the role of the 3 personas who use the vehicle lifecycle system.

- Paul the buyer/owner
- Mike an employee for the car manufacturer called Arium
- Debbie an administrator for the regulator called Vehicle & Drivers Authority (VDA)

These personas together work on ordering, building, transferring ownership of a vehicle whilst keeping all the other parties in the network updated and building the trust between them to allow them to work together efficiently.

### 1.1. Order the New Car

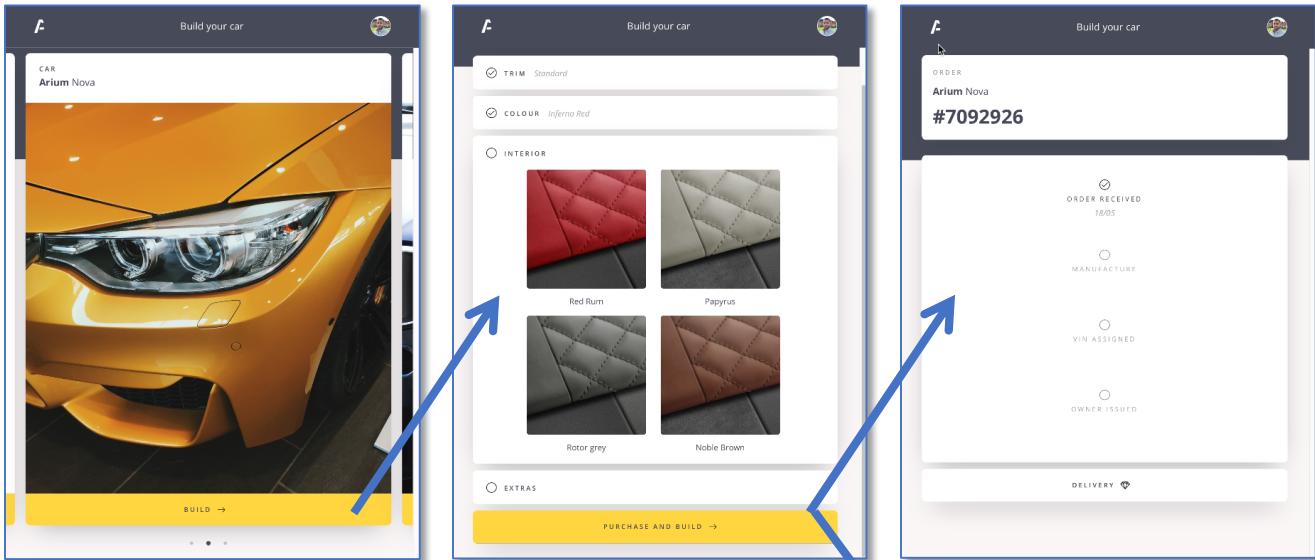
1. Open up the '**Ionic App**' tab (localhost:8100) and if possible the '**Vehicle & Drivers Authority**' tab (localhost:6001) next to it. If the tabs do not appear in the Chrome browser due to the instructor not prestarting the applications, then perform the following:
  - a. Open a terminal window
  - b. cd /VLD
  - c. ./restart.sh

Timestamp	Transaction ID	Transaction Type	Transaction Submitter
May 18, 2017 1:47:27 AM	9100b245-6618-496d-8691-c834af606cc	SetupDemo	Liam Grace

This will log you into the ordering app as Paul and into the VDA dashboard as Debbie able to view the transactions in the network so far.

2. In the Arium 'Ionic App', click '**Build Your Car**'

3. Swipe left and right to decide which car to build and then decide the options on your car

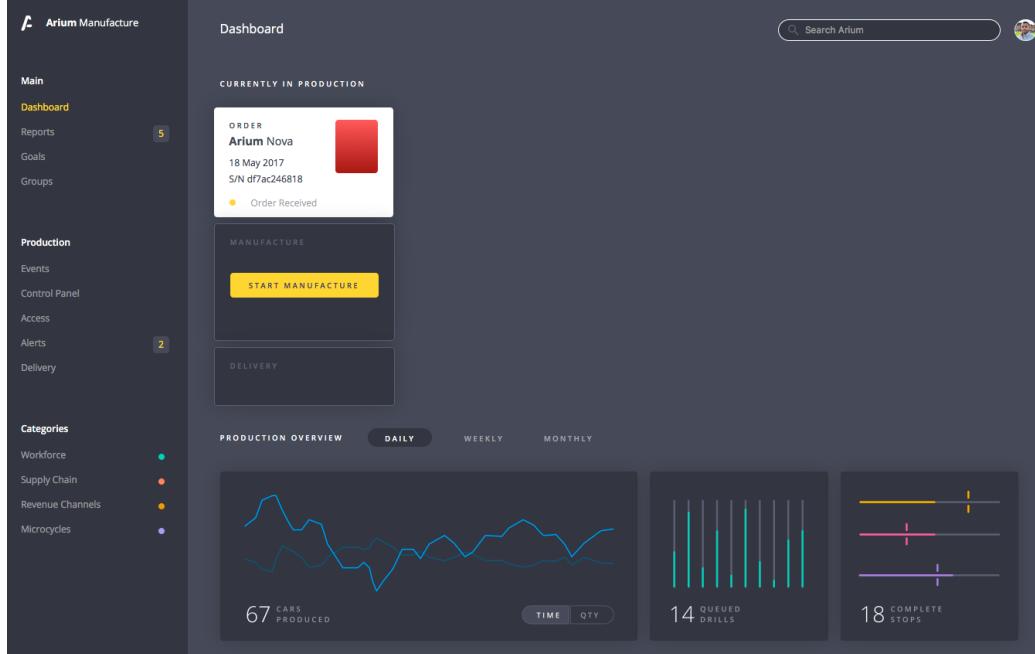


4. Once you place the order, you will immediately see the VDA view update with the latest block in blue and add the transaction to the list in white

Timestamp	Transaction ID	Transaction Type	Transaction Submitter
May 19, 2017 2:55:28 PM	2034f288-c278-498b-95d7-c8aa400699a	UpdateOrderStatus	Arium Vehicles
May 19, 2017 2:55:27 PM	ae1776cd-8c34-4d1b-9826-5c97ddfc04de	PlaceOrder	Arium Vehicles
May 19, 2017 1:47:27 AM	9100b245-6618-496d-8691-c834afdf606cc	SetupDemo	Liam Grace

## 1.2. Manufacturing Process

- After the car has been ordered, switch to the Manufacturer view (localhost:6002) and see your new car



- Click on **Start Manufacture**

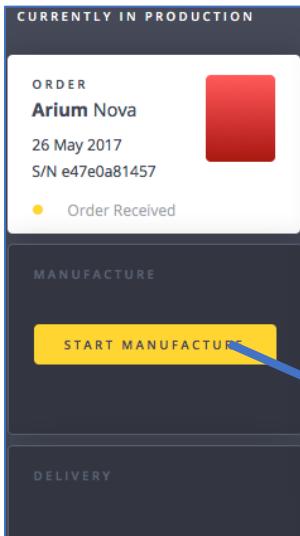
This will initiate the production process which has been simulated, over the next several seconds, the vehicle will be 'built' updating the blockchain as reached certain points of production. These can be seen in both the Manufacturer dashboard and VDA dashboard

The image displays three sequential screenshots of the Arium Manufacturer dashboard, illustrating the progression of an order from 'Incomplete' to 'Complete'.

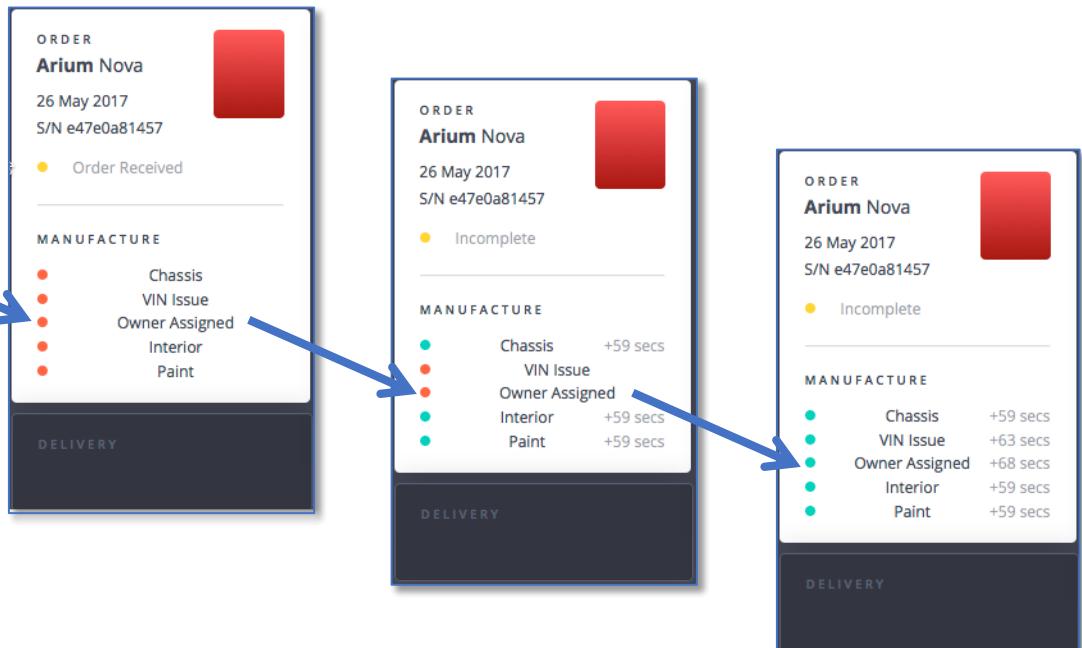
**Screenshot 1 (Incomplete):** The 'CURRENTLY IN PRODUCTION' section shows the order 'Arium Nova' with a status of 'Incomplete'. The 'MANUFACTURE' section lists tasks: Chassis (+286 secs), Exterior (+286 secs), Owner Assigned, Interior (+286 secs), and Paint (+286 secs). The 'DELIVERY' section shows 'Shipping' with a duration of +301 secs. The 'RECENT TRANSACTIONS' table shows four transactions (#138, #139, #140, #141) with details like 'Vehicle & Driver Authority' and 'UnderConstruction PLACED'.

**Screenshot 2 (Complete):** The 'CURRENTLY IN PRODUCTION' section now shows the order 'Arium Nova' with a status of 'Complete'. The 'MANUFACTURE' section shows completed tasks: Chassis, Exterior, Owner Assigned, Interior, and Paint. The 'DELIVERY' section shows 'Shipping' with a duration of +301 secs. The 'RECENT TRANSACTIONS' table shows four transactions (#141, #142, #143, #144) with details like 'Vehicle & Driver Authority' and 'UnderConstruction UNPLACED'.

**Screenshot 3 (New Transaction):** A new transaction is being initiated, indicated by the 'NEW TRANSACTION' button. The transaction ID is #144, and the action is 'UpdateOrderStatus'. The 'RECENT TRANSACTIONS' table shows the transaction with details like 'Vehicle & Driver Authority' and 'UnderConstruction UNPLACED'.



