Hyperledger

Cyrill John G. Mendoza

**Device**

Acer

-4GB Ram

-intel core i3

-Windows 10

**Software Requirements**

Virtual Box (w/ Ubuntu Virtual Machine)

Postman (installed inside the Virtual Machine)

Visual Studio Code (installed inside the Virtual Machine)

Download/clone this repository to skip step 2 and 5

**Steps and Procedures**

*Step 1:*

Open the Terminal and paste the ff. commands

**curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh**

**chmod u+x prereqs-ubuntu.sh**

**./prereqs-ubuntu.sh**

Install Go lang

Go to <https://golang.org/dl/> and download Go for Linux

Go to the directory where the file is located; then inside the folder Right click and Open in terminal.

Type "**tar -C /usr/local -xzf go1.11.5.linux-amd64.tar.gz**" then enter

Add Go environment variable by copying the ff. commands in the terminal:

**export PATH=$PATH:/usr/local/go/bin**

**export GOPATH=$HOME/go**

**export PATH=$PATH:$GOPATH/bin**

*Step 2:*

Download Fabric samples

In the terminal copy the ff. commands:

**git clone https://github.com/hyperledger/fabric-samples.git**

**cd fabric-sample**

*Step 3:*

Download Image and Binaries

In the terminal type "**curl -sSL http://bit.ly/2ysbOFE | bash -s -- 1.4.0**"

*Step 5:*

Open another terminal and type/copy this command:

**cd ..**

**git clone https://github.com/khrandm/blockchain-training-labs**

after downloading the file view and extract it on the file manager. Inside blockchain-training-labs copy the chaincode and supply folder to fabric-samples directory. If the file already exists replace or merge it with the previous file.

After that go to the directory root directory in terminal by typing/pasting the command:

**cd**

Download the required library for our chaincode by typing the following in the terminal:

**go get github.com/golang/protobuf/proto**

**go get github.com/hyperledger/fabric/common/attrmgr**

**go get github.com/pkg/errors**

**go get github.com/hyperledger/fabric/core/chaincode/lib/cid**

Now open file manager go to Home/go/src/github.com and copy these three folders and paste it inside fabric-sample/chaincode:

hyperledger

pkg

golang

*Step 6:*

Running the Program

Type/paste the ff. commands in the terminal

**cd**

**cd fabric –samples/supply**

**./startFabric.sh**

**npm install**

**node enrollAdmin.sh**

**registerSupplier.sh**

**registerOem.sh**

**registerBank.sh**

**node app**

After entering the following commands you should see a “localhost:3000“ message in the terminal

Step 7:

Open postman in the Virtual Machine thereyou should see a GET untitled request; change the method from GET to POST

Now lets try a **raiseInvoice** transaction

In postman do a POST request in localhost:3000/invoice

In Body tab click x-www-form-urlencoded and then the bulk edit on the end of right side

Then paste this:

**invoicenumber:INVOICE6**

**billedto:OEM**

**invoicedate:02/08/19**

**invoiceamount:10000**

**itemdescription:KEYBOARD**

**goodreceived:False**

**ispaid:False**

**paidamount:0**

**repaid:False**

**repaymentamount:0**

now click the key-value-edit now we have the sample body request value

Click the send and it should return a success response and then add another request, now we are going to use the GET method on localhost:3000 to check if we succeed raising an invoice

Step 8:

Now were going to declare **goodreceived**

beside POST localhost:3000/invoice click the plus sign

change the method to PUT and localhost:3000/invoice

Go to header tab and add user with value of oem

Next go to body x-www-form-urlencoded

add these data:

**invoicenumber INVOICE001**

**goodreceived True**

Now click send; after this you should see result : success

Step 9:

Next is the **bank will now pay the supplier**

add another request PUT method and localhost:3000/invoice

on header tab add user with value of bank

now on body tab x-www-form

add these data:

**invoicenumber INVOICE001**

**paidamount 9000**

There are conditions here, the paid amount should be less than invoice amount

Now click send. There you should see **result : success**

Go to GET localhost:3000 tab and click send. Then check if the data is updated

The invoice will indicate that the **isPaid = true** and the **paidamount** will be **9000**

*Step 10:*

The **OEM will pay the bank**

add another request PUT method localhost:3000/invoice

on header tab add user with value of oem

now on body tab x-www-form

add the following data:

**invoicenumber INVOICE001**

**repaymentamount 11000**

there are conditions here, the repayment amount should be more than paidamount

now hit send

you should see the result : success

go to GET localhost:3000 tab then hit send

then check if data is updated

Step 12:

Lastly let’s **check invoice audit trail**

Add another GET request to localhost:3000

On header add user with value of supplier

Now on body x-www-form add invoicenumber with value of INVOICE001. Then click send

You should see the respond from the server change it from Html to Json to see a json format of the response

Keys

We have 3 users registered earlier during setup

supplier

oem

bank

those are values we used in header

user is the key

supplier,oem,bank is the value

we have conditions every user

supplier can only raise invoice

oem can only change the invoice if the good is received

oem can only pay the bank

bank can only pay the supplier