

Laptop Specification

Manufacturer: Acer

Unit: Acer Nitro AN515-52-56CQ

OS: Ubuntu 18.04.1 LTS

OS Type: 64 bit

CPU: Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 1 physical processor; 4 cores; 8 threads

Ram: 8GB

Kernel: Linux 4.15.0-43-generic (x86_64)

Installation

Open terminal and begin Installation by typing the following commands (commands are with yellow highlights).

Commands for installing prerequisites:

```
curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh
chmod u+x prereqs-ubuntu.sh
./prereqs-ubuntu.sh
```

Check the following by typing this:

```
docker --version
npm --version
git --version
python --version
node --version
```

Install Go:

```
wget https://storage.googleapis.com/golang/go1.9.2.linux-amd64.tar.gz && \
sudo tar -C /usr/local -xzf go1.9.2.linux-amd64.tar.gz && \
rm go1.9.2.linux-amd64.tar.gz && \
echo 'export PATH=$PATH:/usr/local/go/bin' | sudo tee -a /etc/profile && \
echo 'export GOPATH=$HOME/go' | tee -a $HOME/.bashrc && \
echo 'export PATH=$PATH:$GOROOT/bin:$GOPATH/bin' | tee -a $HOME/.bashrc && \
mkdir -p $HOME/go/{src,pkg,bin}
```

Then check go:

```
go version
```

Install vscode:

```
sudo snap install vscode --classic
```

Open vscode then go to view > extensions and search hyperledger composer then install it.

Run a Hyperledger application

MANDATORY

1. Open a terminal type the commands to clone the repository.
fabric-sample folder:

github.com/hyperledger/fabric-samples

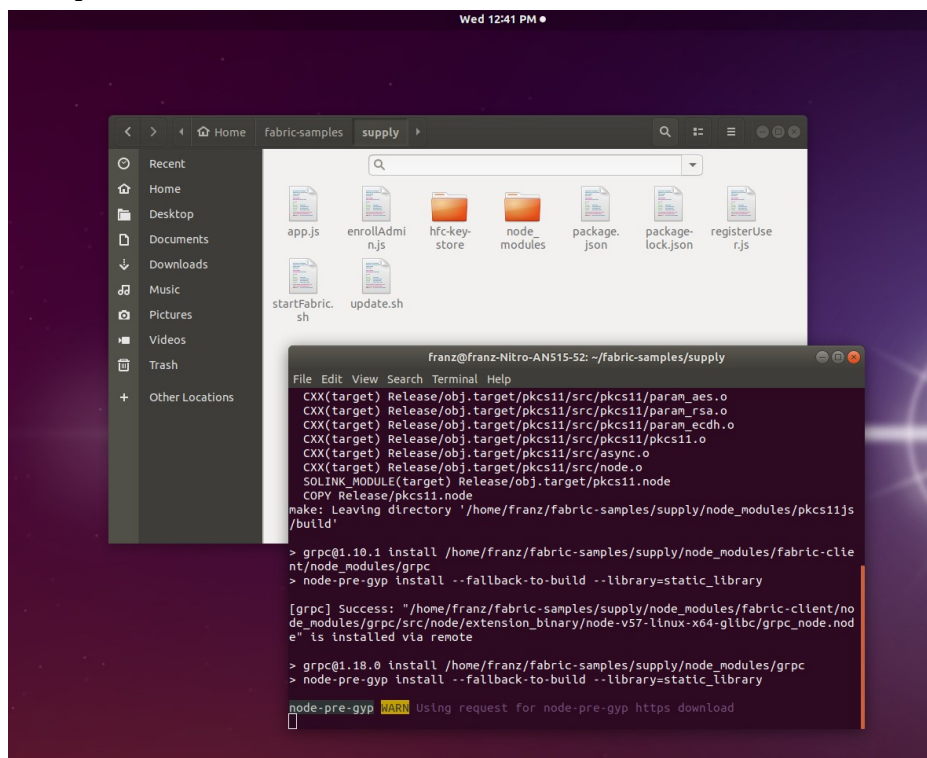
blockchain-training-labs:

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

2. Open the folder you cloned named **blockchain-training-labs** and copy the folder named **supply** and chaincode then paste it to the folder you also cloned named **fabric-samples** and merge existing files.