7. See how the manufacturers dashboard updates as the vehicle is built



At the same time the VDA dashboard is also being updated with the transaction data information

Timestamp	Transaction ID	Transaction Type	Transaction Submitter
May 26, 2017 4:48:48 PM	883faafb-72b7-4eb7-b90f-266d989351975	UpdateOrderStatus	Arium Vehicles
May 26, 2017 4:48:47 PM	eff0c940-f26a-4832-884f-f359cd772eeef	PlaceOrder	Arium Vehicles
May 26, 2017 4:34:48 PM	09daaebb-5fae-4bcf-bf0f-c6d9cb837f2e	SetupDemo	Liam Grace

Timestamp	Transaction ID	Transaction Type	Transaction Submitter
May 26, 2017 4:48:48 PM	883faafb-72b7-4eb7-b90f-266d989351975	UpdateOrderStatus	Arium Vehicles
May 26, 2017 4:48:47 PM	eff0c940-f26a-4832-884f-f359cd772eeef	PlaceOrder	Arium Vehicles
May 26, 2017 4:34:48 PM	09daaebb-5fae-4bcf-bf0f-c6d9cb837f2e	SetupDemo	Liam Grace

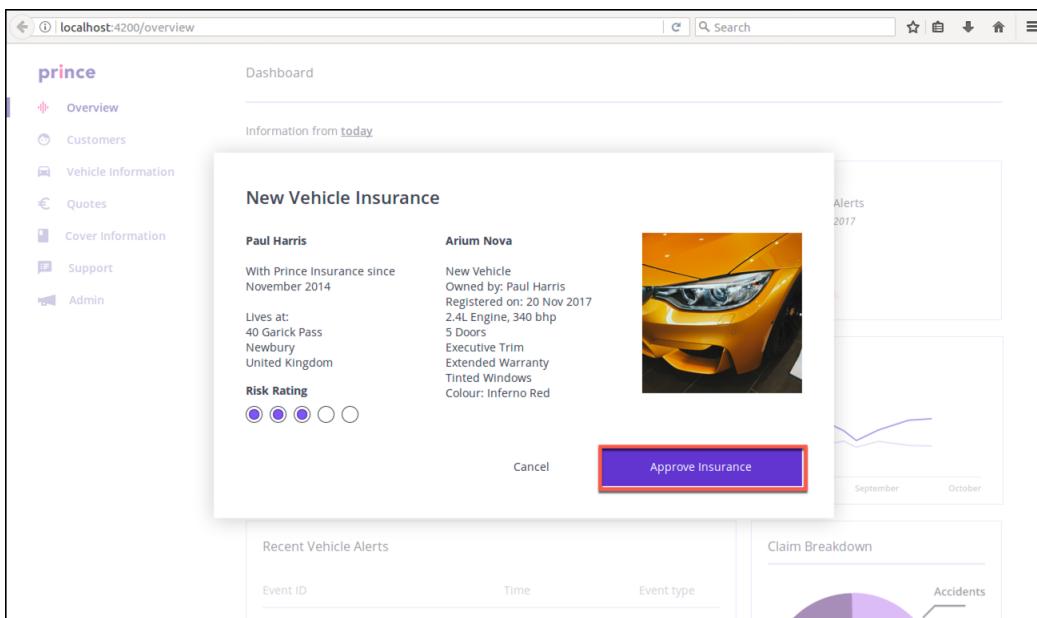
Timestamp	Transaction ID	Transaction Type	Transaction Submitter
May 26, 2017 4:53:53 PM	8a20da3e-4fb8-4bb9-a8a9-709518058537	UpdateOrderStatus	Arium Vehicles
May 26, 2017 4:53:49 PM	1d4cd676-4074-483a-91b1-c45d351fd633	UpdateOrderStatus	Arium Vehicles
May 26, 2017 4:53:43 PM	4f247ed-ded1-4d64-b0da-4daacca671a8	UpdateOrderStatus	Arium Vehicles
May 26, 2017 4:53:36 PM	e8a5af15-ac9b-4330-b49b-f2f47a8e45f7	UpdateOrderStatus	Arium Vehicles
May 26, 2017 4:48:48 PM	883faafb-72b7-4eb7-b90f-266d989351975	UpdateOrderStatus	Arium Vehicles
May 26, 2017 4:48:47 PM	eff0c940-f26a-4932-884f-f359cd772eeef	PlaceOrder	Arium Vehicles
May 26, 2017 4:34:48 PM	09daaebb-5fae-4bcf-bf0f-c6d9cb837f2e	SetupDemo	Liam Grace

### 1.3. Insure the Car

8. After the car has been delivered, switch back to the Ionic App view where the owner ordered the car (localhost:8100), scroll down to the bottom, and click the **INSURE ME** button. Click to **Allow Location Access** if a popup appears.



9. Switch to the Insurer view (localhost:4200/overview) and click the **Approve Insurance** button.



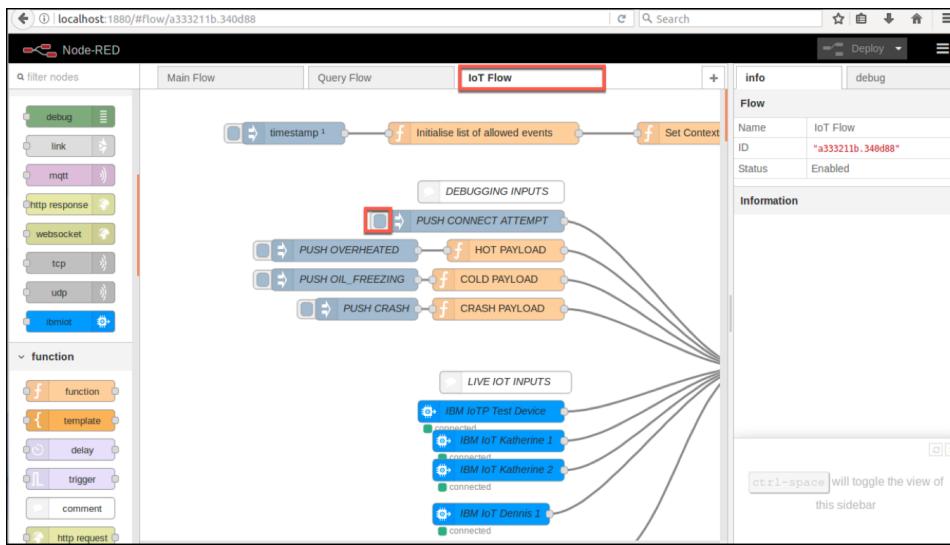
## 10. The owner is now insured by the insurance company.

The screenshot shows a web-based application interface for managing vehicle policy information. The URL in the address bar is `localhost:4200/policy/4f8d883e-4b7d-0a75-e903-429bebaa8549`. The left sidebar has a purple header "prince" and a navigation menu with links: Overview, Customers (which is selected and highlighted in purple), Vehicle Information, Quotes, Cover Information, Support, and Admin. The main content area displays details for a customer named Paul Harris, who is the owner of a vehicle named "Arium Nova".  
**Customer Details:**  
Owner of an **Arium Nova**  
With Prince Insurance since November 2014  
Insured on **20 Nov 2017**  
Lives at:  
40 Garick Pass  
Newbury  
United Kingdom  
**Vehicle Details:**  
**Arium Nova**  
New Vehicle  
Owned by: Paul Harris  
Registered on: 20 Nov 2017  
2.5L Engine 240bhp  
5 Doors  
Colour: Inferno Red  
**Sensor Test:**  
Click device button to test connection  
**Live Vehicle Information:**  
Acceleration: -,-,-,-  
Air Temperature: -,-,-,-  
Engine Temperature: -,-,-,-  
Light: -,-,-,-  
**Location:**

## 1.4. Internet of Things Integration

Once the car has been delivered and insured, we want to capture key vehicle events in the blockchain such as engine overheating, oil freezing, and accidents. The blockchain can easily integrate with the Watson IoT Platform using Node-Red Watson IBM IoT nodes to receive events from a device or send events from devices to the Watson IoT platform. We will use Node-Red Inject nodes to simulate receiving events from a device in the absence of having an actual physical sensor device.

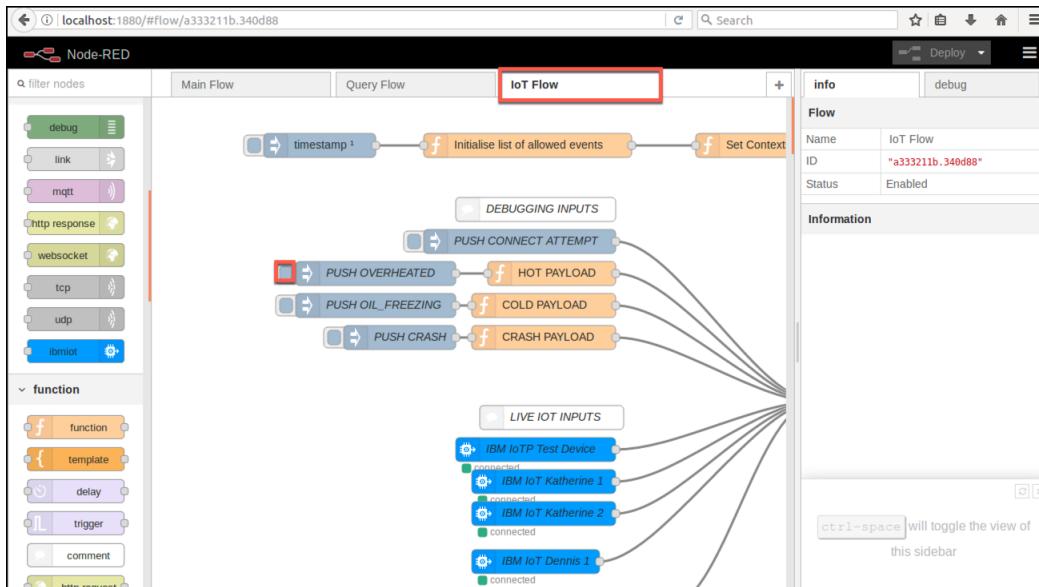
11. Switch to the Node-Red view tab (localhost:1880), select the **IoT Flow** tab, and click the button next to the **PUSH CONNECT ATTEMPT** node.



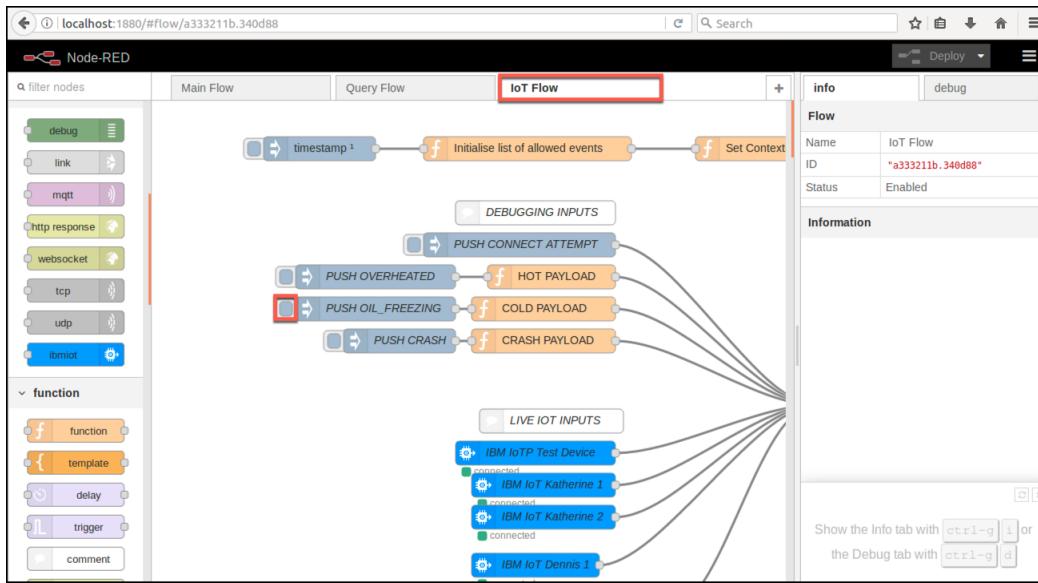
12. Switch to the Insurer view tab (localhost:4200), and notice under **Sensor Test** below that the vehicle sensor is now connected.

The screenshot shows the IBM Blockchain Insurer view tab at localhost:4200. The left sidebar has links for Overview, Customers (selected), Vehicle Information, Quotes, Cover Information, Support, and Admin. The main content area shows a customer profile for Paul Harris, owner of an Arium Nova. It details his ownership since November 2014, a new vehicle, and its registration. Below this is a section for Live Vehicle Information. A red box highlights the "Sensor Test" section, which shows "Device Connected ✓". Below this are four sensor readings: Acceleration (~.2~), Air Temperature (~.2~), Engine Temperature (~.2~), and Light (~.2~).

