IBM Blockchain Hands-On Blockchain Explored

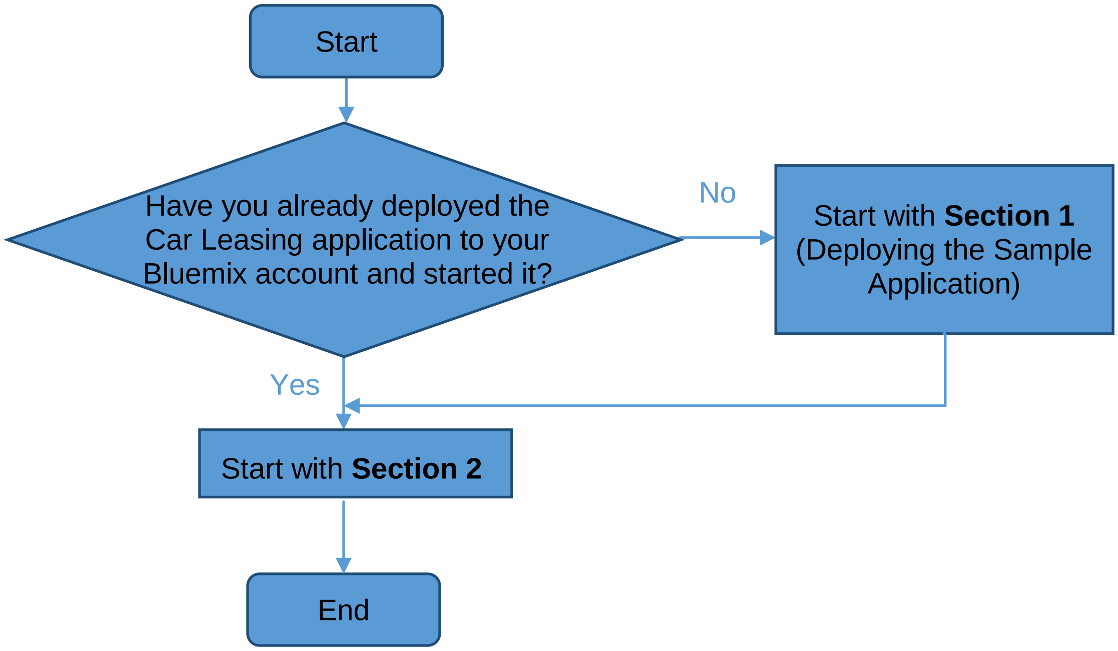
Lab Two –Bluemix –Exercises

Contents

Overview Introduction to the Lab

The purpose of this lab is to introduce you to the Blockchain service on Bluemix. We will build on the car leasing scenario that was introduced in the “Blockchain Explained” lab.

If you are using your own Bluemix account and have already completed the previous lab, you will have already deployed the car leasing application to your account.You can skip section 1 and re-use your existing application:


Start
Have you already deployed the Car Leasing application to your Bluemix account and started it?
No
Start with **Section 1** (Deploying the Sample Application)
Yes
Start with **Section 2**
End

For V0.61 of Hyperledger in Bluemix

Section 1. Deploying thesample application

In this section we will use Bluemix to deploy a copy of the car leasing demo application.

1.1. Creating a Blockchain Service

__1. Open a web browser and go to www.bluemix.net.

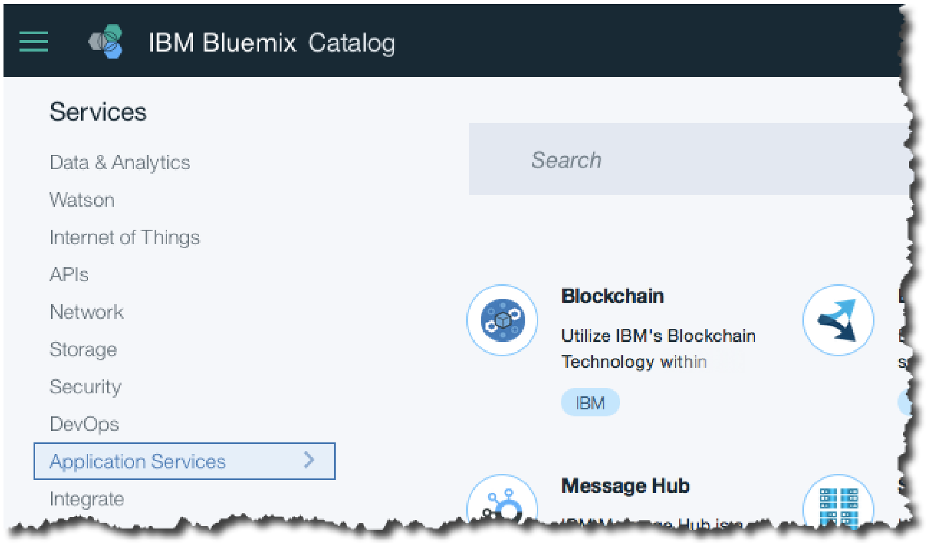
It is recommended to use Firefox or Chrome.

Problems can generally be resolved by clearing the browser's cache and cookies, or running the browser in private mode.

__2. Click '**Sign Up**' or '**Log In**' to create a new Bluemix account or log into your existing account.

__3. Once you have successfully signed up and logged into Bluemix, select Catalog from the top bar.

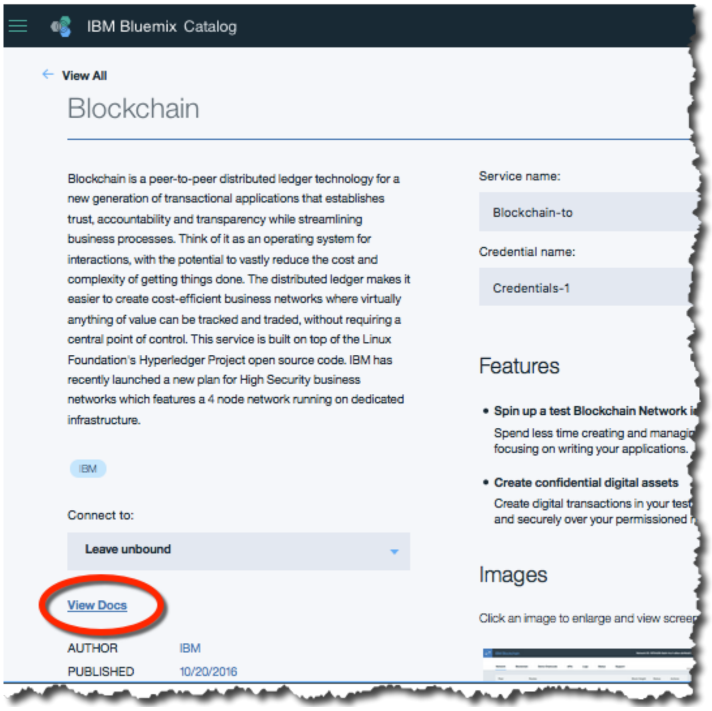
__4. In the 'Services' section of the sidebar, click '**Application Services**'and select **Blockchain**.



Utilize IBM's Blockchain Technology within

__5. Review the service description and information about the service.

__6. Click '**View Docs**' andlearn about the process of creating a blockchain environment.



IBM Bluemix Catalog

← View All

Blockchain

Blockchain is a peer-to-peer distributed ledger technology for a new generation of transactional applications that establishes trust, accountability and transparency while streamlining business processes. Think of it as an operating system for interactions, with the potential to vastly reduce the cost and complexity of getting things done. The distributed ledger makes it easier to create cost-efficient business networks where virtually anything of value can be tracked and traded, without requiring a central point of control. This service is built on top of the Linux Foundation's Hyperledger Project open source code. IBM has recently launched a new plan for High Security business networks which features a 4 node network running on dedicated infrastructure.

IBM

Connect to:

Leave unbound

View Docs

AUTHOR IBM

PUBLISHED 10/20/2016

Service name:

Blockchain-to

Credential name:

Credentials-1

Features

- **Spin up a test Blockchain Network i**
  Spend less time creating and managi
  focusing on writing your applications.
- **Create confidential digital assets**
  Create digital transactions in your test
  and securely over your permissioned

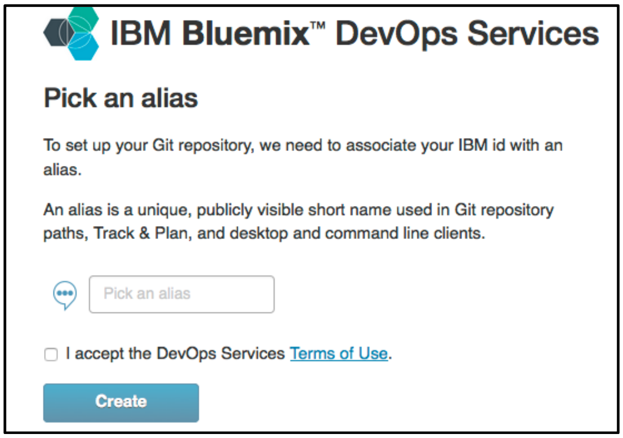
Images

Click an image to enlarge and view scree

__7. Click ‘**Samples and Tutorials**’ on the right of the page.

__8. Click **Deploy to Bluemix** against the Car Lease demo. Log in to Bluemix again if necessary.

__9. If you are asked to pick an alias (usually the first time a given Bluemix ID deploys a sample), come up with a unique username and review and accept the terms of use. Click Create and then Continue on the subsequent page.



