Franco, Al Jasper N.
Blockchain 100

Step 1: Download the Repository

https://github.com/khrandm/blockchain-training-labs

==
Step 2: Copy chaincode and supply folder to fabric-sample

Open file manager and go to the directory where blockchain-training-labs located
Copy chaincode folder and supply folder
Paste it inside the fabric-sample
Click merge if already exist
Go back to terminal
Type "cd supply"

Download the required library for our chaincode
Go language must be installed
type this to 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
copy three folders

hyperledger
pkg
golang

paste it inside fabric-sample/chaincode

**if the folder already exist in the fabric-samples/chain code. Idelete it using the terminal
sudo rm -R **folder name**
then paste it.

==
Step 3: Start the fabric

Type ./startFabric.sh
Type npm install
Type node enrollAdmin.sh
Type node registerSupplier.sh
Type node registerOem.sh
Type node registerBank.sh
Type node app

You should see a Example app listening on port 3000!

==
Step 4: Check if its running

open postman and you should see a GET untitled request
change method from GET to POST
add url localhost:3000/invoice
below url you should see Params Authroization Headers Body
click Headers
Add another key below Content-Type

type user
and the value would be "supplier"

next open the body tab
click the x-www-form-url-encoded click the **bulk edit**
Type

invoicenumber:INVOICE001
billedto:OEM
invoicedate:02/08/19
invoiceamount:10000
itemdescription:KEYBOARD
goodreceived:False
ispaid:False
paidamount:0
repaid:False
repaymentamount:0

Hit send
You should see result: Success
Now we have successfully raise an invoice

add another request GET method localhost:3000
on header add user with value of supplier
now hit send to see your invoices in the respond below

==
Step 5: 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 hit the send

you should see result : success

==
Step 6: Bank will 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

NOTE: there are conditions here, the paid amount should be less than invoice amount

hit send
you should see result : success

go to GET localhost:3000 tab then hit send
then check if data is updated
the invoice will idicate that the isPaid = true

and the paidamount will be 9000

==
Step 7: 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 these 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 8: lets check invoice audit trail
add another request GET localhost:3000
on header add user with value of supplier
now on body x-www-form add invoicenumber with value of INVOICE001

then hit 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