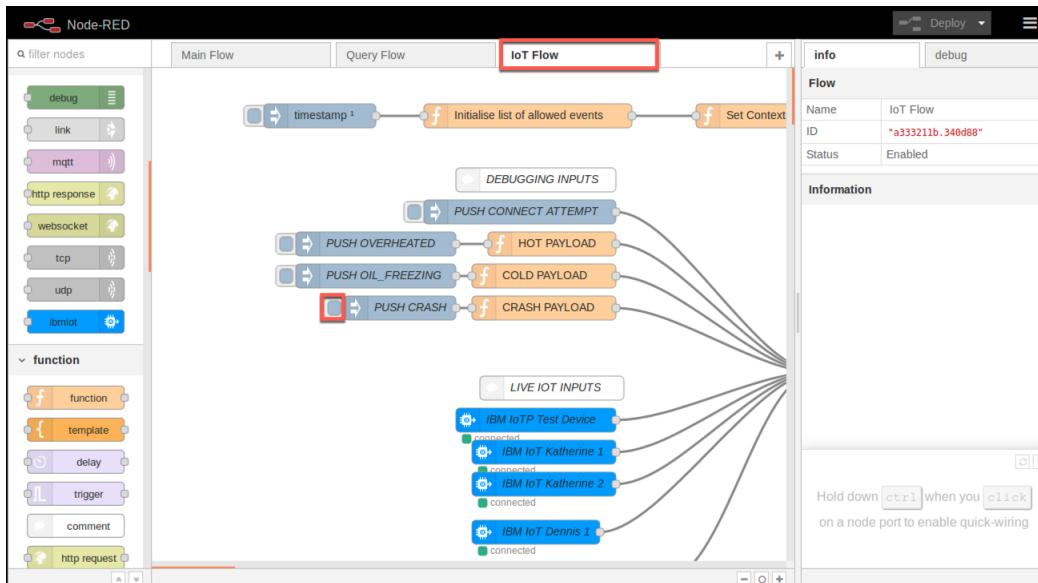
13. Switch back to the Node-Red view tab (localhost:4200), and click the button next to the **PUSH OVERHEATED** node. You should see a **successfully injected** message. We are simulating receiving an engine overheated event from a sensor.



14. Click the button next to the **PUSH OIL\_FREEZING** node. We are simulating receiving an engine oil freezing event from a sensor.



15. Click the button next to the **PUSH CRASH** node. We are simulating receiving a vehicle crashed event from a sensor.



## IBM Blockchain

16. Switch back to the Insurer view tab (localhost:4200). Note, that all three events (engine overheated, oil freezing, and crashed) show in this view. These events were captured in the blockchain and the insurer role has permission see all three events.

The screenshot shows the Prince application interface. On the left, a sidebar menu includes 'Overview', 'Customers' (selected), 'Vehicle Information', 'Quotes', 'Cover Information', 'Support', and 'Admin'. The main content area displays 'Paul Harris' customer details: Owner of an Arium Nova (With Prince Insurance since November 2014, Insured on 20 Nov 2017), vehicle information (Arium Nova, New Vehicle, 2.5L Engine 240bhp, 5 Doors, Colour: Inferno Red), and location (Lives at: 40 Garick Pass, Newbury, United Kingdom). Below this is a 'Live Vehicle Information' section with sensor test results (Device Connected ✓) and current values: Acceleration (1.02G), Air Temperature (24.91C), Engine Temperature (4C), and Light (476.96LUX). A map shows the location as Indiana State University. To the right is an 'Alert Stream' section titled 'Information history 6 months' with three alerts: 'CRASHED' (Event ID: 302b2669-05ec-cf4d-2ad3-dbb324b...), 'OIL FREEZING' (Event ID: 86450703-626c-c68e-2c1-b959f2a1...), and 'OVERHEATED' (Event ID: 232.78C). A 'See more' button is present under each alert.

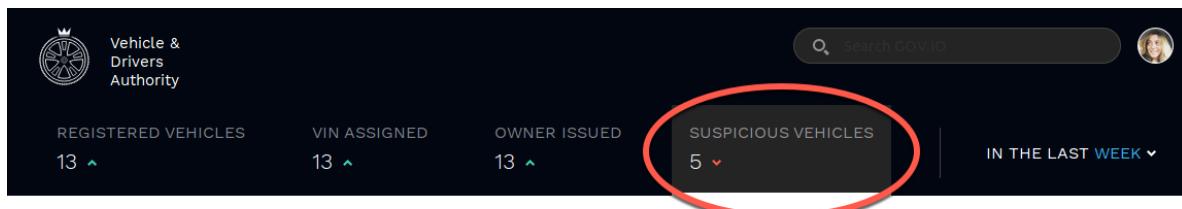
17. Switch back to the Manufactuer view tab (localhost:6002) and click the **Reports** link. The engine overheated events show in this view. These events were captured in the blockchain and the manufacturer role has permission see this type of event. The manufacturer wishes to detect trends in engine overheated failures in order to determine if a factory defect is causing this condition.

The screenshot shows the Arium Manufacture application interface. The left sidebar includes 'Main' (Dashboard, Reports selected, Goals, Groups), 'Production' (Events, Control Panel, Access, Alerts, Delivery), and 'Categories'. The main content area is titled 'POST-MANUFACTURE FAILURES' and displays a report for an 'OVERHEAT' failure on an 'Arium Nova' vehicle. The failure occurred on 11/20/2017 at 5:56:17 PM with S/N dd8abed390. The report details the failure type, occurrence date/time, and vehicle information. It also lists sensor data: Engine Temp. (101.00C), Acceleration (1.02G), Pitch (-0.01°), Air Temp. (24.91C), Roll (-0.05°), and Light (476.96LUX). An 'INSIGHT' section notes that it's the 2nd Arium Nova failure this week due to Overheating, with a common factor of Mileage < 100,000. To the right, a 'SIMILAR FAILURE' section shows another 'OVERHEAT' event for an 'Arium Nova' on 8/27/2017 at 11:49:05 AM with S/N 48c7d37e383, listing Engine Temp. (232.78C).

## 1.5. Regulator Analytics

The VDA (regulator) in this scenario has run additional analytics on the transactions on the blockchain to look for suspicious behaviour that the smart contract was not designed to prevent. This gives the ability to have the power of data analytics on top of the benefits of a blockchain such as a verifiably clean source of information to analyse among the most relevant for analytics.

18. In the Vehicle & Driver Authority app (<http://localhost:6001>) click the **SUSPICIOUS VEHICLES** tab near the top.



Vehicle	VIN	Notification
Ridge Cannon - White	312457645	Suspicious ownership sequence
Ridge Rancher - White	326548754	Uninsured vehicle
Morde Pluto - Green	564215468	Insurance write-off but still active
Morde Putt - Black	6437956437	Mileage anomaly
Ridge Cannon - Silver	65235647	Untaxed vehicle

Here we can see the analytics has found a number of vehicles with associated suspicious transactions that may warrant further investigation.

19. Click on the 'mileage anomaly' in the list of vehicles

This shows a list of transactions that are associated with this anomaly and In this instance, the mileage of the vehicle may not match with insurance records or possibly decreased from previous records.

Timestamp	Transaction	Car Owner	Contact Current Owner	X
Nov 3, 2017 6:27:51 PM	eb5be077-ffb5-0117-6b9e-150b6e0c257b	Anastasia		

20. In the Vehicle & Driver Authority app (<http://localhost:6001>) click the **ASSET ACTIVITY** tab near the top.

The screenshot shows the IBM Blockchain Vehicle & Driver Authority application interface. At the top, there's a navigation bar with icons for back, forward, search, and user profile. Below the navigation is a header with the title 'Vehicle & Drivers Authority' and a logo. The header also displays statistics: 'REGISTERED VEHICLES 15', 'VIN ASSIGNED 15', 'ASSET ACTIVITY 11' (which is highlighted with a red box), 'SUSPICIOUS VEHICLES 5', and a dropdown for 'IN THE LAST WEEK'. To the right of the header is a search bar and a user profile icon. The main content area is titled 'Asset Activity' and includes a 'Transaction Type' dropdown set to 'Vehicle'. It lists several transaction types with their corresponding hash IDs and status indicators (e.g., 'Vehicle', 'Insurer', 'Vehicle Owner').

Transaction Type	Hash ID	Status
Vehicle	Ceb2b5efba7a50ced0cc0e5a244ca759c51ld0a8d8b7ate3fee8263fda46b34a	Vehicle
Vehicle	4f88414beb3750be52ecdf396602eccdf7832aed0c50cc4a5240692b9dbd0183	Vehicle
Vehicle	Db01ca63864c5a57b810b3b6271d3c17d672ade73e2ad42cf75e94bf9fc2876b	Vehicle
Insurer	B478c642b2c089159cf42802e75c99392c6fa355adbc94ab4aeecc92467737f37	Vehicle Owner
Vehicle Owner	D58a764f6d4ecf4d2cbbb2d3864604fe0495b25a94f5d2c60b536d459b03c79	Vehicle

Here we can see a record of all transactions for the vehicles including all the Internet of Things events.

21. Cleanup the hyperledger fabric environment for the next lab. Perform the following at the command prompt in the VLD directory:

a. `./stopAll.sh`

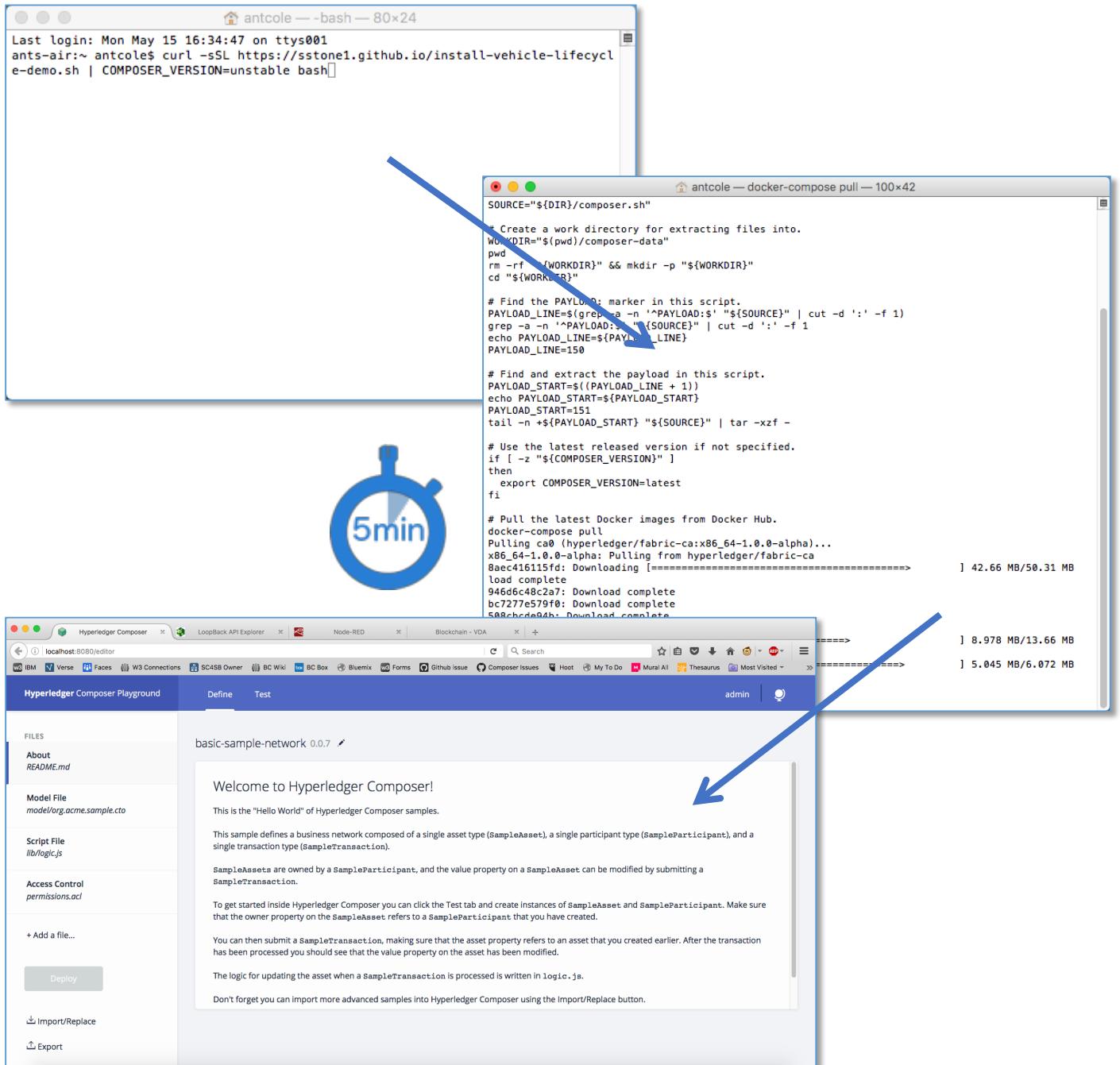
**Congratulations on completing the Vehicle Lifecycle Demo lab!**