3. Open supply folder. Then, right click to open a terminal and type **npm install**.

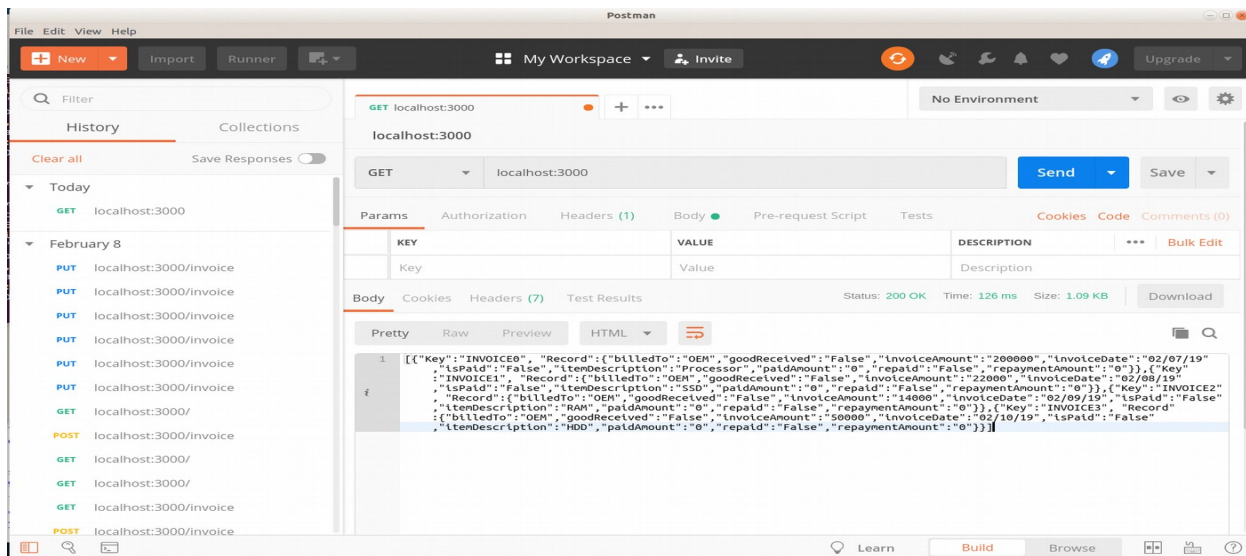
4. you will notice that it will stuck with the WARN like in the picture below. Just press “ Ctrl + C” to cancel the current process.



5. Run the file **startFabric.sh** by typing `./startFabric.sh` in the terminal. (Notice the output it should be like the picture below)

10. search in you ubuntu/pc **ubuntu software** (press windows key and search ubuntu software) after opening it search postman and install.

11. Open postman and send



12. Use the Post HTTP Method. In this function it will request to server to accept the data enclosed in the body of the request message.

For example we will use these following fields and their specific value.