IBM Bluemix™ DevOps Services

Pick an alias

To set up your Git repository, we need to associate your IBM id with an alias.

An alias is a unique, publicly visible short name used in Git repository paths, Track & Plan, and desktop and command line clients.

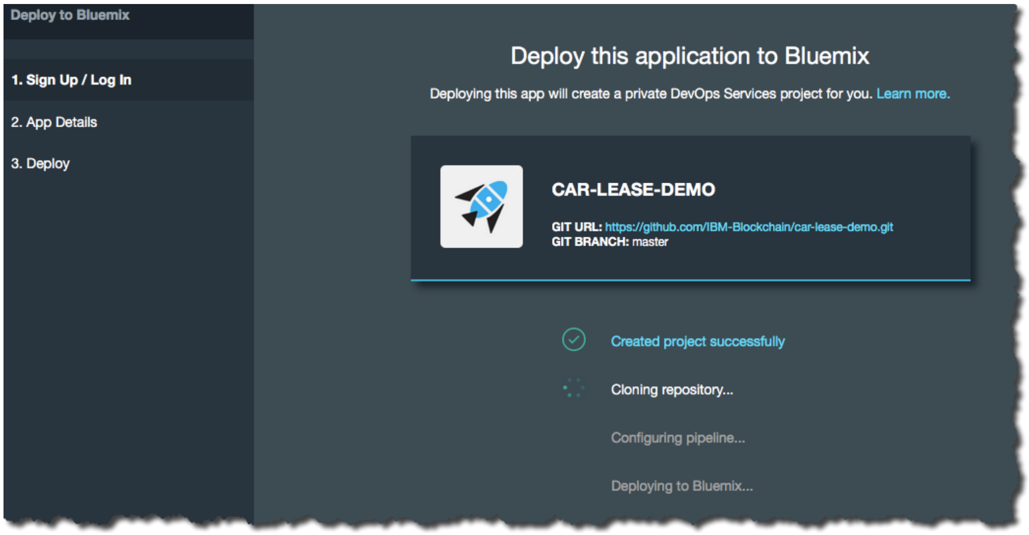
Pick an alias

□ I accept the DevOps Services Terms of Use.

Create

__10. Wait a few seconds for the default field values to be populated. Then leave the App Name, Region, Organization and Space default (unless told otherwise) and click DEPLOY.

This will cause the car leasing demo to be deployed into your Bluemix environment, and will probably take a few minutes to complete.



Deploy this application to Bluemix

Deploying this app will create a private DevOps Services project for you. Learn more.

CAR-LEASE-DEMO

GIT URL: https://github.com/IBM-Blockchain/car-lease-demo.git
GIT BRANCH: master

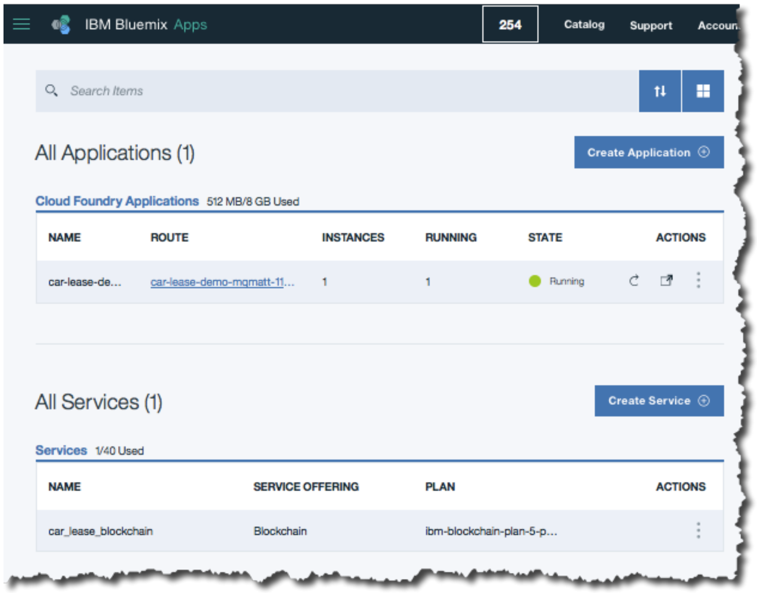
Created project successfully

Cloning repository...

Configuring pipeline...

Deploying to Bluemix...

__11. Once you see the ‘Success!’ message click DASHBOARD to see a description of the new car leasingapplication (and associated Blockchain service) you created.



All Applications (1)

Cloud Foundry Applications 512 MB/8 GB Used

NAME	ROUTE	INSTANCES	RUNNING	STATE	ACTIONS
car-lease-de...	car-lease-demo-mgmatt-11...	1	1	● Running	

All Services (1)

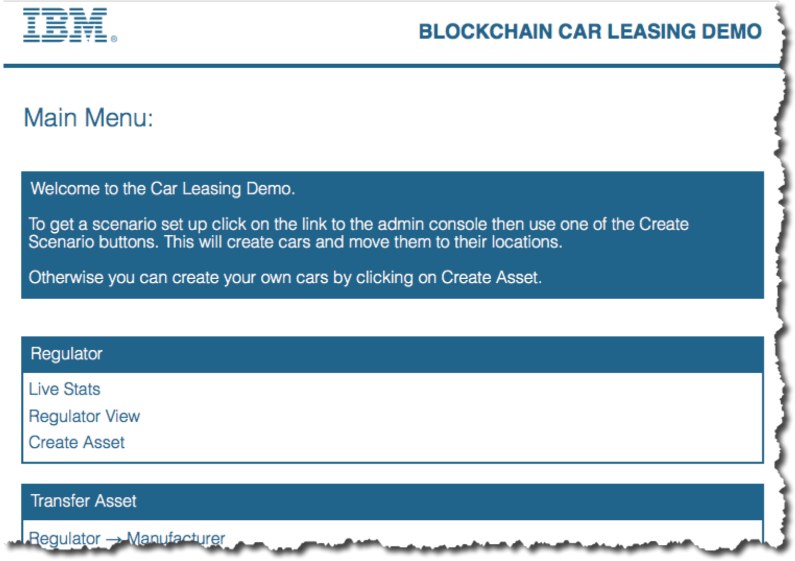
Services 1/40 Used

NAME	SERVICE OFFERING	PLAN	ACTIONS
car_lease_blockchain	Blockchain	ibm-blockchain-plan-5-p...	

1.2. Initializing the Car Leasing Demo

__12. Click the blue hyperlink under the ‘Route’ column of your application, and this will load the Car Leasing demo webpage. (Do not click elsewhere on this line, as this will load the administration interface for the application, which we will look at later).

You will now see the front page of the car leasing demo.



BLOCKCHAIN CAR LEASING DEMO

Main Menu:

Welcome to the Car Leasing Demo.

To get a scenario set up click on the link to the admin console then use one of the Create Scenario buttons. This will create cars and move them to their locations.

Otherwise you can create your own cars by clicking on Create Asset.

Regulator

Live Stats

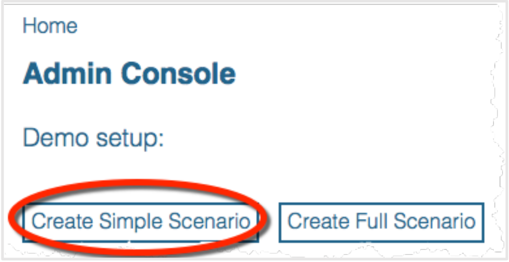
Regulator View

Create Asset

Transfer Asset

Regulator → Manufacturer

__13. From the Car Leasing demo front page, click '**Admin Console**' and '**Create Simple Scenario**'.



Home

Admin Console

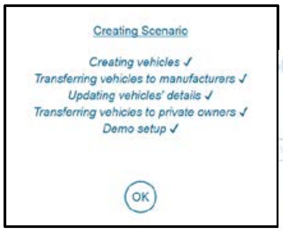
Demo setup:

Create Simple Scenario | Create Full Scenario

This will preload the blockchain with a set of transactions. (The Full Scenario works fine too; the difference between the Simple Scenario and the Full Scenario is that in the Full Scenario more assets are initially loaded onto the blockchain; this takes longer to initialize, however.)

Wait a few moments for the initialization to complete.

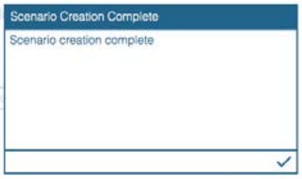
__14. Click '**OK**' to close the Creating Scenario log, and then dismiss the 'Scenario Creation complete' by clicking the check mark.