## Appendix A. Starting the sample Application

On the VM, the samples have been pre-downloaded from Github and already started. If for any reason they have broken or need restarting, them follow these steps:

1. Open a terminal and input:

```
cd Documents/composer-sample-applications-master/packages/vehicle-lifecycle
./build.sh
cat installers/hlfv1/install.sh | bash
```

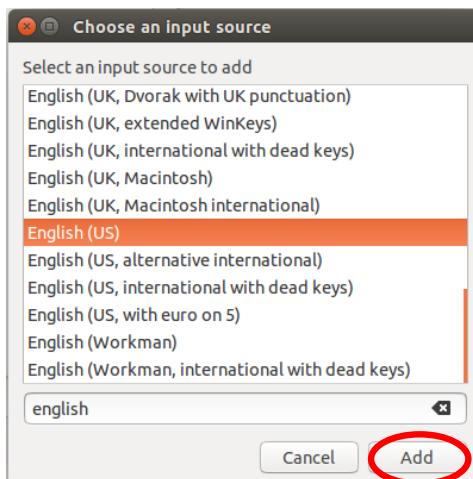
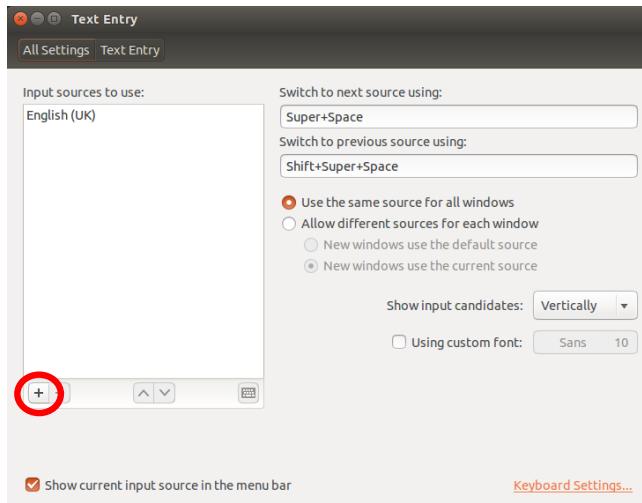


## Appendix B. Keyboard Language Change

To change the keyboard language to enable you to use foreign laptops follow these steps:

Click on the  icon in the top right & select **Text Entry Settings...**

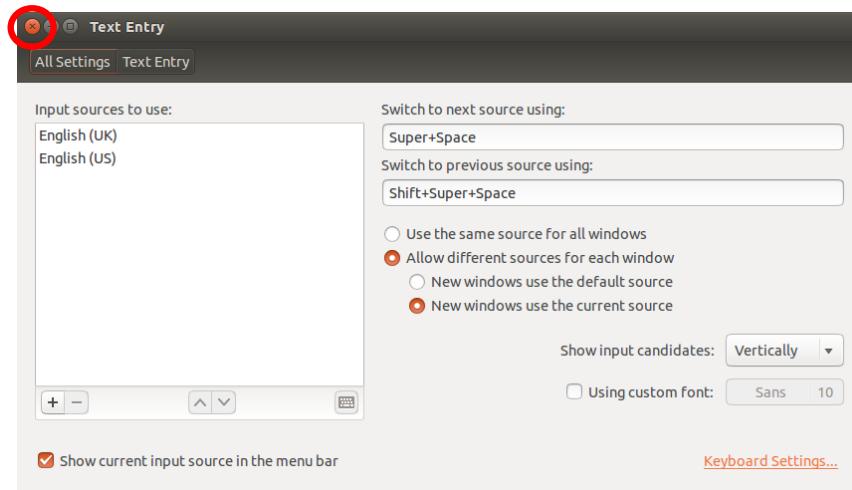
Select the  symbol



Type your **Language** (E.G. English) and then **country** (E.G. US)

Select the appropriate keyboard and click '**Add**'

**Close** the Settings box



Select the 'EN' in the top right of the screen and select your new keyboard



Your keyboard is now ready to use

---

## Appendix C. 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.

**COPYRIGHT LICENSE:**

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Appendix D. 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	
<i>System z</i>	<i>Tivoli</i>	<i>WebSphere</i>	<i>Workplace</i>	<i>System p</i>	

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.

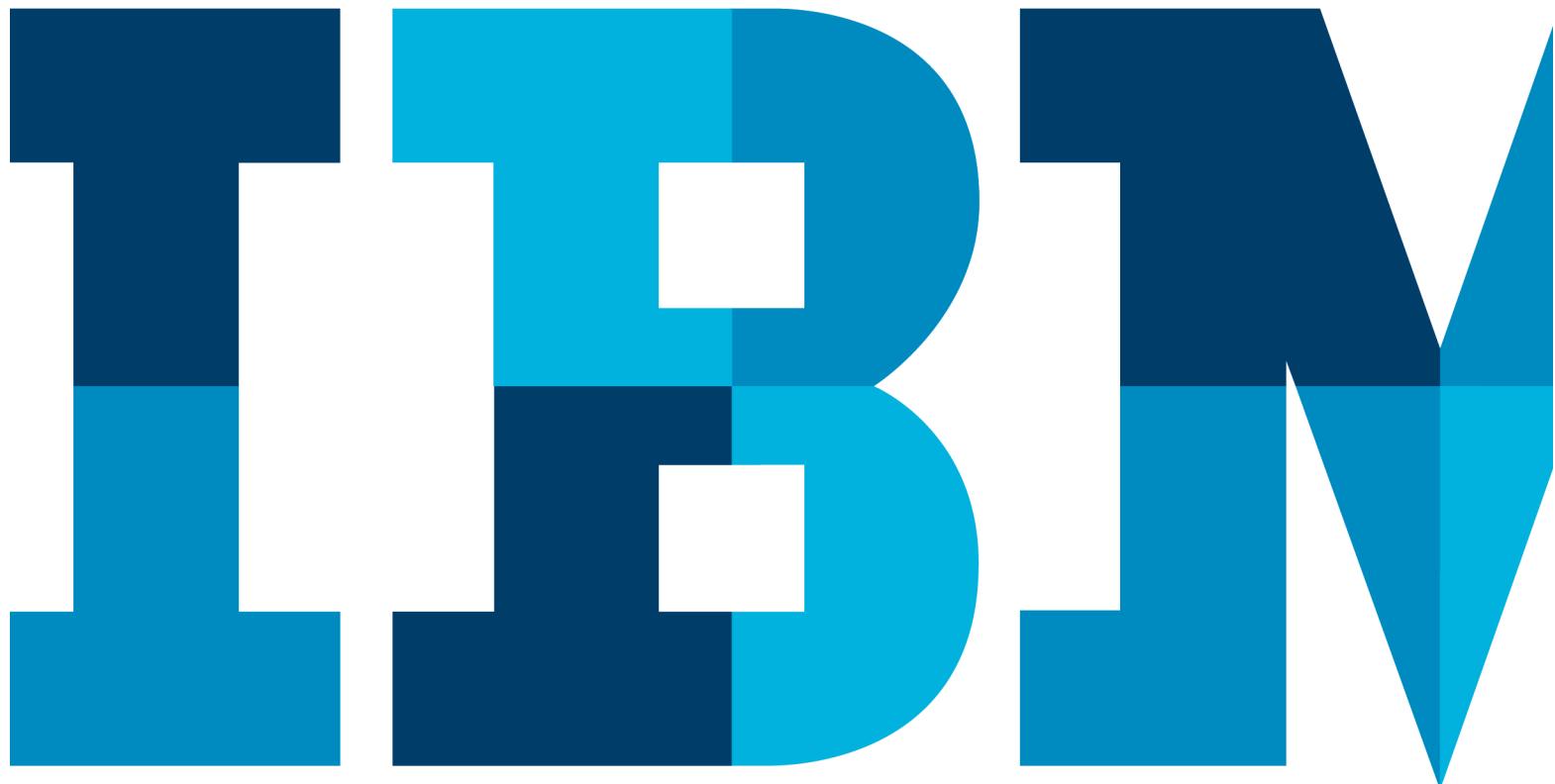
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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.

# IBM Blockchain Hands-On Hyperledger Composer Playground

*Lab Two – Exercises*



---

## Contents

<b>SECTION 1. USING HYPERLEDGER COMPOSER.....</b>	<b>3</b>
CAR AUCTION SAMPLE.....	4
1.1.1. OPEN THE PLAYGROUND .....	4
1.1.2. ADD THREE PARTICIPANTS .....	7
1.1.3. ADD AN ASSET .....	11
1.1.4. ADD A VEHICLE LISTING .....	13
1.1.5. SUBMIT OFFERS ON THE VEHICLE.....	15
1.1.6. CLOSING THE BIDDING .....	18
EXPLORE THE EDITOR VIEWS .....	20
1.1.7. MODEL FILE .....	20
1.1.8. TRANSACTION PROCESSORS .....	23
1.1.9. ACCESS CONTROL LIST.....	25
UPDATING THE MODEL (ADVANCED AND OPTIONAL) .....	26
EXPORT THE BUSINESS NETWORK ARCHIVE .....	26
<b>APPENDIX B. TRADEMARKS AND COPYRIGHTS .....</b>	<b>29</b>

## Introduction to this section of the lab

Skill requirements:

- There are no skill prerequisites to completing the first section called ‘Car Auction Sample’. It is desirable but not essential to have some background knowledge of JavaScript for the later section called ‘Explore the Editor Views’.

Technical pre-requisites:

- Internet Connection
- Web browser

This section of the lab takes place entirely in the web browser using Hyperledger Composer *Playground*.

Playground simulates the entire blockchain network within the browser by providing a sandpit environment to define, test and explore business networks defined using Composer. It is possible to connect to a live blockchain Hyperledger Fabric instance, or install the Composer Playground on a local machine for more developer friendly tools.

Hyperledger Composer Playground is one method to use Hyperledger Composer, other methods are also available at <https://hyperledger.github.io/composer/installing/installing-index.html>.

---

## Section 1. Using Hyperledger Composer

Hyperledger Composer (<https://hyperledger.github.io/composer>) is an open-source set of tools designed to make building blockchain applications easier.

It allows users to model the business networks, assets and transactions that are required for blockchain applications, and to implement those transactions using simple JavaScript functions. The blockchain applications run on instances of Linux Foundation Hyperledger Fabric ([www.hyperledger.org](http://www.hyperledger.org)).

The purpose of this lab is to introduce you to the concepts of a blockchain by showing you how a blockchain transfers assets between participants in a business network. We will use the implementation of a simple blind car auction as the scenario for the demo.

The car auction business network has a set of known participants (buyers and sellers), assets (cars and car listings) and transactions (placing bids and closing auctions). We will model these using Hyperledger Composer and test the business logic that makes the auction work.

Crucially, a blockchain could be used to bring together the buyers and sellers of these assets without needing any trusted third party. However, an auctioneer could be used to provide visibility and governance of the network if required.



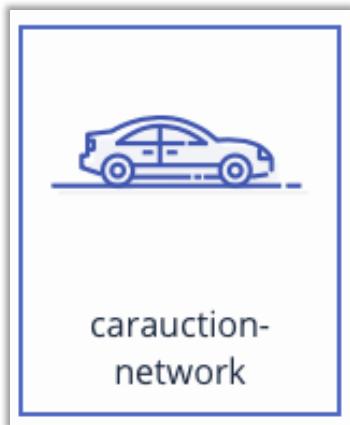
## Car Auction Sample

### 1.1.1. Open the Playground

1. Open a web browser and go to <http://composer-playground.mybluemix.net>. Dismiss the welcome screen to show the playground wallet screen which is used to connect and deploy new business networks:

The screenshot shows the Hyperledger Composer Playground interface. At the top, there is a blue header bar with the text "Hyperledger Composer Playground" on the left and a "Get local version" button with a download icon on the right. Below the header, the main content area has a light gray background. On the left, there is a sidebar with the title "My Business Networks" and the subtitle "Connection: Web Browser". Inside this sidebar, there is a box containing a globe icon with Japanese text "こんにちは" and an English "Hello, Composer!" message. Below this, there is a link to "Get started with the basic-sample-network, or view our [Playground tutorial](#)". Underneath, there is a section titled "BUSINESS NETWORK" with the name "basic-sample-network" and a "Get Started →" button. To the right of this sidebar, there is a large dashed rectangular box containing a plus sign icon and the text "Deploy a new business network". At the bottom of the page, there is a horizontal navigation bar with links for "Legal", "GitHub", "Playground v0.14.2", "Tutorial", "Docs", and "Community".