invoicenumber: INVOICE6

billedto: OEM

invoicedate: 02/08/19

invoiceamount: 10000

itemdescription: KEYBOARD

goodreceived: False

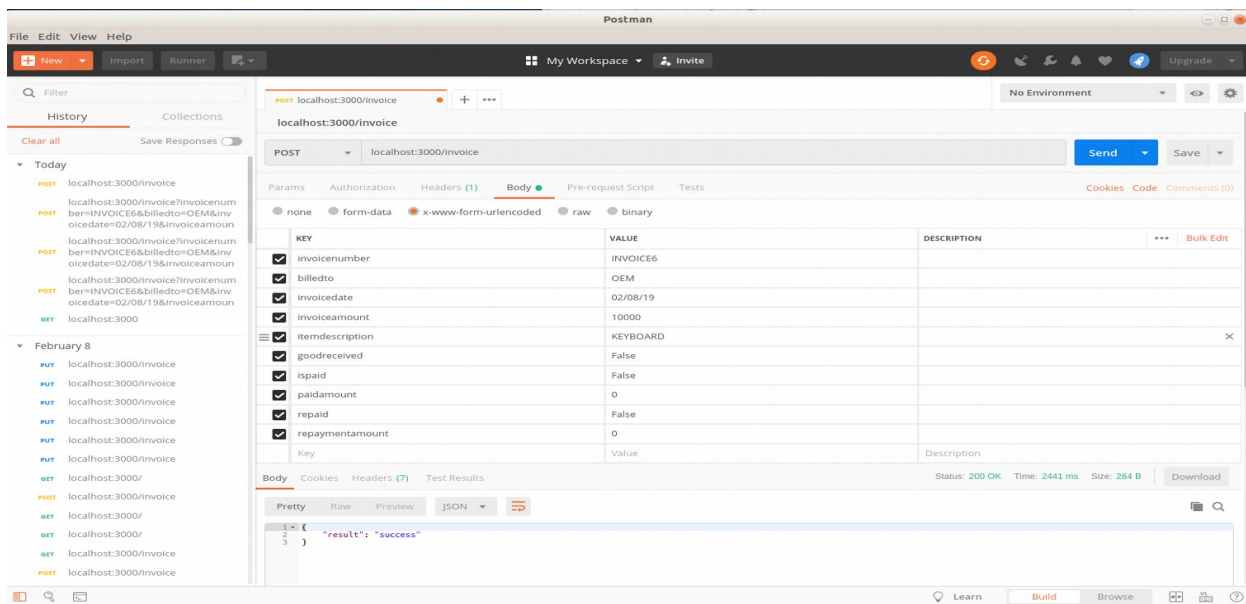
ispaid: False

paidamount: 0

repaid: False

repaymentamount: 0

The result below will look like these:



Your terminal should like these:

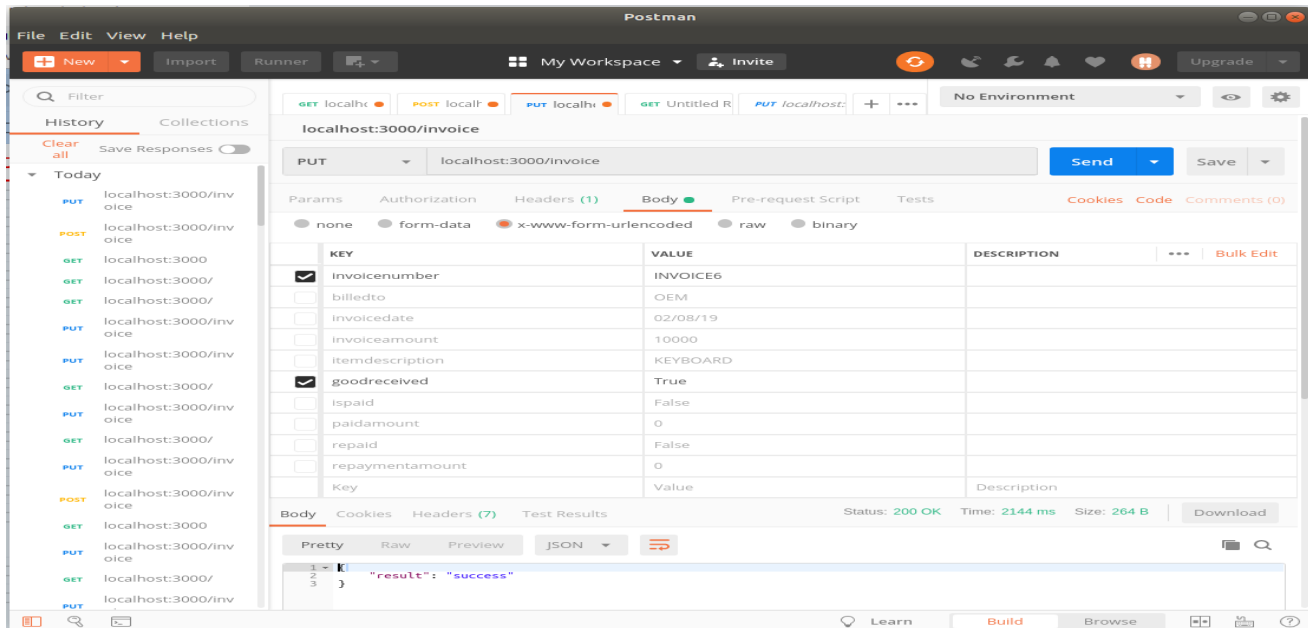
```

jhun@jhun-FX503VD: ~/fabric-samples/supply
File Edit View Search Terminal Help
Successfully loaded user1 from persistence
Assigning transaction_id: 8fdd2e3a5639d10ddbff72c6cfd16752286f5bb4922488720343ddc81af3b8e2
----- { chaincodeId: 'supply',
  chainId: 'mychannel',
  txId:
    TransactionID {
      _nonce: <Buffer cf 8b d6 26 8a e6 33 d6 e7 e7 ff 1c ca 0d 00 fc bf 37 29 02
87 e9 68 66>,
      _transaction_id: '8fdd2e3a5639d10ddbff72c6cfd16752286f5bb4922488720343ddc81
af3b8e2',
      _admin: false },
  fcn: 'raiseInvoice',
  args:
    [ 'INVOICE6',
      'OEM',
      '02/08/19',
      '10000',
      'KEYBOARD',
      'False',
      'False',
      '0',
      'False',
      '0' ] }
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message
- ""
The transaction has been committed on peer localhost:7051
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer

```

The transaction has been sent in your peer. A new invoice has been created!

13. Use the PUT HTTP Method. In this function as we are modifying a data Select x-www-form URL Encoded as a structure

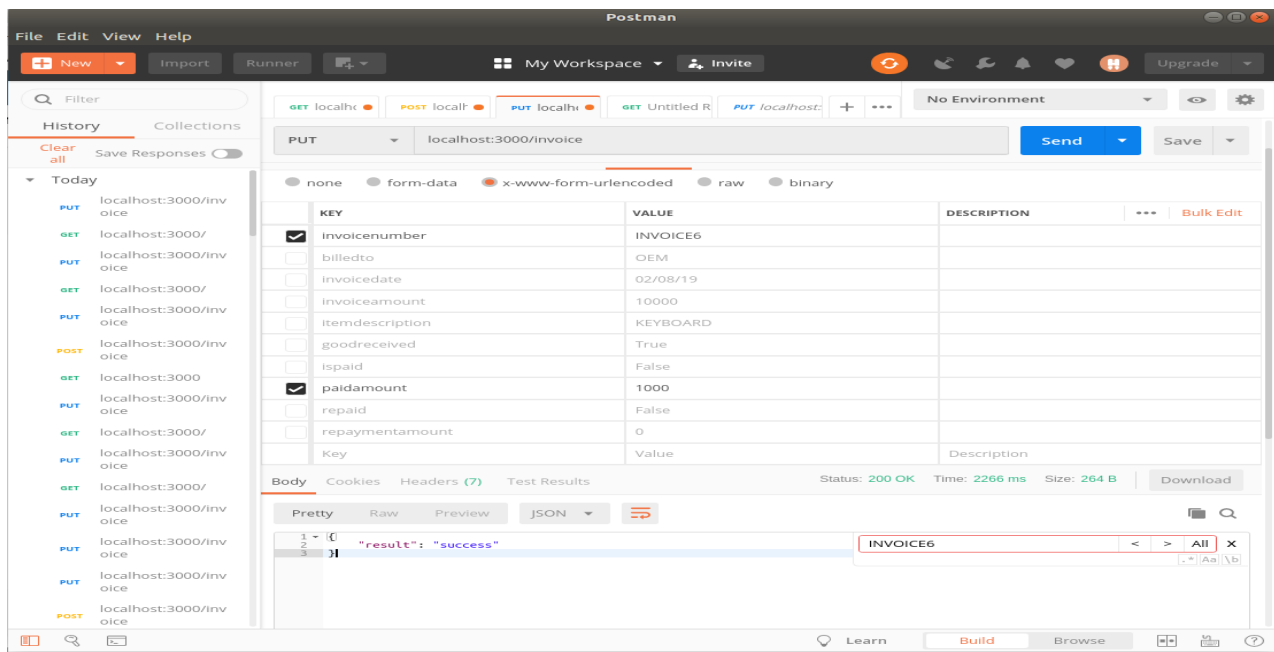


And you will see an output in your terminal like these:

```
jhun@jhun-FX503VD: ~/fabric-samples/supply
File Edit View Search Terminal Help
Successfully loaded user1 from persistence
Assigning transaction_id: e6786af730a52ec437edc9487bd09260dd443d546ee044503b4f5eab1f28048b
----- { chaincodeId: 'supply',
  chainId: 'mychannel',
  txId:
    TransactionID {
      _nonce: <Buffer 2b f4 93 72 29 b5 b9 0f 88 6d cc 12 fa 9d 35 77 e8 f3 a2 83 ec 5f 24 1f
>,
      _transaction_id: 'e6786af730a52ec437edc9487bd09260dd443d546ee044503b4f5eab1f28048b',
      _admin: false },
      fcn: 'goodReceived',
      args: [ 'INVOICE6', 'True' ] }
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message - ""
The transaction has been committed on peer localhost:7051
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
```