Creating Scenario

Creating vehicles ✓
Transferring vehicles to manufacturers ✓
Updating vehicles' details ✓
Transferring vehicles to private owners ✓
Demo setup ✓

OK



Scenario Creation Complete

Scenario creation complete

✓

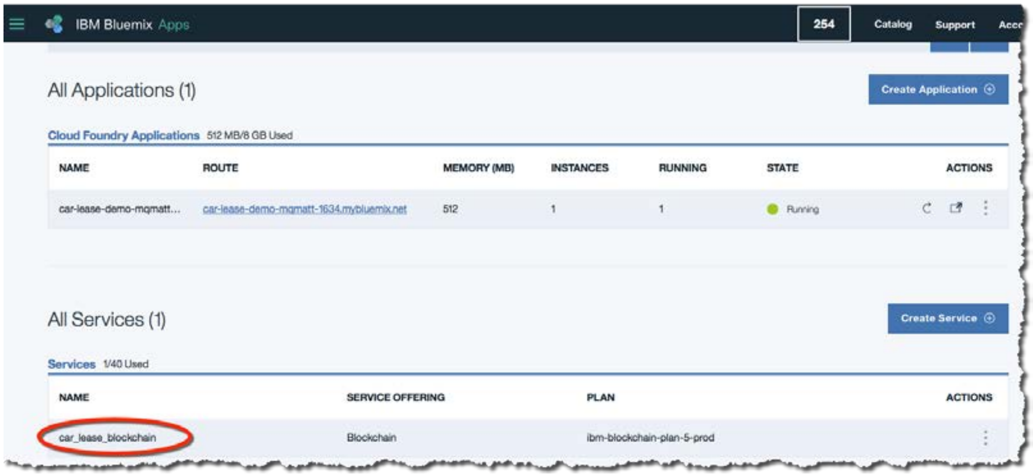
Section 2. Managing the sample application

In this section we will use the monitoring tools available inside the Bluemix environment to view and manage the blockchain.

2.1. Viewing the components of the Blockchain service

__15. Return to the Bluemix dashboard, either by selecting the ☰ icon in the top left of Bluemix and selecting '**Dashboard**' or by going directly to https://console.ng.bluemix.net/dashboard/applications.

__16. Click on the **car_lease_blockchain** service in the Services section of the dashboard.



IBM Bluemix Apps 254 Catalog Support Acc

All Applications (1) Create Application

Cloud Foundry Applications 512 MB/8 GB Used

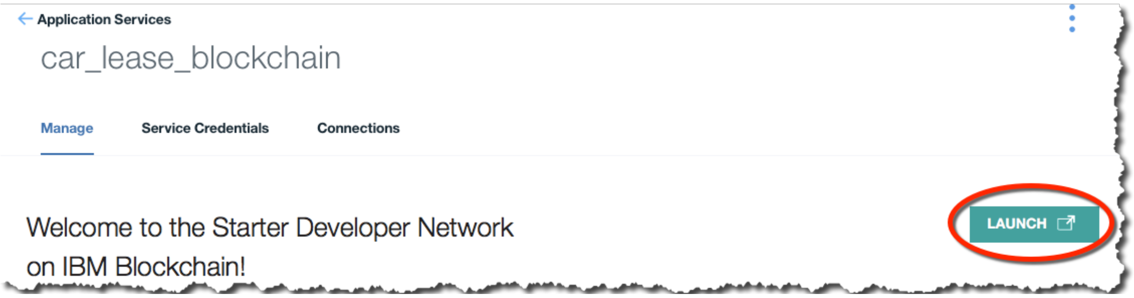
NAME	ROUTE	MEMORY (MB)	INSTANCES	RUNNING	STATE	ACTIONS
car-lease-demo-mqmatt...	car-lease-demo-mqmatt-1634.mybluemix.net	512	1	1	● Running	

All Services (1) Create Service

Services 1/40 Used

NAME	SERVICE OFFERING	PLAN	ACTIONS
car_lease_blockchain	Blockchain	ibm-blockchain-plan-5-prod	

__17. Review the details and select LAUNCH to launch the blockchain adminstration console.



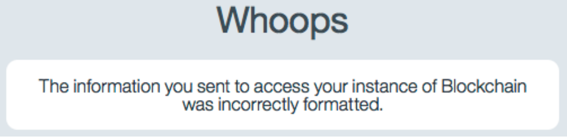
Application Services

car_lease_blockchain

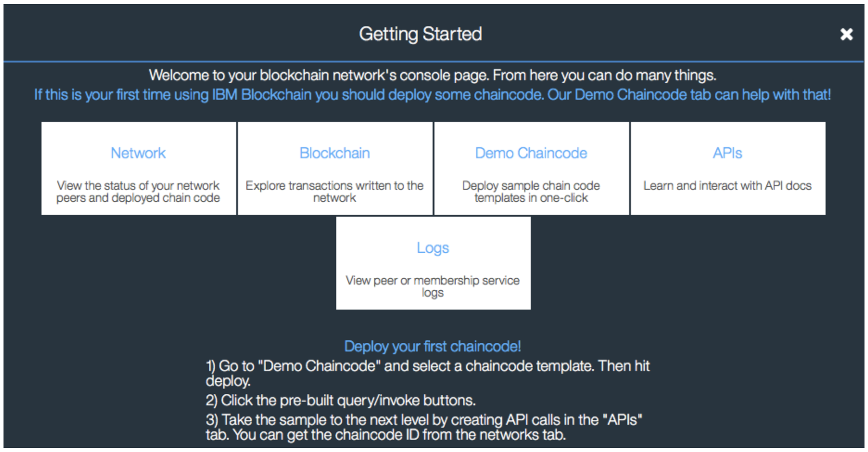
Manage Service Credentials Connections

Welcome to the Starter Developer Network on IBM Blockchain!

LAUNCH

⚠	If you see the following error, then try changing to a Firefox or Chrome browser:  Whoops The information you sent to access your instance of Blockchain was incorrectly formatted.

__18. Close the pop-up showing information about the sections. We'll look at these sections in more detail throughout this lab.



Getting Started ×

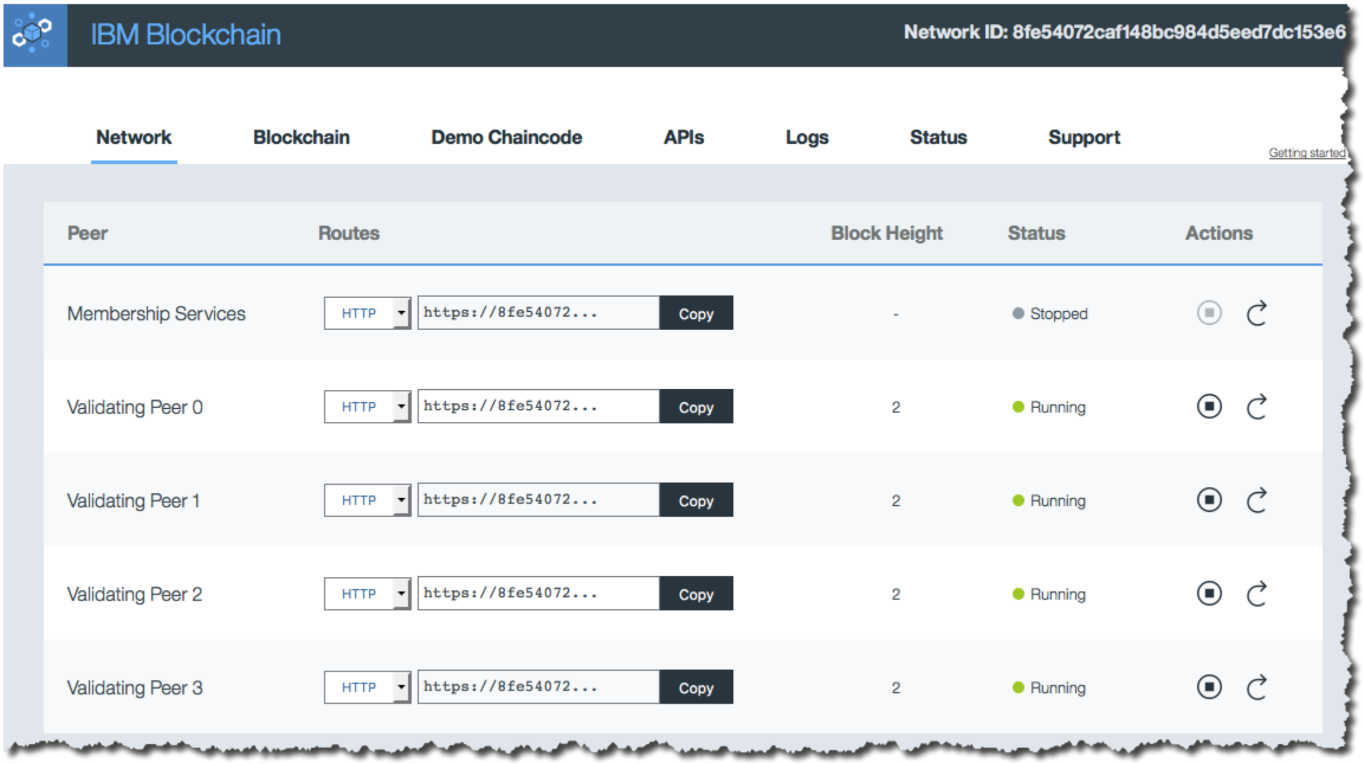
Welcome to your blockchain network's console page. From here you can do many things.
If this is your first time using IBM Blockchain you should deploy some chaincode. Our Demo Chaincode tab can help with that!

Network	Blockchain	Demo Chaincode	APIs
View the status of your network peers and deployed chain code	Explore transactions written to the network	Deploy sample chain code templates in one-click	Learn and interact with API docs

Logs
View peer or membership service logs

Deploy your first chaincode!
1) Go to "Demo Chaincode" and select a chaincode template. Then hit deploy.
2) Click the pre-built query/invoke buttons.
3) Take the sample to the next level by creating API calls in the "APIs" tab. You can get the chaincode ID from the networks tab.

You should now see the administration page with seven tabs along the top. The '**Network**' tab will be selected.