2. Click the “Deploy a business network” box. Then scroll down and select the carauction-network:



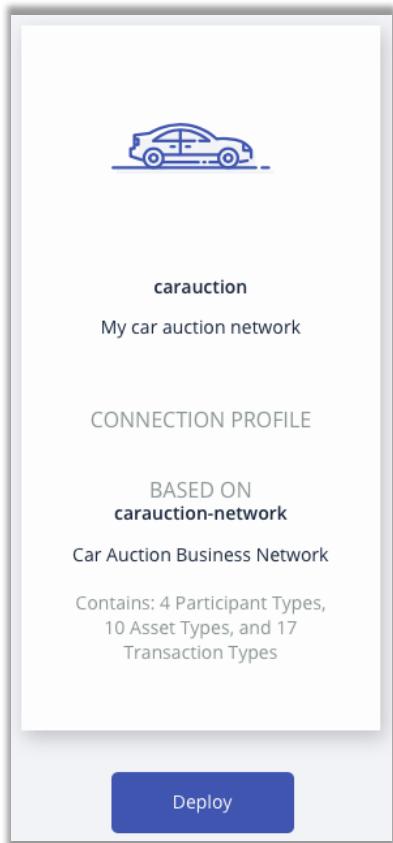
3. Next give the business network a name and description:

Deploy New Business Network

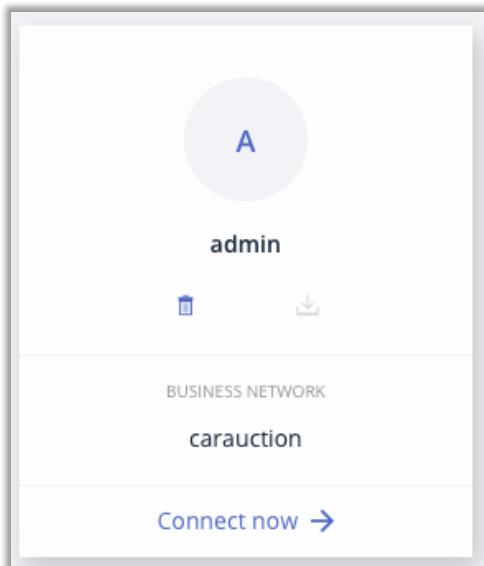
<b>1. BASIC INFORMATION</b>	Give your new Business Network a name:	carauction
	Describe what your Business Network will be used for:	My car auction network

4. Click the Deploy button to deploy the new car auction business network:





5. Click "Connect now" in the new identity card for the carauction network:



6. Take a few minutes to read through the description of the car auction sample, to help understand the participants, assets and transactions associated with this particular network.

The screenshot shows the Hyperledger Composer Playground interface. On the left, there's a sidebar titled 'FILES' with options like 'About', 'Model File', 'Script File', 'Access Control', and '+ Add a file...'. A 'Deploy' button is also present. The main area is titled 'carauction-network 0.1.5' and contains a description of the 'Car Auction Network'. It lists participants ('Member', 'Auctioneer'), assets ('Vehicle', 'VehicleListing'), and transactions ('Offer', 'CloseBidding'). Below this, there's a note about the logic for 'makeOffer' and 'closeBidding' functions. At the bottom, instructions for testing and creating participants are provided.

### 1.1.2. Add Three Participants

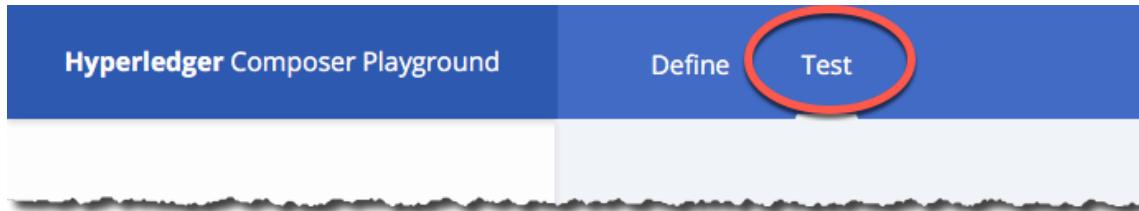
In the next section we will now work with the deployed car auction blockchain network.

We will first instantiate three *Member* participants of the car auction business network:

- Alice Smith (alice@email.com), who will make a bid on a car,
- Bob Jones (bob@email.com), who will also make a bid on a car, and
- Charlie Brown (charlie@email.com), who currently owns a car.

We will not instantiate an *Auctioneer* in this demo; this could be used in order to provide oversight of the network, although is not necessary.

7. Click the **Test** tab and then click on the *Member* participant registry:



The registry is empty as no members have currently been defined.

8. Click on **Member** to view there are no current members in the environment

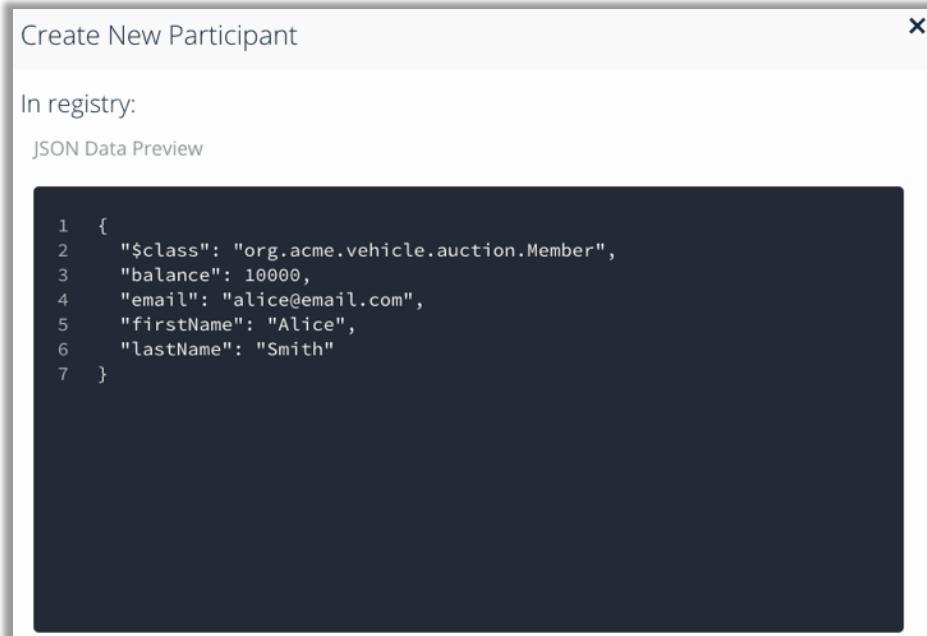




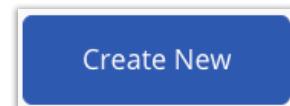
9. Click **Create New Participant** to add a new Member.



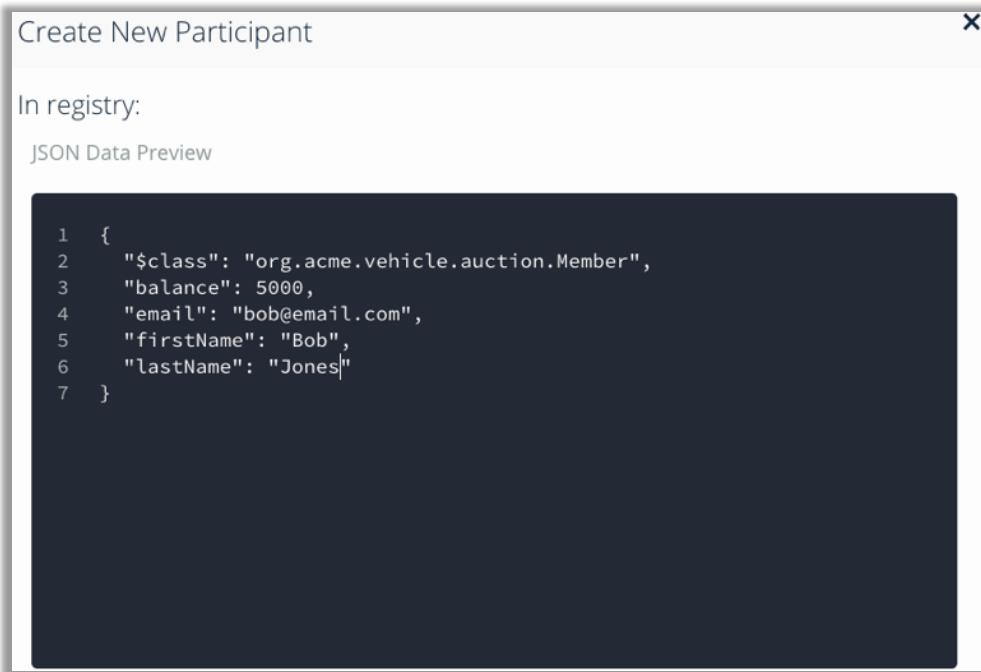
10. Type the correct values into the JSON data structure to add Alice to the business network. Let's give her a starting balance of 10000.



11. Click **Create New** to add Alice to the registry.



12. Do the same for Bob. Let's give him a starting balance of 5000.



13. Finally do the same for Charlie. He hasn't got so much money (he's selling his car, after all) so let's give him a starting balance of 100.

Create New Participant X

In registry:

JSON Data Preview

```

1  {
2    "$class": "org.acme.vehicle.auction.Member",
3    "balance": 100,
4    "email": "charlie@email.com",
5    "firstName": "Charlie",
6    "lastName": "Brown"
7  }

```

14. Verify that all participants in the business network have been correctly defined. Use the appropriate Edit button (✎) to make any changes.

Participant registry for org.acme.vehicle.auction.Member	
ID	Data
alice@email.com	<pre>{   "\$class": "org.acme.vehicle.auction.Member",   "balance": 10000,   "email": "alice@email.com",   "firstName": "Alice",   "lastName": "Smith" }</pre> <div style="text-align: right; margin-right: 10px;"> <span style="border: 1px solid #ccc; padding: 2px 5px;">Edit</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">Delete</span> </div> <div style="text-align: right; margin-top: -10px;"> <span>Show All</span> </div>
bob@email.com	<pre>{   "\$class": "org.acme.vehicle.auction.Member",   "balance": 5000,   "email": "bob@email.com",   "firstName": "Bob",   "lastName": "Jones" }</pre> <div style="text-align: right; margin-right: 10px;"> <span style="border: 1px solid #ccc; padding: 2px 5px;">Edit</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">Delete</span> </div> <div style="text-align: right; margin-top: -10px;"> <span>Show All</span> </div>
charlie@email.com	<pre>{   "\$class": "org.acme.vehicle.auction.Member",   "balance": 100,   "email": "charlie@email.com",   "firstName": "Charlie",   "lastName": "Brown" }</pre> <div style="text-align: right; margin-right: 10px;"> <span style="border: 1px solid #ccc; padding: 2px 5px;">Edit</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">Delete</span> </div> <div style="text-align: right; margin-top: -10px;"> <span>Show All</span> </div>

### 1.1.3. Add an Asset

We will now add Charlie's car to the Vehicle Asset registry.

15. Click the **Vehicle** asset registry.



16. This registry contains no assets currently. Click the **Create New Asset** button to add a new asset.

17. Instantiate the car by adding a vehicle identification number (VIN) of 1234 and assign it to Charlie by filling in the JSON object as follows. (We use his email address to identify him; this was specified as the key field in the User definition using the 'identified by' statement.)

A screenshot of a modal dialog box titled "Create New Asset". The title bar has a close button (X) on the right. Inside the dialog, there is a header "In registry:" followed by a "JSON Data Preview" section. Below this is a code editor area containing the following JSON code:

```
1  {
2    "$class": "org.acme.vehicle.auction.Vehicle",
3    "vin": "1234",
4    "owner":
5      "resource:org.acme.vehicle.auction.Member#charlie@email.com"
```

The code editor has a dark background with light-colored text. The entire dialog has a light gray background.

18. Click **Create New** to add the new vehicle to the registry.



Create New

## 19. View your newly added asset in the registry.

Asset registry for org.acme.vehicle.auction.Vehicle

+ Create New Asset

ID	Data
1234	{ "\$class": "org.acme.vehicle.auction.Vehicle", "vin": "1234", "owner": "resource:org.acme.vehicle.auction.Member#charlie@email.com" }

#### 1.1.4. Add a Vehicle Listing

In this section we will put the car up for sale by creating a *VehicleListing* instance.