14. Use the PUT HTTP Method. In this function as we are modifying a data Select x-www-form URL Encoded as a structure

Unchecked all the fields except to **invoicenumber** and **paidamount**. Then click send and it should like these.

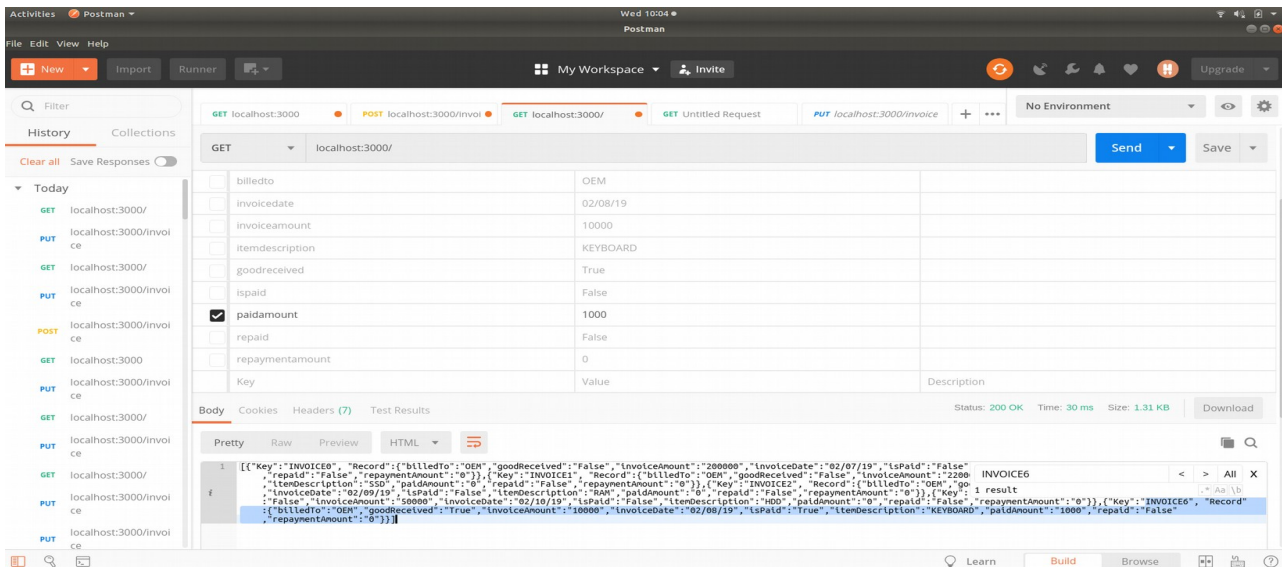


And you will see an output in your terminal like these:

```
jhun@jhun-FX503VD: ~/fabric-samples/supply
File Edit View Search Terminal Help
Successfully loaded user1 from persistence
Assigning transaction_id: f7e49db39cc797629488540c7c7fd5aca864d536d0ab700d1bc1a5f1f905d37d
----- { chaincodeId: 'supply',
  chainId: 'mychannel',
  txId:
    TransactionID {
      _nonce: <Buffer 7e b0 77 f5 29 40 f1 ef de 07 d6 8f ff fe f6 be 7a 42 62 38
50 7a 67 53>,
      _transaction_id: 'f7e49db39cc797629488540c7c7fd5aca864d536d0ab700d1bc1a5f1f
905d37d',
      _admin: false },
      fcn: 'bankPayment',
      args: [ 'INVOICE6', '1000' ] }
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message
- ""
The transaction has been committed on peer localhost:7051
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
```

In your postman the result will be something like these:

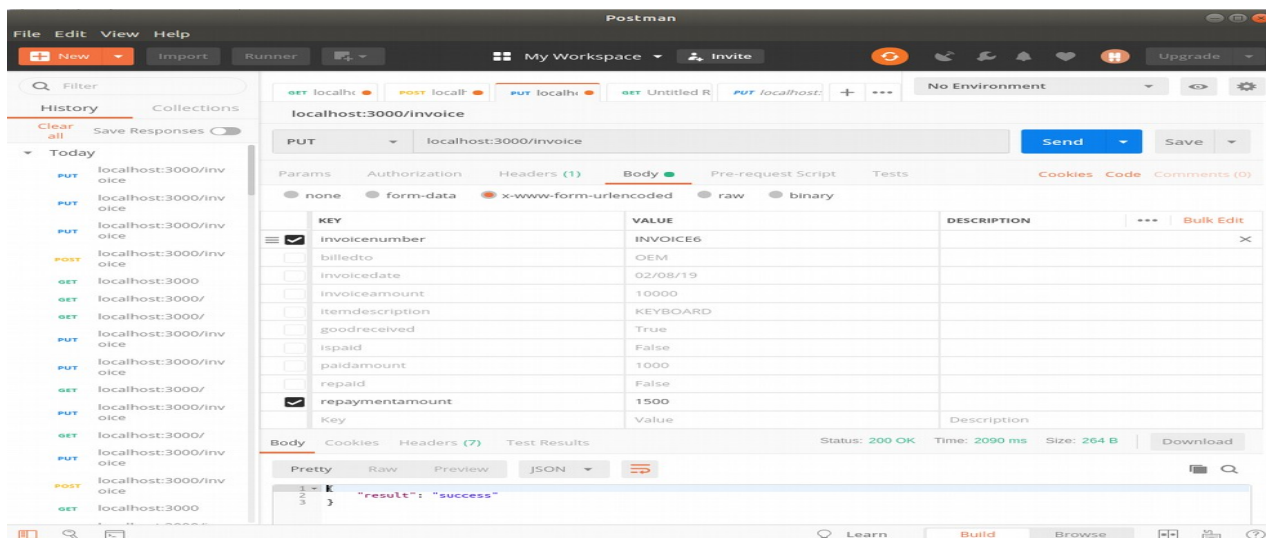
Use the GET HTTP Method then type “localhost:3000” and add value to the **paidamount** key.
Here is the example.



As you can see in the results below the value of the field **isPaid** have changed from “False” to “True” it means that the product has been paid.

15. Use the PUT HTTP Method. In this function as we are modifying a data
Select x-www-form URL Encoded as a structure

Unchecked all the fields except to **invoicenumber** and **repaymentamount**. Then click send and it should like these.