IBM Blockchain   Network ID: 8fe54072caf148bc984d5eed7dc153e6

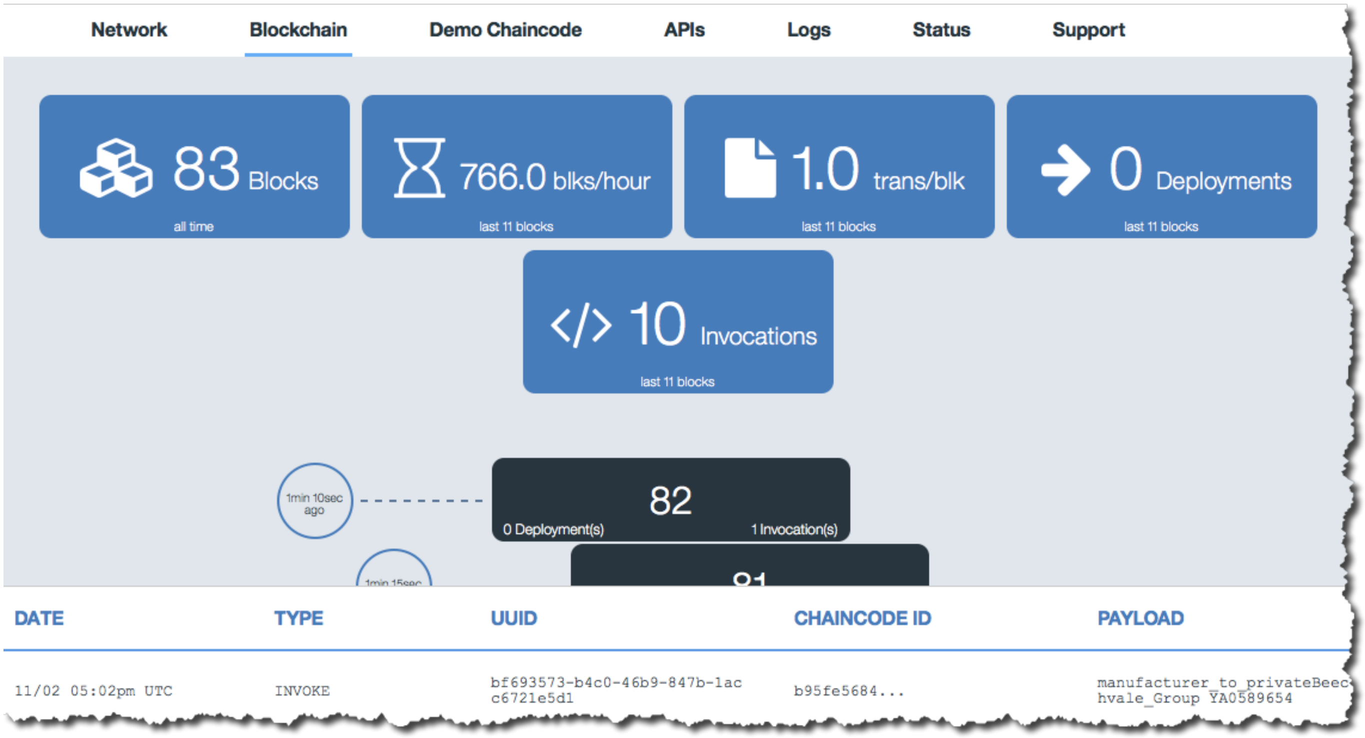
Peer	Routes			Block Height	Status	Actions
Membership Services	HTTP	https://8fe54072...	Copy	-	● Stopped	⊙ ↻
Validating Peer 0	HTTP	https://8fe54072...	Copy	2	● Running	⊙ ↻
Validating Peer 1	HTTP	https://8fe54072...	Copy	2	● Running	⊙ ↻
Validating Peer 2	HTTP	https://8fe54072...	Copy	2	● Running	⊙ ↻
Validating Peer 3	HTTP	https://8fe54072...	Copy	2	● Running	⊙ ↻

This view confirms that four validating peers and a certificate authority (Membership Services) are running under your Blockchain service. The block height should be the same for each of the validating peers.

2.2. Viewing the Blockchain Explorer

The Blockchain tab shows a visual representation of the state of the Blockchain.

__19. Click the ‘**Blockchain**’tab at the top of the page.



DATE	TYPE	UUID	CHAINCODE ID	PAYLOAD
11/02 05:02pm UTC	INVOKE	bf693573-b4c0-46b9-847b-1ac c6721e5d1	b95fe5684...	manufacturer_to_privateBeec hvale_Group YA0589654

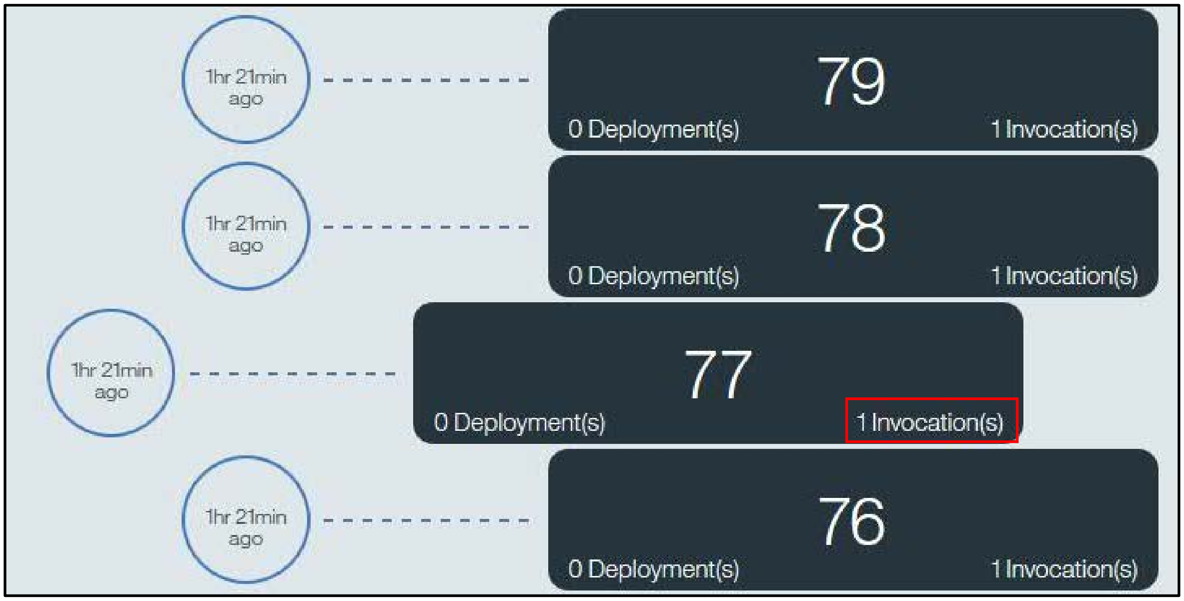
The icons show:

8 Blocks all time	Total number of blocks in the chain
0.1 blks/hour last 8 blocks	Average number of blocks per hour
1.0 trans/blk last 8 blocks	Average number of transactions per block
→ 3 Deployments last 8 blocks	Number of deployment calls made to deploy chaincode
</> 4 Invocations last 8 blocks	Number of invoke requests made within this blockchain

Each block contains a set of transactions. In Hyperledger, a transaction is the record of the request to interact with chaincode (a smart contract). Two important transaction types are:

- *DEPLOY*: The request to deploy a piece of chaincode across all validating peers, so that it can be executed at a later date.
- *INVOKE*: The request to invoke a piece of chaincode (for example, invoke the chaincodeto transfer the ownership of a car)

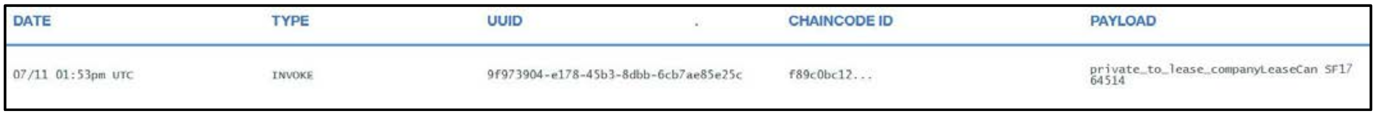
Other request types exist (e.g. query, update, terminate). Not all request types are recorded on the blockchain.



The blocks also include when that block was committed to the blockchain.

__20. Click on a block that contains at least one invocation request.

__21. Look through the list of transactions that are contained within the block.



DATE	TYPE	UUID	CHAINCODE ID	PAYLOAD
07/11 01:53pm UTC	INVOKE	9f973904-e178-45b3-8dbb-6cb7ae85e25c	f89c0bc12...	private_to_lease_companyLeaseCan SF17 64514

Each line of information is a transaction stored within the block. A block may contain multiple transactions but in this demo there will often only be one transaction per block due to the low frequencyof transactions being made. The information displayed is:

Date	The date the transaction was submitted.
Type	The type of transaction taking place (e.g. INVOKE or DEPLOY).
UUID	The unique identifier for each transaction.
Chaincode ID	Refers to the chaincode that is being invoked or deployed.
Payload	The input parameters to the chaincode.