20. Click the **VehicleListing** asset registry. Once more, the VehicleListing registry should be empty.



21. Click the **Create New Asset** button to add the asset.

22. Update the fields and remove the random offers to show the below. Syntactic validation of the object occurs at this point, so correct any errors if necessary.

A screenshot of a modal dialog box titled "Create New Asset". Inside the dialog, there is a section labeled "In registry:" followed by a "JSON Data Preview" area. The preview contains the following JSON code:

```
1  {
2    "$class": "org.acme.vehicle.auction.VehicleListing",
3    "listingId": "listing1",
4    "reservePrice": 500,
5    "description": "One careful owner",
6    "state": "FOR_SALE",
7    "vehicle": "resource:org.acme.vehicle.auction.Vehicle#1234"
8 }
```

The dialog has a close button in the top right corner.

23. Click **Create New** to add the new vehicle listing to the registry.

A screenshot of a blue rectangular button with the text "Create New" in white, centered within the button's area.



## 24. View the listing in the registry.

Asset registry for org.acme.vehicle.auction.VehicleListing

+ Create New Asset

ID	Data
listing1	<pre>{     "\$class": "org.acme.vehicle.auction.VehicleListing",     "listingId": "listing1",     "reservePrice": 500,     "description": "One careful owner",     "state": "FOR_SALE",     "vehicle": "resource:org.acme.vehicle.auction.Vehicle#1234" }</pre> <p><a href="#">Edit</a> <a href="#">Delete</a></p> <p><a href="#">Collapse</a></p>

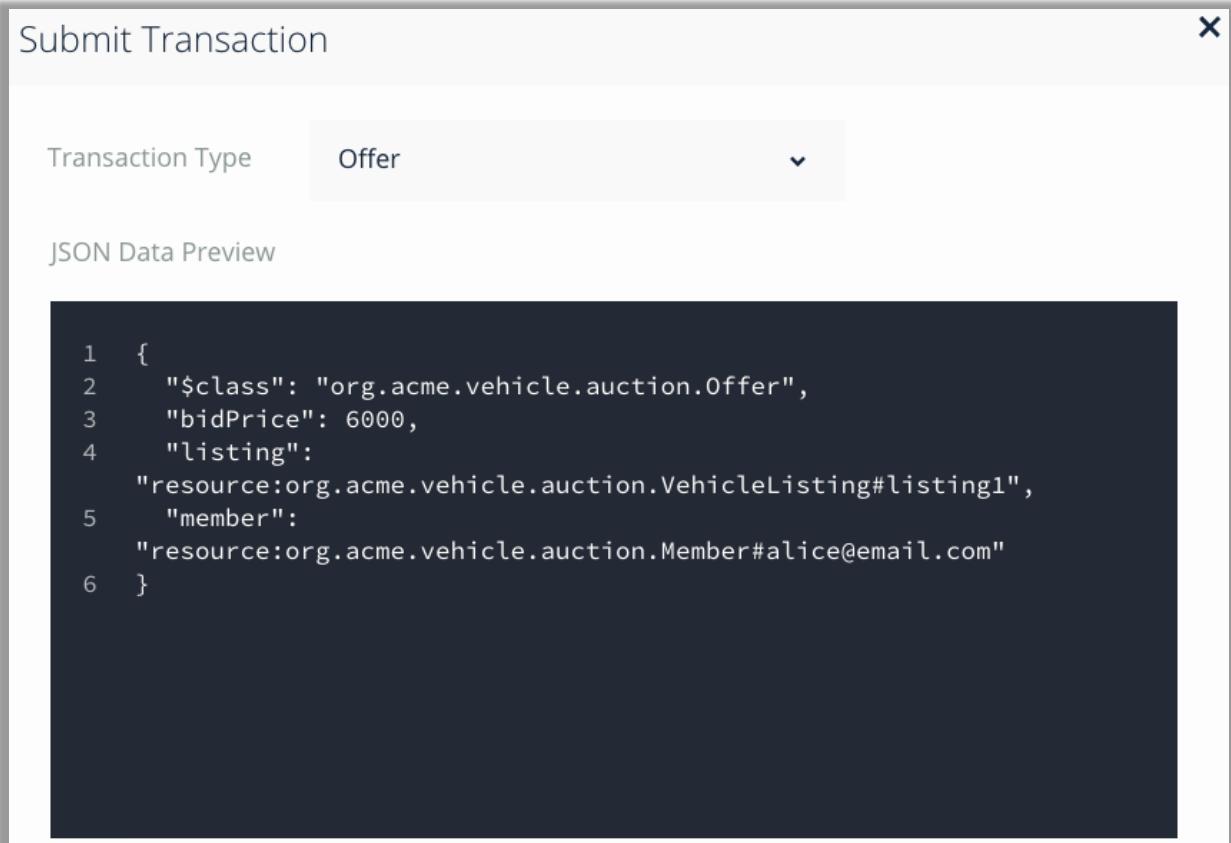
### 1.1.5. Submit offers on the vehicle

We will now let Alice and Bob bid on the vehicle.

25. Click on the Submit Transaction button

A blue rectangular button with the text "Submit Transaction" in white, centered within a rounded rectangle.

26. Let Alice put in a bid of 6000.

A screenshot of a "Submit Transaction" dialog box. At the top left is the title "Submit Transaction" and at the top right is a close button (an "X"). Below the title is a dropdown menu labeled "Transaction Type" with "Offer" selected. To the right of the dropdown is a small dropdown arrow. Below the dropdown is a section titled "JSON Data Preview" containing a block of JSON code. The JSON code is:

```
1  {  
2      "$class": "org.acme.vehicle.auction.Offer",  
3      "bidPrice": 6000,  
4      "listing":  
5          "resource:org.acme.vehicle.auction.VehicleListing#listing1",  
6          "member":  
7              "resource:org.acme.vehicle.auction.Member#alice@email.com"  
8  }
```

27. Click **Submit** to submit the offer transaction.

A blue rectangular button with the word "Submit" in white, centered within a rounded rectangle.

28. See the transaction successful appear in the Historian registry. Switch to view all transactions by clicking 'All Transactions':

### All Transactions

29. You will also notice additional transactions for creating participants and assets. Click "view record" for more information.

Date, Time	Entry Type	Participant	
2017-12-04, 17:37:55	Offer	admin (NetworkAdmin)	<a href="#">view record</a>

30. Let Bob put in a bid of 4000.

Submit Transaction X

Transaction Type Offer ▼

JSON Data Preview

```
1  {
2    "$class": "org.acme.vehicle.auction.Offer",
3    "bidPrice": 4000,
4    "listing":
5      "resource:org.acme.vehicle.auction.VehicleListing#listing1",
6      "member":
7        "resource:org.acme.vehicle.auction.Member#bob@email.com"
8  }
```

31. Verify the transactions in the registry.

Date, Time	Entry Type	Participant	
2017-12-04, 17:43:19	Offer	admin (NetworkAdmin)	<a href="#">view record</a>
2017-12-04, 17:37:55	Offer	admin (NetworkAdmin)	<a href="#">view record</a>

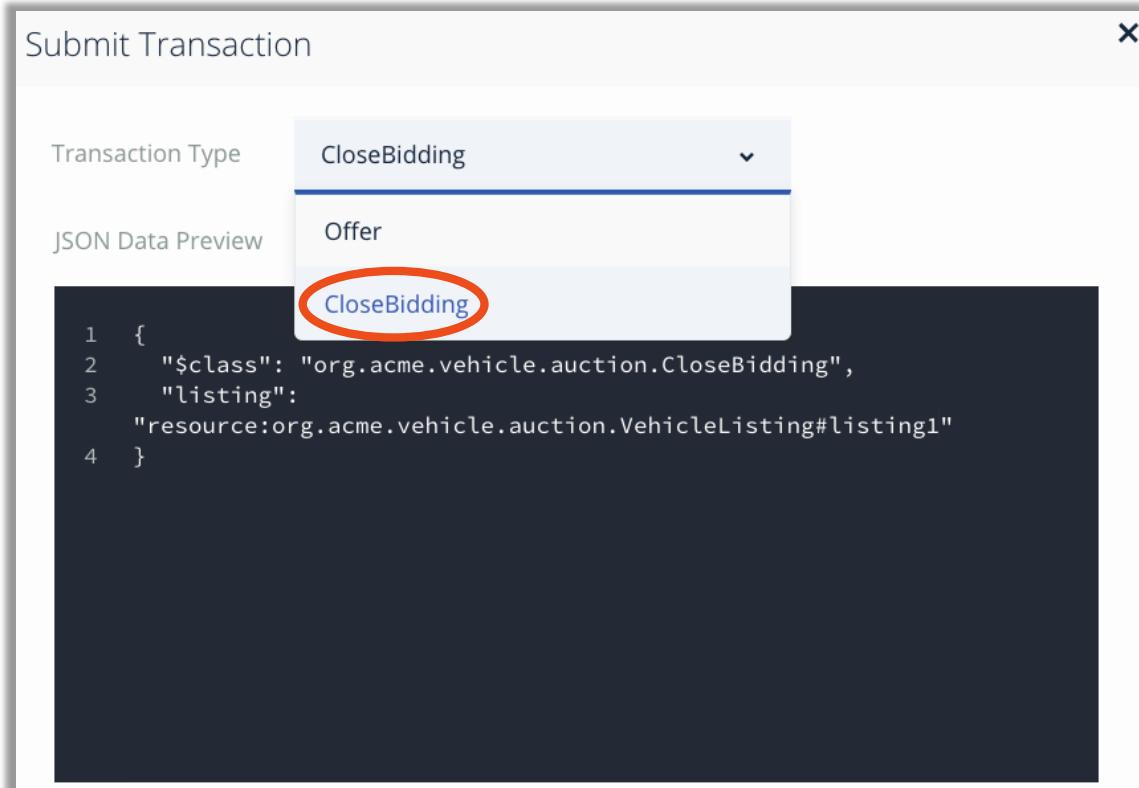
Note that the transactions cannot be edited or individually deleted once submitted; this is one of the defining characteristics of a blockchain.



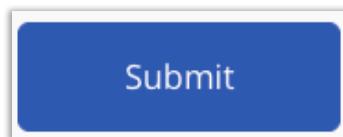
### 1.1.6. Closing the bidding

To close the bidding on the listing we need to submit a *CloseBidding* transaction.

32. Submit a new transaction, this time selecting **CloseBidding** from the drop-down 'Transaction Type' field.



33. Click **Submit** to submit the CloseBidding transaction.



34. Verify that the transaction has been added to the blockchain transaction registry. Click on 'view record' to see the content of the transaction.

Date, Time	Entry Type	Participant	
2017-12-04, 17:46:00	CloseBidding	admin (NetworkAdmin)	<a href="#">view record</a>

Historian Record

Transaction    Events (0)

```
1  {
2    "$class": "org.acme.vehicle.auction.CloseBidding",
3    "listing": "resource:org.acme.vehicle.auction.VehicleListing#listing1",
4    "transactionId": "a3f7d0c0-f78b-4238-86cd-17e03fec9ea",
5    "timestamp": "2017-12-04T17:46:00.208Z"
6 }
```

Based on the bids we submitted, Alice should now be the owner as she put in the highest bid. We should also be able to verify that the owner of the car has changed and appropriate balances increased or decreased accordingly.

35. Go to the **Vehicle** asset registry to see the vehicle owner has been updated to Alice.



36. You will see the following vehicle owned by Alice in the vehicle registry.

Asset registry for org.acme.vehicle.auction.Vehicle

+ Create New Asset

ID	Data
1234	{ "\$class": "org.acme.vehicle.auction.Vehicle", "vin": "1234", "owner": "resource:org.acme.vehicle.auction.Member#alice@email.com" }



37. Go to the **Member** asset registry to see that Charlie's balance has increased by the winning bid amount, and that Alice's balance has decreased by the same.