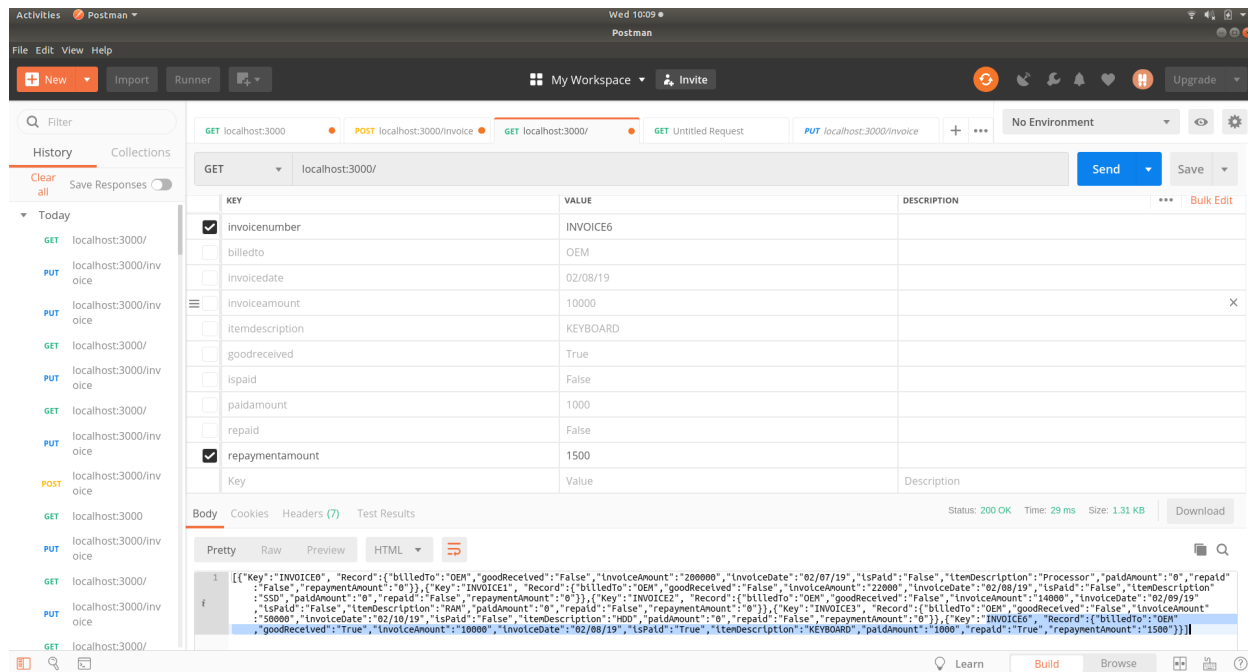
And you will see an output in your terminal like these:

```
jhun@jhun-FX503VD: ~/fabric-samples/supply
File Edit View Search Terminal Help
Successfully loaded user1 from persistence
Assigning transaction_id:  ad889600678292b6f362661b2b51e8c12c6ea25753afd2bf095660f8bf7df42a
----- { chaincodeId: 'supply',
  chainId: 'mychannel',
  txId:
    TransactionID {
      nonce: <Buffer 63 73 da fa 66 5f 41 ca 20 e9 24 d2 ab e5 2f 4a 9f 54 5a 43 0a 4d d3 28>,
      transaction_id: 'ad889600678292b6f362661b2b51e8c12c6ea25753afd2bf095660f8bf7df42a',
      admin: false },
      fcn: 'oemPayment',
      args: [ 'INVOICE6', '1500' ] }
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status - 200, message
- " "
The transaction has been committed on peer localhost:7051
Send transaction promise and event listener promise have completed
Successfully sent transaction to the orderer.
Successfully committed the change to the ledger by the peer
```

In
your

postman the result will be something like these:

Use the GET HTTP Method then type “localhost:3000” and add value to the **repayment** key. Here is the example.



Notice the value of **repaid** if it change from False to True.

OPTIONAL

Step 1

Go back to Terminal type:

```
cd
cd fabric-samples
docker kill $(docker ps -q)
docker rm $(docker ps -aq)
docker rmi $(docker images dev-* -q)
cd supply
```

Then:

```
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

Type:

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

You should see a localhost:3000

In postman change method from **GET** to **POST**

below url in **Params** | **Authroization** | **Headers** | **Body**

click **Headers**

check if key below is: (correct it if wrong)

Content-Type		Value
user		supplier

next go to the **Body** tab

click the x-www-form-url-encoded

you should see a form there and find from the right side the **bulk edit (orange color)** and click then copy and paste this: (if there is a letters or words just replace it)

```
invoicenumber:INVOICE6
billedto:OEM
invoicedate:02/08/19
invoiceamount:10000
```

itemdescription:KEYBOARD
goodreceived:False
ispaid:False
paidamount:0
repaid:False
repaymentamount:0

Hit send and You should see result **Success** at the bottom

Now we have successfully raise an invoice

add another request GET method localhost:3000 (or just change the **GET** to **POST** and delete **/invoice** from localhost:3000/invoice)

now hit send to see your invoices in the respond below

Step 2

Beside POST localhost:3000/invoice click the plus sign

change the method to **PUT** and localhost:3000/invoice

Go to **Headers** tab and just like earlier add **user** with value of **oem**

then go to **Body** click x-www-form-urlencoded

add these data

KEY	VALUE
invoicenumber	INVOICE001
goodreceived	True

Now hit the send

you should see result : **“success”**

Step 3

click the plus sign again to add another request **PUT** method and localhost:3000/invoice

on **Header** tab add **user** with value of **bank**

on **Body** tab x-www-form-urlencoded

add these data:

invoicenumber	INVOICE001
paidamount	9000

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 indicate that the isPaid = true

and the paidamount will be 9000

DONE!