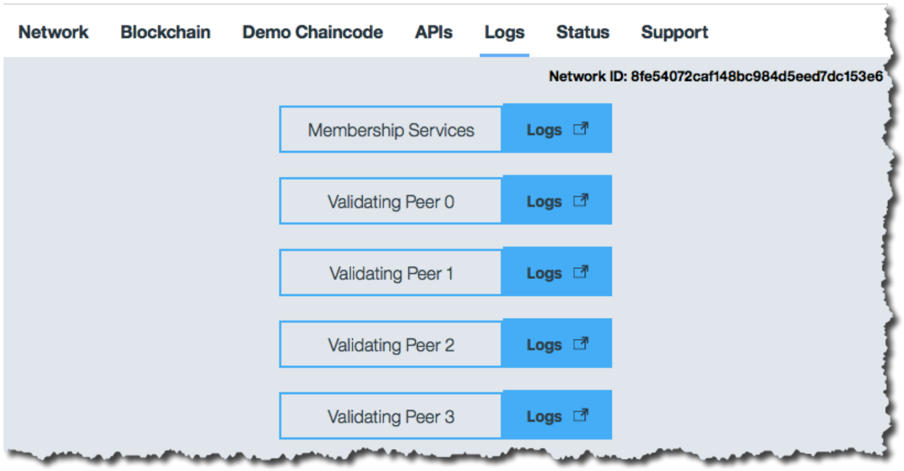
__22. Repeat this for other blocks to understand how the transactions are stored.

> When the Blockchain service is initialised for the car leasing application, the first two blocks in the chain usually contain 'Deploy' transactions, where the chaincode is deployed to the validating peers.
>
> View these blocks If you're willing to scroll down the Blockchain explorer that far!

2.3. Understanding the Blockchain Peers

We are now going to review the logs associated with the peers. This is useful for understanding how the blockchain works, and for diagnosing problems.

__23. Click on the '**Logs**' tab at the top of the service page.

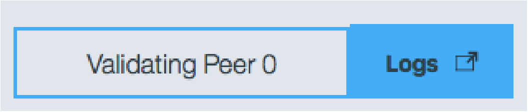


Network ID: 8fe54072caf148bc984d5eed7dc153e6

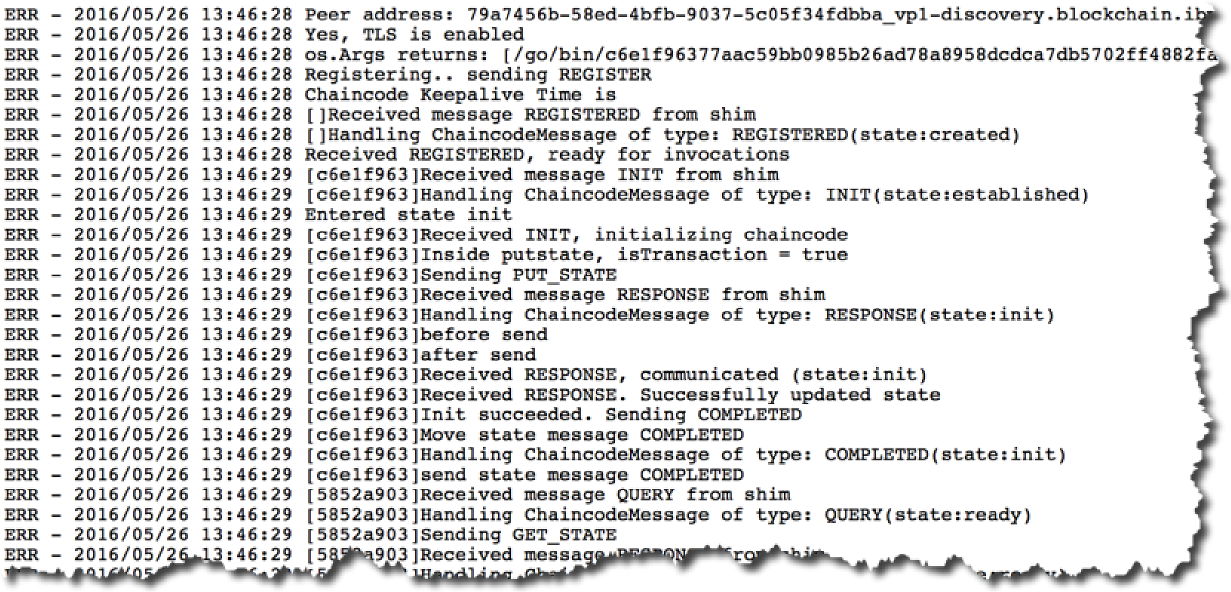
Here we can see that this Blockchain service contains four validating peers and a Certificate Authority.

By looking at the logs for each peer you can verify that every node has executed every transaction.

__24. Click the Logs button against one of the validating peers.



This will show the logs for the selected peer in a new window.



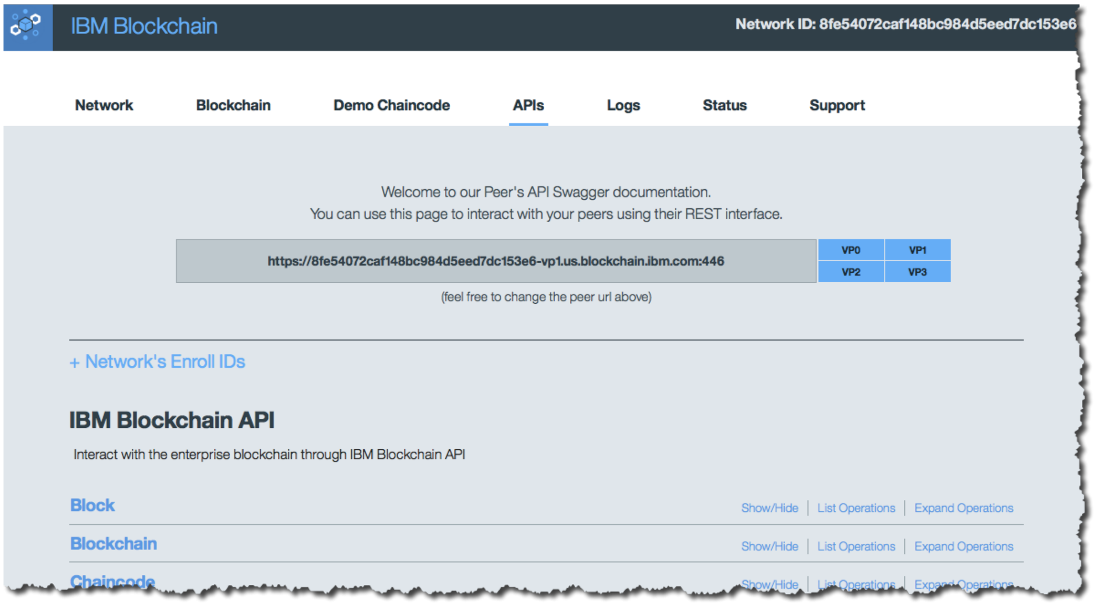
```
ERR - 2016/05/26 13:46:28 Peer address: 79a7456b-58ed-4bfb-9037-5c05f34fdbba_vp1-discovery.blockchain.ib
ERR - 2016/05/26 13:46:28 Yes, TLS is enabled
ERR - 2016/05/26 13:46:28 os.Args returns: [/go/bin/c6e1f96377aac59bb0985b26ad78a8958dcdca7db5702ff4882fa
ERR - 2016/05/26 13:46:28 Registering.. sending REGISTER
ERR - 2016/05/26 13:46:28 Chaincode Keepalive Time is
ERR - 2016/05/26 13:46:28 []Received message REGISTERED from shim
ERR - 2016/05/26 13:46:28 []Handling ChaincodeMessage of type: REGISTERED(state:created)
ERR - 2016/05/26 13:46:28 Received REGISTERED, ready for invocations
ERR - 2016/05/26 13:46:29 [c6e1f963]Received message INIT from shim
ERR - 2016/05/26 13:46:29 [c6e1f963]Handling ChaincodeMessage of type: INIT(state:established)
ERR - 2016/05/26 13:46:29 Entered state init
ERR - 2016/05/26 13:46:29 [c6e1f963]Received INIT, initializing chaincode
ERR - 2016/05/26 13:46:29 [c6e1f963]Inside putstate, isTransaction = true
ERR - 2016/05/26 13:46:29 [c6e1f963]Sending PUT_STATE
ERR - 2016/05/26 13:46:29 [c6e1f963]Received message RESPONSE from shim
ERR - 2016/05/26 13:46:29 [c6e1f963]Handling ChaincodeMessage of type: RESPONSE(state:init)
ERR - 2016/05/26 13:46:29 [c6e1f963]before send
ERR - 2016/05/26 13:46:29 [c6e1f963]after send
ERR - 2016/05/26 13:46:29 [c6e1f963]Received RESPONSE, communicated (state:init)
ERR - 2016/05/26 13:46:29 [c6e1f963]Received RESPONSE. Successfully updated state
ERR - 2016/05/26 13:46:29 [c6e1f963]Init succeeded. Sending COMPLETED
ERR - 2016/05/26 13:46:29 [c6e1f963]Move state message COMPLETED
ERR - 2016/05/26 13:46:29 [c6e1f963]Handling ChaincodeMessage of type: COMPLETED(state:init)
ERR - 2016/05/26 13:46:29 [c6e1f963]send state message COMPLETED
ERR - 2016/05/26 13:46:29 [5852a903]Received message QUERY from shim
ERR - 2016/05/26 13:46:29 [5852a903]Handling ChaincodeMessage of type: QUERY(state:ready)
ERR - 2016/05/26 13:46:29 [5852a903]Sending GET_STATE
ERR - 2016/05/26 13:46:29 [5852a903]Received message RESPONSE from shim
```

2.4. Interacting with the peers

It is possible to invoke the management APIs that interact directly with the peers. In this section we will be trying out these APIs directly from the Bluemix environment.