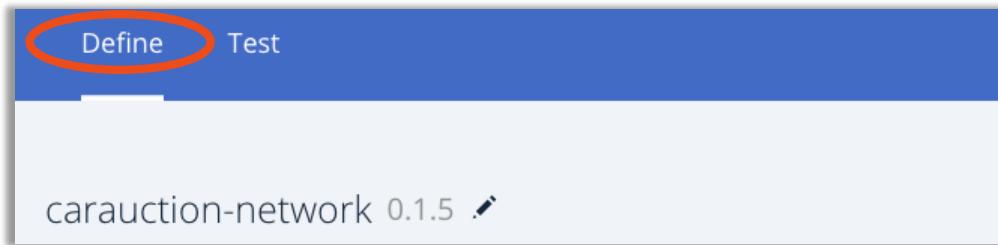
ID	Data
alice@email.com	{           "\$class": "org.acme.vehicle.auction.Member",           "balance": 4000,           "email": "alice@email.com",           "firstName": "Alice",           "lastName": "Smith"         }
bob@email.com	{           "\$class": "org.acme.vehicle.auction.Member",           "balance": 5000,           "email": "bob@email.com",           "firstName": "Bob",           "lastName": "Jones"         }
charlie@email.com	{           "\$class": "org.acme.vehicle.auction.Member",           "balance": 6100,           "email": "charlie@email.com",           "firstName": "Charlie",           "lastName": "Brown"         }

Congratulations! You have completed the first part of this lab.

## Explore the Editor Views

### 1.1.7. Model File

38. Click on the define tab to go back to the main playground window.



39. Click the Model File (models/auction.cto) to open it.



This .cto file models the assets, participants and transactions for this blockchain application.



40. Look at the Vehicle asset:

```
asset Vehicle identified by vin {
    o String vin
    --> Member owner
}
```

This uses the Hyperledger Composer Modeling Language which will be looked at more later. An *asset* is anything of worth that will be transferred around the blockchain. Here we can see the asset class is called ‘*Vehicle*’ and will have an associated *vin* and a reference (indicated by “*-->*”) to a ‘*Member*’ participant that we will call ‘*owner*’.

41. Type and add some characters in an appropriate point to show the live validation of the model.

```
asset VehicleListing identified by listingId {
    o String listingId
    o Double reservePrice
    o String description
    o ListingState state
    o Offer[] offers optional
    --> Vehicle vehicle
}
```

 Error found!

Error: Syntax error in file undefined. Expected "extends", "identified by", "{", comment, end of line or whitespace but "i" found. Line 17 column 22

42. Scroll down and look at the abstract ‘*User*’ participant.

The participant will be the people or companies within the business network. Each *User* participant will be defined as having a *email*, *firstName* and *lastName*. As the class is **abstract** instances of it cannot be created; instances are instead implemented by the *Member* and *Auctioneer* classes.

```
abstract participant User identified by email {  
    o String email  
    o String firstName  
    o String lastName  
}  
  
participant Member extends User {  
    o Double balance  
}  
  
participant Auctioneer extends User {  
}
```

Here the user can become a *Member* requiring a *balance*, or an *Auctioneer* that does not.

43. Look at the *Offer* and *CloseBidding* transaction definitions:

```
transaction Offer {  
    o Double bidPrice  
    --> VehicleListing listing  
    --> Member member  
}  
  
transaction CloseBidding {  
    --> VehicleListing listing  
}
```

The *transaction* definitions give a description of the transactions that can be performed on the blockchain. They are implemented in a Transaction Processor file using the Javascript language.

#### 1.1.8. Transaction Processors

44. Click on the lib/logic.js file:



45. Scroll to **the bottom of the file** to review the logic used to make an offer on a car being auctioned:

```
/**  
 * Make an Offer for a VehicleListing  
 * @param {org.acme.vehicle.auction.Offer} offer - the offer  
 * @transaction  
 */  
function makeOffer(offer) {  
    var listing = offer.listing;  
    if (listing.state !== 'FOR_SALE') {  
        throw new Error('Listing is not FOR SALE');  
    }  
    if (listing.offers == null) {  
        listing.offers = [];  
    }  
    listing.offers.push(offer);  
    return getAssetRegistry('org.acme.vehicle.auction.VehicleListing')  
        .then(function(vehicleListingRegistry) {  
            // save the vehicle listing  
            return vehicleListingRegistry.update(listing);  
        });  
}
```

This implements the *makeOffer* function, which is executed when the *Offer* transaction is invoked on the blockchain. (It is the **@param** comment above the function that links the full transaction name as defined by the model to the Javascript method that implements it.)

Other Interesting areas of the function implementation include:

- The logic that the vehicle must be for sale to submit an offer on it
- The retrieval and update of the asset registry a few lines later
- Saving the updated asset back to the registry

### 1.1.9. Access Control List

The final file that defines the blockchain application is the Access Control List, which describes the rules which govern which participants in the business network can work with which parts of the blockchain.

46. Click the permissions.acl file:



47. Look at the ACL rules defined:

```
/***
 * Access Control List for the auction network.
 */
rule Auctioneer {
    description: "Allow the auctioneer full access"
    participant: "org.acme.vehicle.auction.Auctioneer"
    operation: ALL
    resource: "org.acme.vehicle.auction.*"
    action: ALLOW
}

rule Member {
    description: "Allow the member read access"
    participant: "org.acme.vehicle.auction.Member"
    operation: READ
    resource: "org.acme.vehicle.auction.*"
    action: ALLOW
}

rule VehicleOwner {
    description: "Allow the owner of a vehicle total access"
    participant(m): "org.acme.vehicle.auction.Member"
```

The rule allows or denies users to access aspects of the blockchain.



## Updating the Model (Advanced and Optional)

48. Try updating the model (*auction.cto*) for the *Vehicle* asset definition to include manufacturer make and model fields. Add in new *String* fields and click 'Deploy' to make the changes live.

Note that when you update the model, the syntax of any existing assets in the registry must be compatible with the new model. Use either the **optional** or **default="..."** qualifiers next to the new fields. If you make incompatible changes, you must first reset the demo.

Once you've deployed the changes, try adding new *Vehicle* assets to the registry to test the changes.

For more information on the Hyperledger Composer modelling language please refer to:  
[https://hyperledger.github.io/composer/reference/cto\\_language.html](https://hyperledger.github.io/composer/reference/cto_language.html)

## Export the Business Network Archive

49. Exporting to a Business Network Archive will save the Read Me, Model File(s), Script File(s) and Access Control rules that can be easily imported to a local developer environment, handed to a network operator to deploy to a live network or saved as a backup. More details on local installation at <https://hyperledger.github.io/composer/installing/installing-index.html>.



Congratulations! You have completed this lab.

---

## Appendix A. 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



## IBM Blockchain

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.

### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Appendix B. 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	
<i>System z</i>	<i>Tivoli</i>	<i>WebSphere</i>	<i>Workplace</i>	<i>System p</i>	

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**

## **NOTES**



---

© Copyright IBM Corporation 2014.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is

## **NOTES**

available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).



## **NOTES**





---

© Copyright IBM Corporation 2014.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).



Please Recycle

---

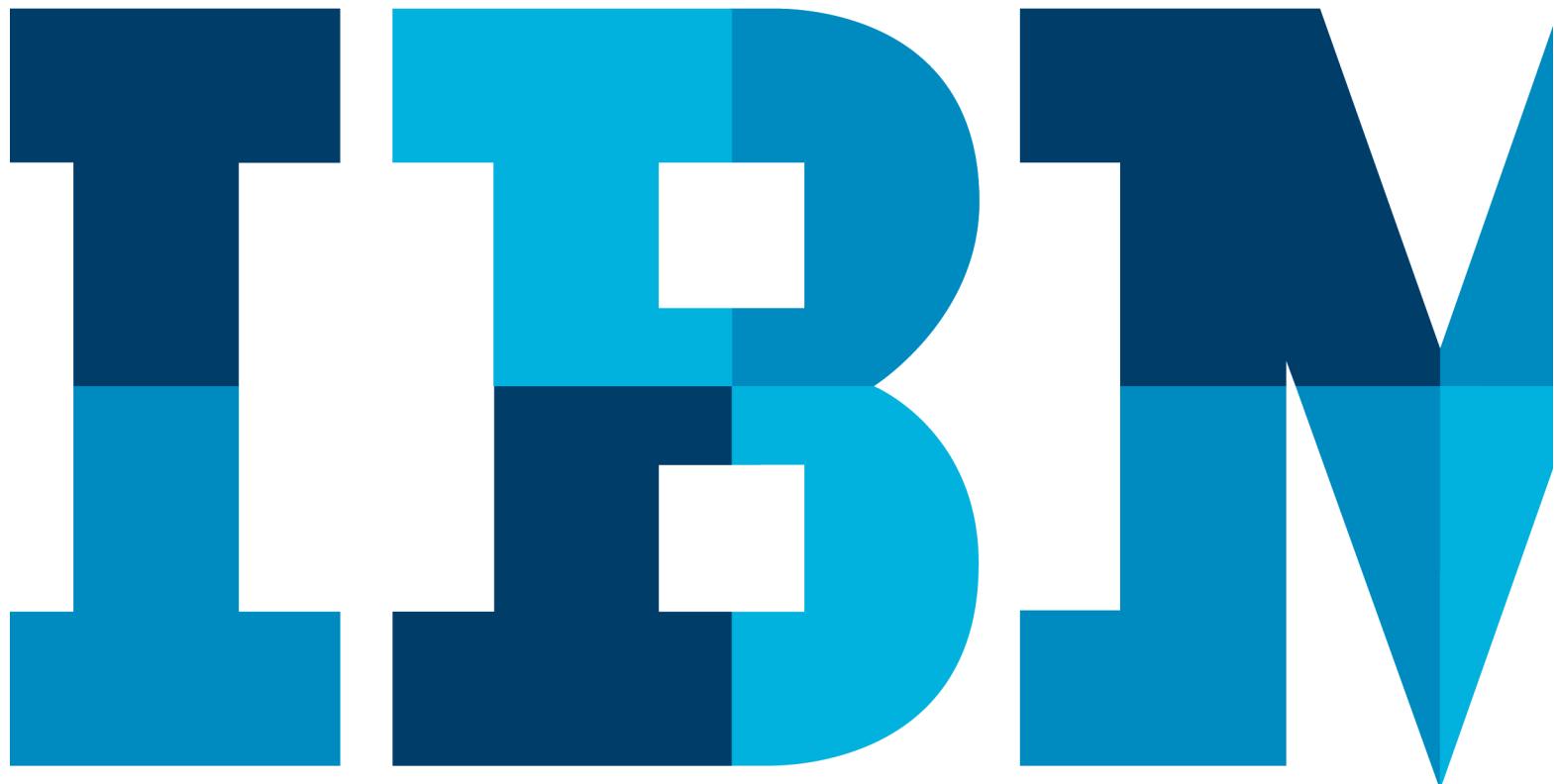
## **NOTES**

---

## **NOTES**

# IBM Blockchain Hands-On Hyperledger Composer - Developer Lab

*Lab Three VM Exercises Guide*





---

## Overview

The purpose of this lab is to introduce you to the Hyperledger Composer development environment. It is intended to be run on any machine that can meet the Hyperledger Composer specification.

Operating Systems: Ubuntu Linux 14.04 / 16.04 LTS (both 64-bit), or Mac OS 10.12

Docker Engine: Version 17.03 or higher

Docker-Compose: Version 1.8 or higher

Node: 6.x (note version 7 is not supported)

npm: 3.10.x

git: 2.9.x

A code editor of your choice, we recommend VSCode.

## Where to start?

**Section 1** will lead you through the installation instructions for Hyperledger Composer and Hyperledger Fabric.

**Section 2** will lead you through the creation, deployment and testing of a sample business network application. It will also show you how to generate a REST interface.

If you are running on a machine that has not been configured for Hyperledger Composer (for example, your personal laptop), then install the pre-requisites above and then start with Section 1.

If you are running on a machine that is provided for you as part of a classroom environment, your instructor will tell you where to begin this lab.

## Section 1: Installing Hyperledger Composer Development Tools

The master copy of the instructions for this section are online. It is recommended that you use the online version where possible, as this may contain updates to the instructions.

1. **Optionally just read thru the material on the Web page below for your own reference. These steps have already been performed for you on the VMWare image.** Bring up a web browser and navigate to the following page:

<https://hyperledger.github.io/composer/installing/development-tools.html>

The screenshot shows a web browser window with the title 'Installing a development en...' and the URL 'https://hyperledger.github.io/composer/installing/development-tools.html'. The page content is titled 'Installing and developing with Hyperledger Composer' and provides instructions for setting up a development environment. It includes sections for prerequisites, command-line instructions for Ubuntu, and shell commands for downloading prerequisites. A sidebar on the left lists navigation links such as Tutorials, Docs, Community, and specific sections for Installing, Try Online Playground, Install Local Playground, and Install a Development Environment.