Note that the APIs concern *operationally managing* the Blockchain service – this is not the same as adding and invoking transactions through chaincode!

__25. Click on the '**APIs**' tab on the service page.



IBM Blockchain

Network ID: 8fe54072caf148bc984d5eed7dc153e6

Welcome to our Peer's API Swagger documentation.
You can use this page to interact with your peers using their REST interface.

https://8fe54072caf148bc984d5eed7dc153e6-vp1.us.blockchain.ibm.com:446

(feel free to change the peer url above)

+ Network's Enroll IDs

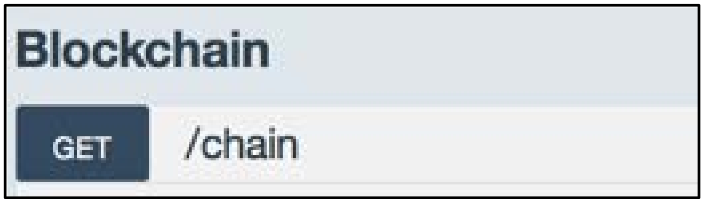
IBM Blockchain API

Interact with the enterprise blockchain through IBM Blockchain API

Block
Show/Hide | List Operations | Expand Operations
Blockchain
Show/Hide | List Operations | Expand Operations
Chaincode
Show/Hide | List Operations | Expand Operations

This page allows you to invoke APIs that will directly interrogate and manage the blockchain. First we will use the API interface to query the height of the Blockchain (the number of blocks).

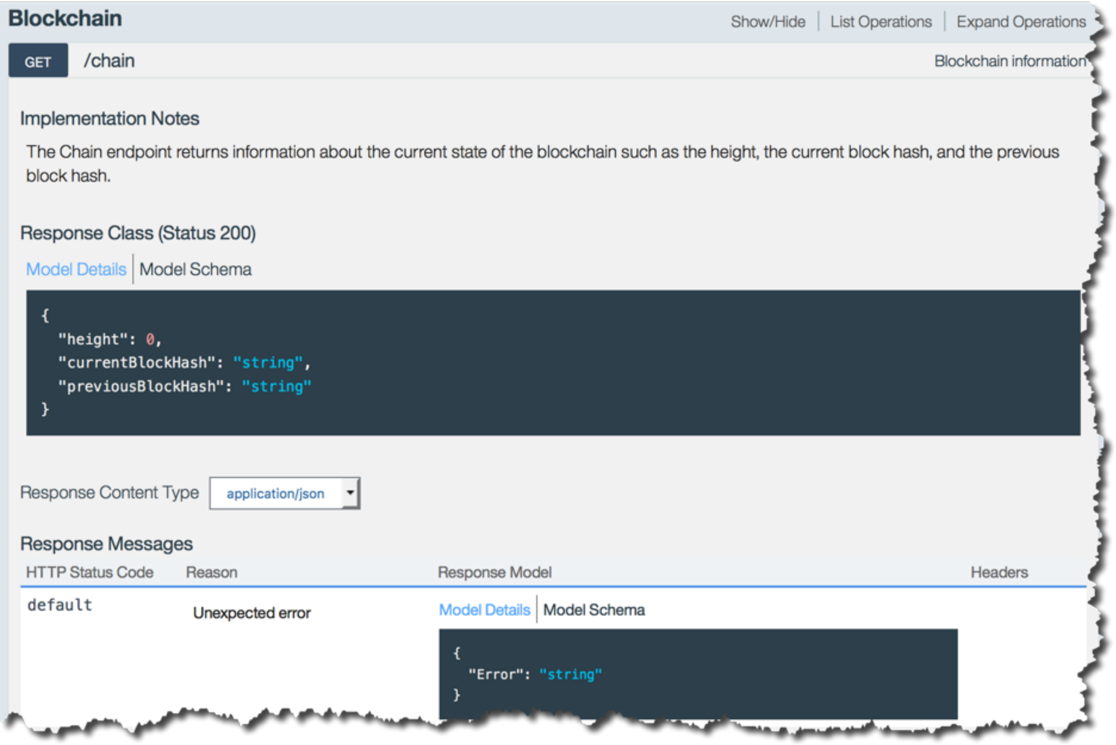
__26. Click the ‘**Blockchain**’ section.



This reveals the **GET /chain** operation which is a valid method to call on the peer.

__27. Click ‘**Expand Operation**’ to view information about this API.

This reveals the input and output data formats.



Implementation Notes

The Chain endpoint returns information about the current state of the blockchain such as the height, the current block hash, and the previous block hash.

Response Class (Status 200)

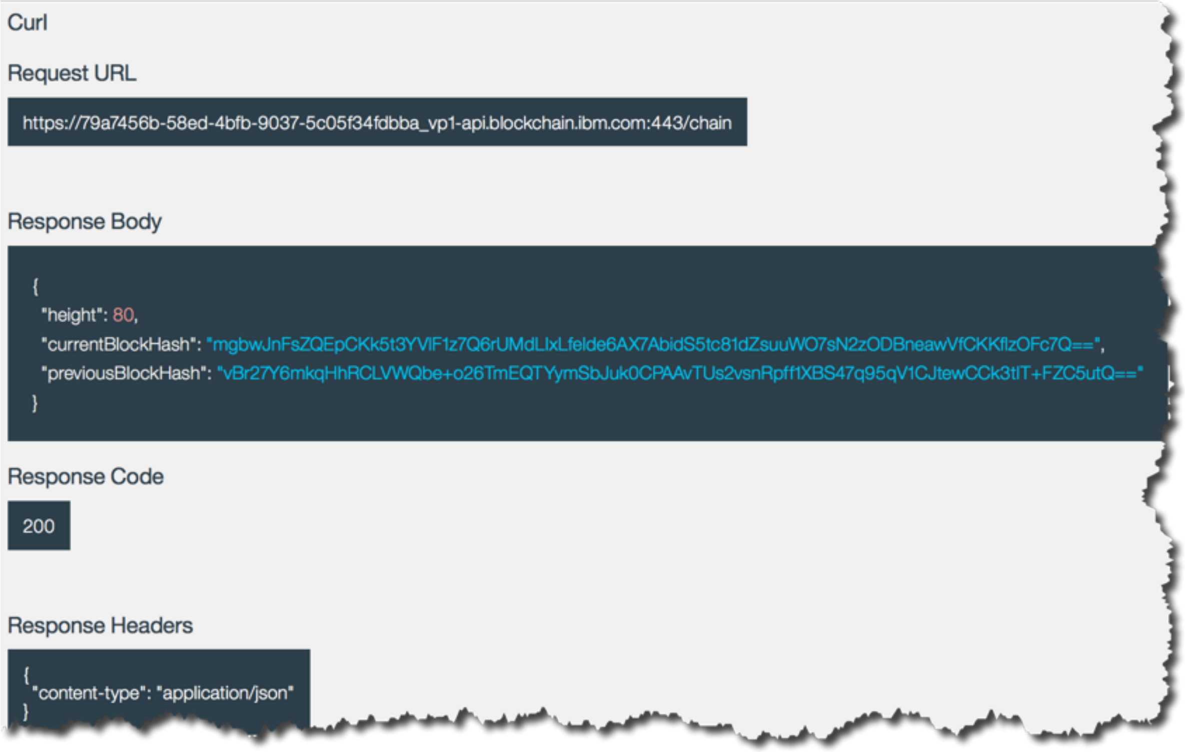
```
{
  "height": 0,
  "currentBlockHash": "string",
  "previousBlockHash": "string"
}
```

Response Content Type: application/json

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	Unexpected error	Model Details | Model Schema { "Error": "string" }	

__28. Click ‘**Try It Out**’ to invoke the API.



Curl

Request URL

```
https://79a7456b-58ed-4bfb-9037-5c05f34fdbba_vp1-api.blockchain.ibm.com:443/chain
```

Response Body

```
{
  "height": 80,
  "currentBlockHash": "mgbwJnFsZQEpCKk5t3YVlF1z7Q6rUMdLlxLfelde6AX7AbidS5tc81dZsuuWO7sN2zODBneawVfCKKflzOFc7Q==",
  "previousBlockHash": "vBr27Y6mkqHhRCLVWQbe+o26TmEQTYymSbJuk0CPAAvTUs2vsnRpff1XBS47q95qV1CJtewCCk3tIT+FZC5utQ=="
}
```

Response Code

```
200
```

Response Headers

```
{
 "content-type": "application/json"
}
```

Review the displayed fields:

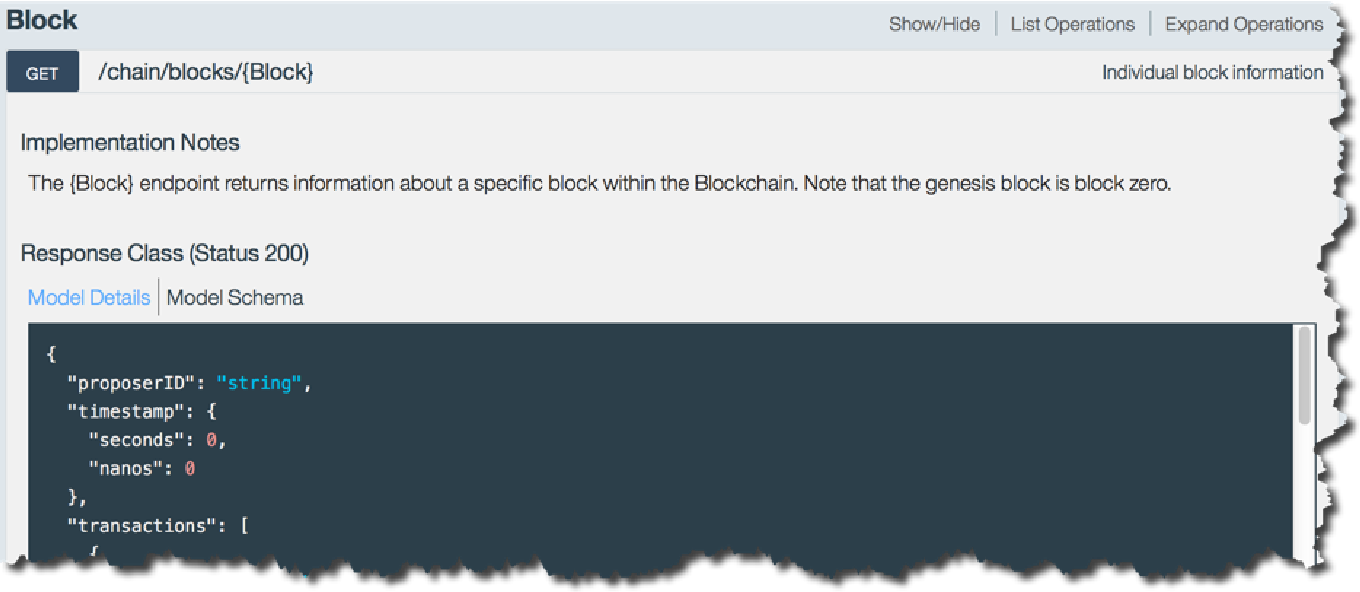
- The *Request URL* shows the URL that was invoked, including the endpoint information of the peer (hostname:port) and the method call (/chain).

- The *Response Body* shows the information that was returned including, importantly, the height of the blockchain.

- The *Response Code* 200 shows that the request was successful.

- The *Response Headers* confirms that the response body has been returned in a JSON data structure.

__29. Expand the '**Block**' section and review the information on how to interrogate an individual block in the Blockchain.



Block

Show/Hide | List Operations | Expand Operations

GET /chain/blocks/{Block} Individual block information

Implementation Notes

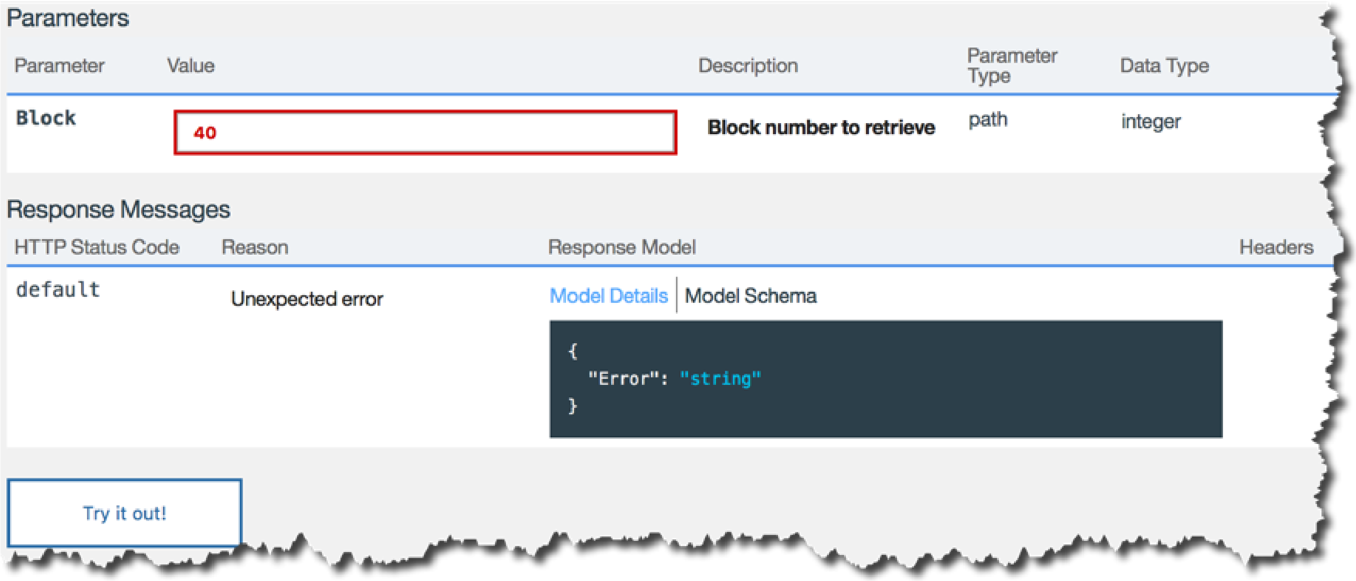
The {Block} endpoint returns information about a specific block within the Blockchain. Note that the genesis block is block zero.

Response Class (Status 200)

Model Details | Model Schema

```
{
  "proposerID": "string",
  "timestamp": {
    "seconds": 0,
    "nanos": 0
  },
  "transactions": [
    {
```

__30. Fill in the ‘**Block**’ parameter to be a number less than the height of the chain and click ‘**Try it out!**’.



Parameters

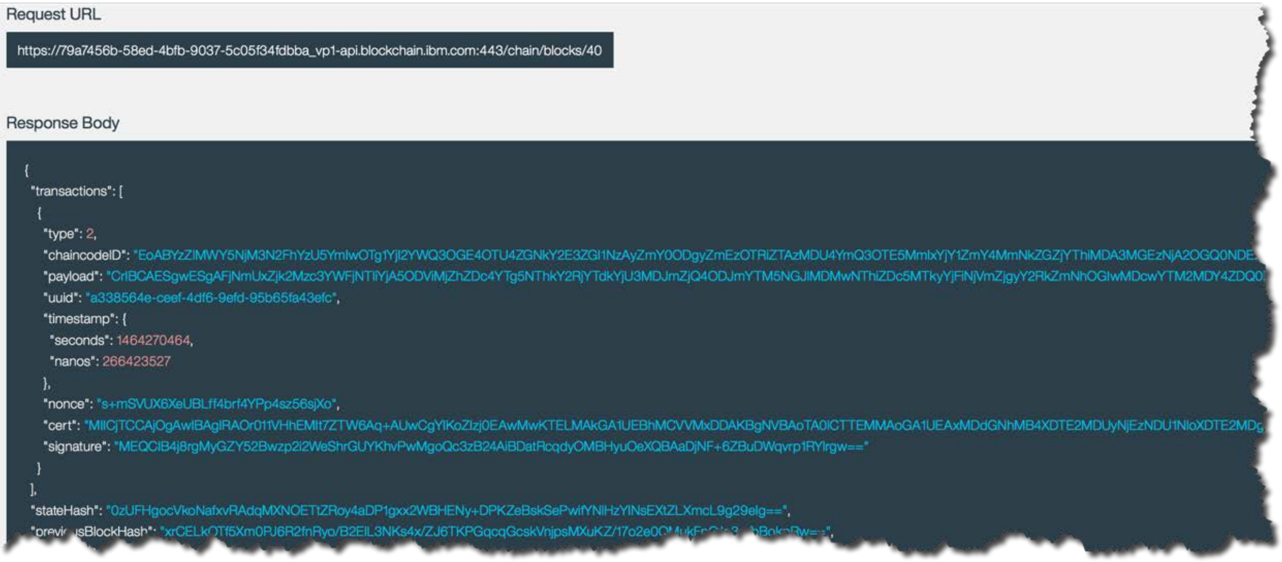
Parameter	Value	Description	Parameter Type	Data Type
Block	40	Block number to retrieve	path	integer

Response Messages

HTTP Status Code	Reason	Response Model	Headers
default	Unexpected error	Model Details | Model Schema { "Error": "string" }	

```
{
  "Error": "string"
}
```

__31. Review the information returned in the Response Body.



Request URL

```
https://79a7456b-58ed-4bfb-9037-5c05f34fdbba_vp1-api.blockchain.ibm.com:443/chain/blocks/40
```

Response Body

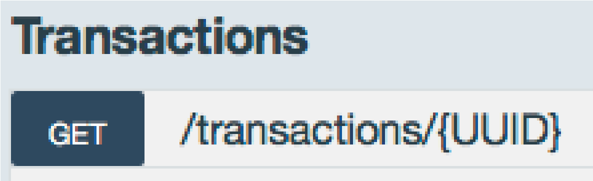
```
{
 "transactions": [
  {
   "type": 2,
   "chaincodeID": "EoABYzZlMWY5NjM3N2FhYzU5YmIwOTg1YjI2YWQ3OGE4OTU4ZGNkY2E3ZGI1NzAyZmY0ODgyZmEzOTRiZTAzMDU4YmQ3OTE5MmIxYjY1ZmY4MmNkZGZjYThiMDA3MGEzNjA2OGQ0NDE
   "payload": "CrIBCAESgwESgAFjNmUxZjk2Mzc3YWFjNTliYjA5ODViMjZhZDc4YTg5NThkY2RjYTdkYjU3MDJmZjQ4ODJmYTM5NGJiMDMwNThiZDc5MTkyYjFiNjVmZjgyY2RkZmNhOGIwMDcwYTM2MDY4ZDQ0
   "uuid": "a338564e-ceef-4df6-9efd-95b65fa43efc",
   "timestamp": {
    "seconds": 1464270464,
    "nanos": 266423527
   },
   "nonce": "s+mSVUX6XeUBLff4brf4YPp4sz56sjXo",
   "cert": "MIICjTCCAjOgAwIBAgIRAOr011VHhEMIt7ZTW6Aq+AUwCgYIKoZIzj0EAwMwKTELMAkGA1UEBhMCVVMxDDAKBgNVBAoTA0lCTTEMMAoGA1UEAxMDdGNhMB4XDTE2MDUyNjEzNDU1NloXDTE2MDg
   "signature": "MEQCIB4j8rgMyGZY52Bwzp2i2WeShrGUYKhvPwMgoQc3zB24AiBDatRcqdyOMBHyuOeXQBAaDjNF+6ZBuDWqvrp1RYIrgw=="
  }
 ],
 "stateHash": "0zUFHgocVkoNafxvRAdqMXNOETtZRoy4aDP1gxx2WBHENy+DPKZeBskSePwifYNIHzYINsEXtZLXmcL9g29elg==",
 "previousBlockHash": "xrCELkOTf5Xm0RJ6R2fnRyo/B2EIL3NKs4x/ZJ6TKPGqcqGcskVnjpsMXuKZ/17o2e0QMukFnGik3 bBokpRw==",
```

Block contentdescription:

transactions	An array of transactions stored in the block.
type	This shows the possible values: 0. Undefined 1. Deployment 2. Invoke 3. Query
chaincodeID	ID of the chaincode that was invoked or deployed.
payload	Input parameters to the chaincode.
uuid	Unique identifier of this transaction.
timestamp	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	Hash of the world state changes.
previousBlockHash	Hash of the previous block in the chain.
nonHashData	Data stored with the block, but not included in the block's hash. This allows data to be different per peer or discarded without affecting the blockchain.
localLedgerCommitTimestamp	Time the block was added to the ledger on the local peer.

__32. Copy the UUID field of a transaction from a block; this will be of the form “a338564e-ceef-4df6-9efd-95b65fa43efc”.

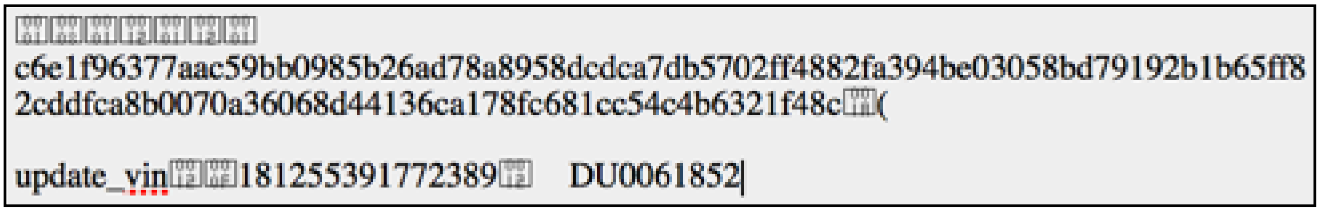
__33. Click the ‘**Transactions**’ section.



This reveals the **GET /transactions/{UUID}** operation which is a valid method to call on the peer.

__34. Paste the transaction UUID and click ‘**Try it out!**’.

The ‘payload’ field is base64 encoded (use a web tool such as http://www.base64decode.orgfor decoding this information); when decoded you’ll see that the payload includes the chaincode ID of the smart contract being called together with its input parameters. For example:



```
□□□□□□□
c6e1f96377aac59bb0985b26ad78a8958dcdca7db5702ff4882fa394be03058bd79192b1b65ff8
2cddfca8b0070a36068d44136ca178fc681cc54c4b6321f48c□(

update_vin□□181255391772389□    DU0061852
```

Note that this application does not encrypt the transactions, so the payloads are visible (albeit base64 encoded) to all.

__35. Now spend some time interacting with the other APIs available to you.

2.5. Viewing the Service Status, Support Contacts and Samples

__36. Click on the ‘**Status**’ tab at the top of the service page.

This page shows you the recent availability of the Blockchain service on Bluemix, and also the version of Hyperledger Fabric that is being used by your network.

__37. Click on the ‘**Support**’ tab at the top of the service page.

This page shows you how to get more help with IBM Bluemix and the Blockchain service.

__38. Click on the ‘**Demo Chaincode**’ tab at the top of the service page.

This page gives the opportunity to deploy more samples to the Blockchain service, and also some how to get started with writing your own blockchain applications and chaincode.

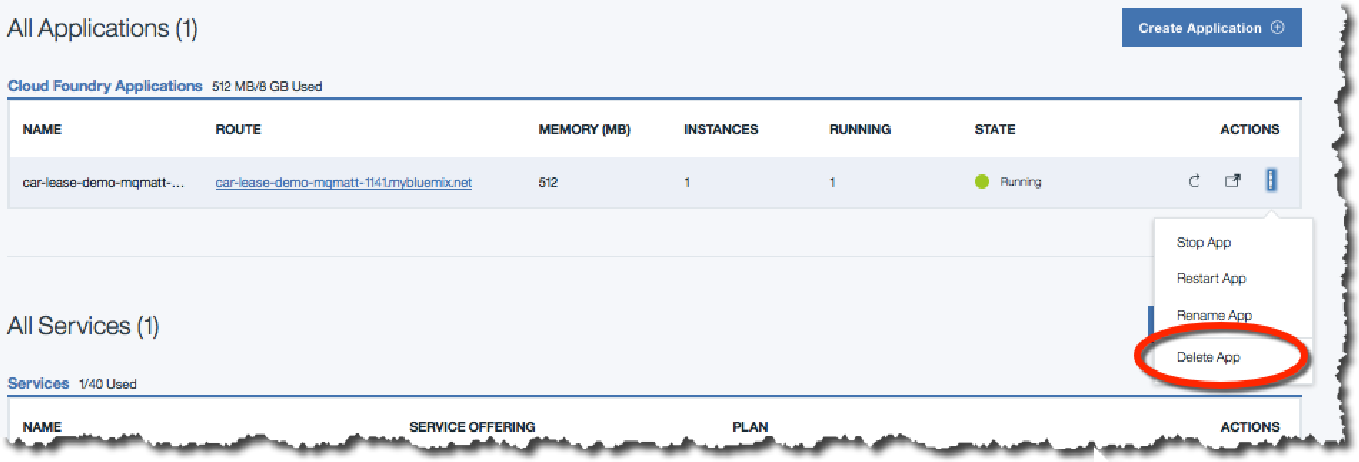
We will look at chaincode development in more detail in the follow-on lab “Blockchain Unchained”.

Section 3. Removing the sample application

The final section of this lab aims to stop and remove the Blockchain service you created.

__39. Return to the Bluemix Dashboard (https://console.ng.bluemix.net/dashboard/applications).

__40. Click the three vertical dots at the right of the Car Leasing application Settings icon in the car lease demo application and select ‘**Delete App**’ from the menu.



All Applications (1)

Cloud Foundry Applications 512 MB/8 GB Used

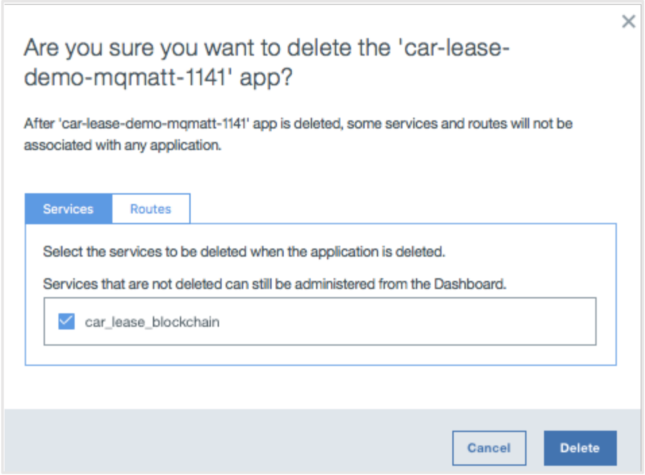
NAME	ROUTE	MEMORY (MB)	INSTANCES	RUNNING	STATE	ACTIONS
car-lease-demo-mqmatt-...	car-lease-demo-mqmatt-1141.mybluemix.net	512	1	1	● Running	

All Services (1)

Services 1/40 Used

NAME	SERVICE OFFERING	PLAN	ACTIONS

__41. Ensure that the ‘car_lease_blockchain’ service is selected for deletion and click ‘**Delete**’.



Are you sure you want to delete the 'car-lease-demo-mqmatt-1141' app?

After 'car-lease-demo-mqmatt-1141' app is deleted, some services and routes will not be associated with any application.

Select the services to be deleted when the application is deleted.

Services that are not deleted can still be administered from the Dashboard.

☑ car_lease_blockchain

__42. Wait for the items to be stopped and deleted. Once this is done, both the application and the associated service will no longer be visible in the Bluemix dashboard.

Congratulations on completing Lab two – “Blockchain Explored”!

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.

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	
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