2. Read thru the material in the link above ***but do not perform any of the steps.***
3. Once you have been able to successfully start the fabric and create a composer profile, you will have completed this section. Run the following shell commands and scripts to ensure the containers are started and the environment is ready for you:
  - **cd ~/fabric-tools**
  - **./teardownFabric.sh**
  - **./startFabric.sh**
  - **./createPeerAdminCard.sh**

## Section 2: Developer and Queries Tutorial

The master copy of the instructions for this section is online. The online site allows you to more easily copy and paste snippets of text, which is necessary for some of the steps.

4. Bring up a web browser and navigate to the following page:

<https://hyperledger.github.io/composer/tutorials/developer-tutorial.html>

### Developer Tutorial for creating a Hyperledger Composer solution

This tutorial will walk you through building a Hyperledger Composer blockchain solution from scratch. In the space of a few hours you will be able to go from an idea for a disruptive blockchain innovation, to executing transactions against a real Hyperledger Fabric blockchain network and generating/running a sample Angular 2 application that interacts with a blockchain network.

This tutorial gives an overview of the techniques and resources available to apply to your own use case.

*Note:* This tutorial was written against the latest Hyperledger Composer build on Ubuntu Linux running with Hyperledger Fabric v1.0 where referenced below and also tested for a Mac environment.

#### Prerequisites

Before beginning this tutorial:

- [Set up your development environment](#)
- Install an editor e.g. VSCode or Atom

#### Step One: Creating a business network structure

The key concept for is the **business network definition (BND)**. It defines the data model, transaction logic and access control rules for your blockchain solution. To create a BND, we need to create a suitable project structure on disk.

Follow the instructions contained within this page starting at the **Create a business network structure** section. Ensure your terminal window is open to the **/home/blockchain** directory as all git clone commands will clone projects from Github in that directory. There is no need to install Hyperledger Composer or the VSCode Editor since the VMWare image has this installed for you.

5. Bring up a web browser and navigate to the following page:

<https://hyperledger.github.io/composer/tutorials/queries.html>

Follow the instructions contained within this page. Once you have been able to successfully validate you only have one commodity, you will have completed this section. Be sure to complete step 6 below to cleanup the environment before moving on to the next lab.

6. Cleanup the Hyperledger Fabric environment for the next lab. Perform the following steps:

- **Enter control-C at the terminal window where the composer-rest-server is running to stop the server.**
- `cd ~/fabric-tools`
- `./stopFabric.sh`
- `./teardownFabric.sh`

---

## Appendix A. 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.

**COPYRIGHT LICENSE:**

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Appendix B. 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	
<i>System z</i>	<i>Tivoli</i>	<i>WebSphere</i>	<i>Workplace</i>	<i>System p</i>	

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**

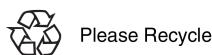


---

© Copyright IBM Corporation 2014.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).



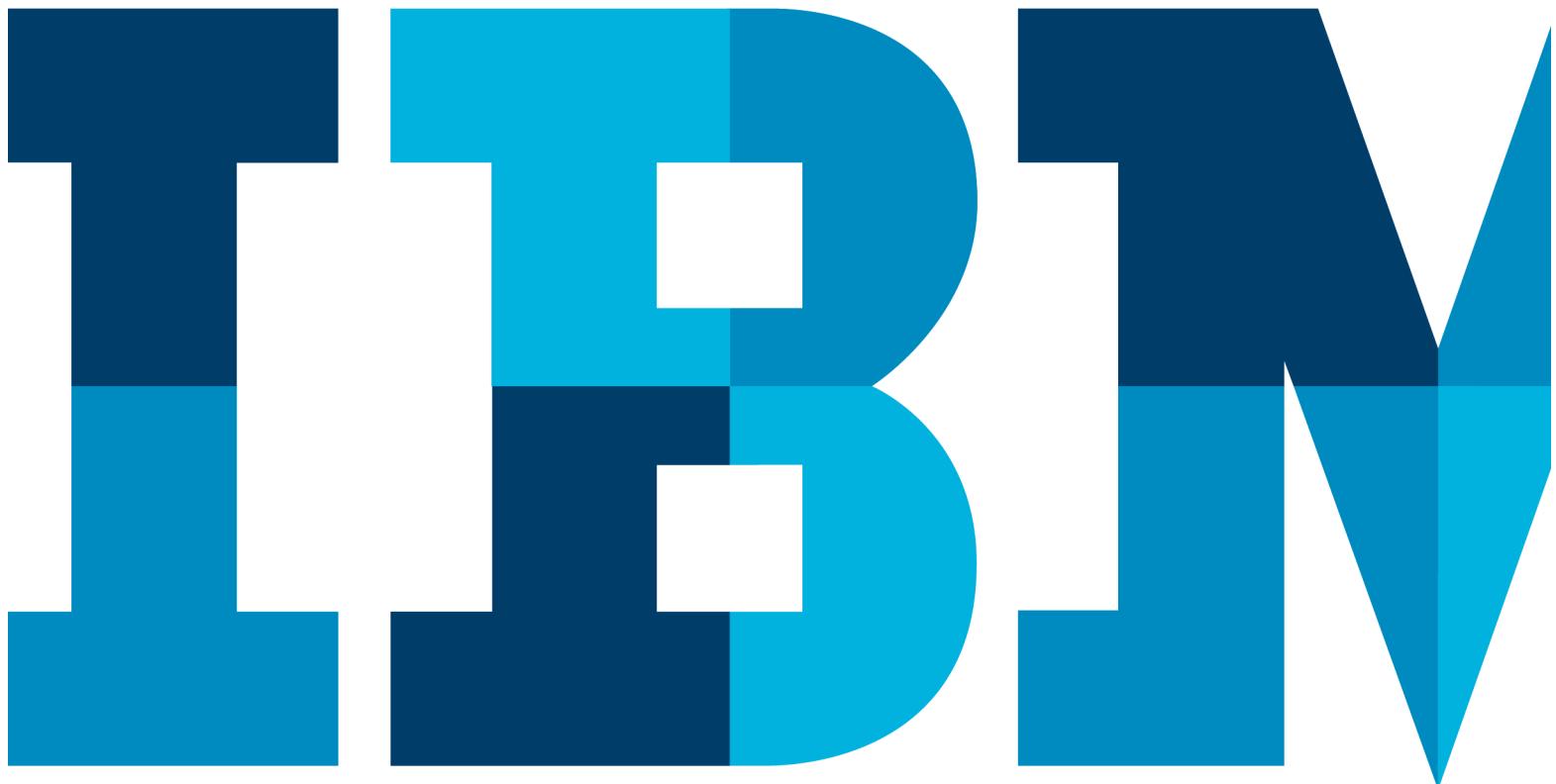
Please Recycle

---

# IBM Blockchain Hands-On

## Hyperledger Fabric Lab – Writing Your First Application

*Lab 4 – VM Exercises*





---

## Overview

The purpose of this lab is to enable you to write your first blockchain application by introducing you to the Hyperledger Fabric SDK.

## Introduction

The lab can be run on any supported level of Mac OSX, Linux and Windows machines. A browser and internet connectivity is required to complete the lab.

Please note that as several hundred MBs in the form of docker images will be downloaded, suitable internet bandwidth and disk space is required.

The following prerequisite software are also required:

- Git commandLine
- cURL (or Windows equivalent)
- Docker
- Docker Compose
- Node.js

It is important to ensure the correct versions of Docker, Docker Compose and Node.js are installed. Incorrect versions will lead to random errors. Please follow directions on this page for installing the correct versions: <http://hyperledger-fabric.readthedocs.io/en/latest/prereqs.html>

# Writing your first Hyperledger Fabric application

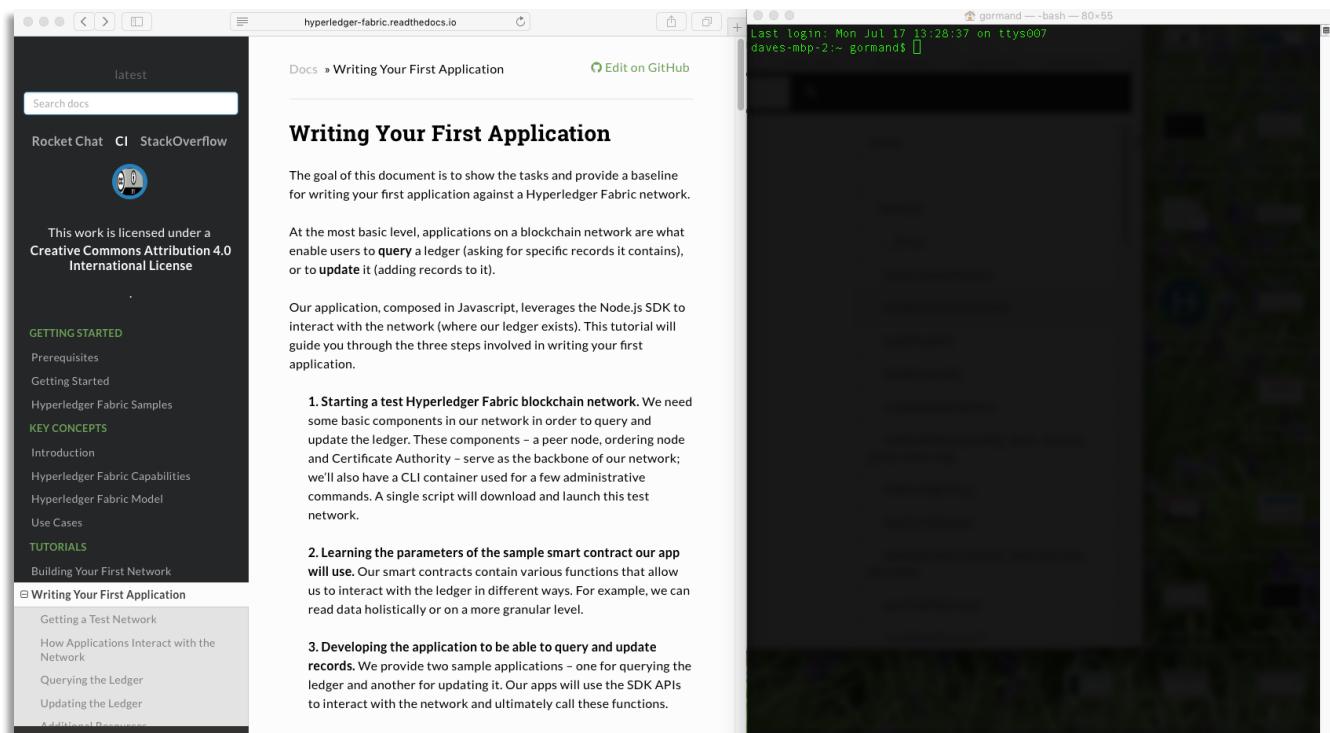
The master copy of the instructions for this lab are online. It is recommended that you use the online version where possible, as this may contain updates to the instructions. The online site also allows you to more easily copy and paste snippets of text, which is necessary for some of the steps.

1. Bring up a web browser and navigate to the following page:

[http://hyperledger-fabric.readthedocs.io/en/latest/write\\_first\\_app.html](http://hyperledger-fabric.readthedocs.io/en/latest/write_first_app.html)

You will also need to open a terminal window.

It is recommended that you open the browser side-by-side with the terminal screen, as you will be working from the browser page and following the instructions in the terminal window.



2. Follow all the instructions contained within the tutorial starting at the Visit the Hyperledger Fabric Samples page. Do not visit the prerequisites page as the prereqs have been installed for you. Ensure your terminal window is initially open to the **/home/blockchain** directory. Once you have successfully run the query.js and invoke.js applications to transfer ownership of a car, you will have completed the lab.

---

## Appendix A. 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.

**COPYRIGHT LICENSE:**

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Appendix B. 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	
<i>System z</i>	<i>Tivoli</i>	<i>WebSphere</i>	<i>Workplace</i>	<i>System p</i>	

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**

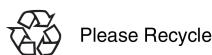


---

© Copyright IBM Corporation 2014.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).



Please Recycle

---