Rodrigo S. Cruz Jr.
Blockchain Cadet

Laptop Model	Asus FX503VD.303
CPU: Intel Core i5-7300HQ	Intel Core i5-7300HQ
GPU:	NVIDIA GeForce GTX 1050 (2GB GDDR5)
RAM:	8GB
Operating System:	Ubuntu 18.04.1 LTS

For this activity we need to download and install the ff:

Installation of Hyperledger Fabric prerequisites
Open your terminal (Copy and Paste in this commands)

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

Install Node (in terminal)

- nvm install v8 (To download and install node version 8)
- nvm use 8 (To switch to node version 8)
- node –version (To check the version of the node your using)

Install Go Language

- Download Go for Linux in this url https://golang.org/dl/
- Go to the directory where the file located.
- Right click inside the folder and Open the Terminal
- Type "tar -C /user/local -xzf go1.11.5.linux-amd64.tar.gz"

Add Go Environment variable

- export PATH=$PATH:/usr/local/go/bin
- export GOPATH=$HOME/go
- export PATH=$PATH:$GOPATH/bin

Install Docker

https://www.youtube.com/watch?v=hY34PpllKf4
you can follow the video steps in the video

Install the prerequisites for docker (in terminal)

- sudo apt install \
- apt-transport-https \
- ca-certifcates \
- curl \
- software-properties-common

Installing the development environment for Hyperledger

Installing components
(open terminal)

Step 1 : Install the CLI tools
Essential CLI tools:
npm install -g composer-cli@0.20.5

Utility for running a REST Server on your machine to expose your business networks as RESTful APIs:

npm install -g composer-rest-server@0.20.5
Useful utility for generating application assets:

npm install -g generator-hyperledger-composer@0.20.5

Yeoman is a tool for generating applications, which utilises generator-hyperledger-composer
npm install -g yo

Step 2: Install Playground
Browser app for simple editing and testing Business Networks:
npm install -g composer-playground@0.20.5 (latest version of composer-playground)

Step 3: Install Hyperledger Fabric
mkdir ~/fabric-dev-servers && cd ~/fabric-dev-servers
curl -O https://raw.githubusercontent.com/hyperledger/composer-tools/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz
tar -xvf fabric-dev-servers.tar.gz

cd ~/fabric-dev-servers
export FABRIC_VERSION=hlfv12
./downloadFabric.sh

To start hyperledger composer
in terminal type in
composer-playground

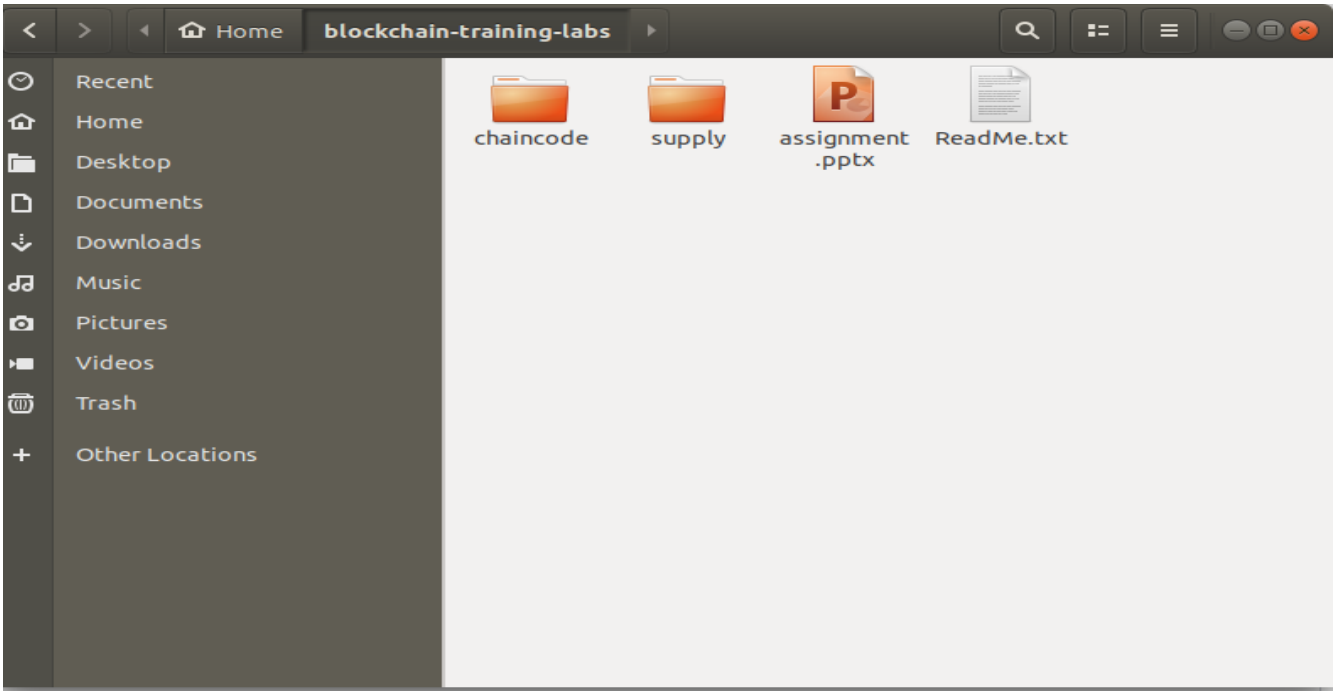
Hyperledger Activity Mandatory and Optional Assignment

Step 1. Clone the two repository from Github

git clone https://github.com/hyperledger/fabric-samples
git clone https://github.com/xjhuncruzx/blockchain-training-labs.git

Result : This will create two folders named **fabric-samples** and **blockchain-training-labs**

Step 2.:Open your **blockchain-training-labs**



Copy the **chaincode** and **supply** folder to your **fabric-samples** folder

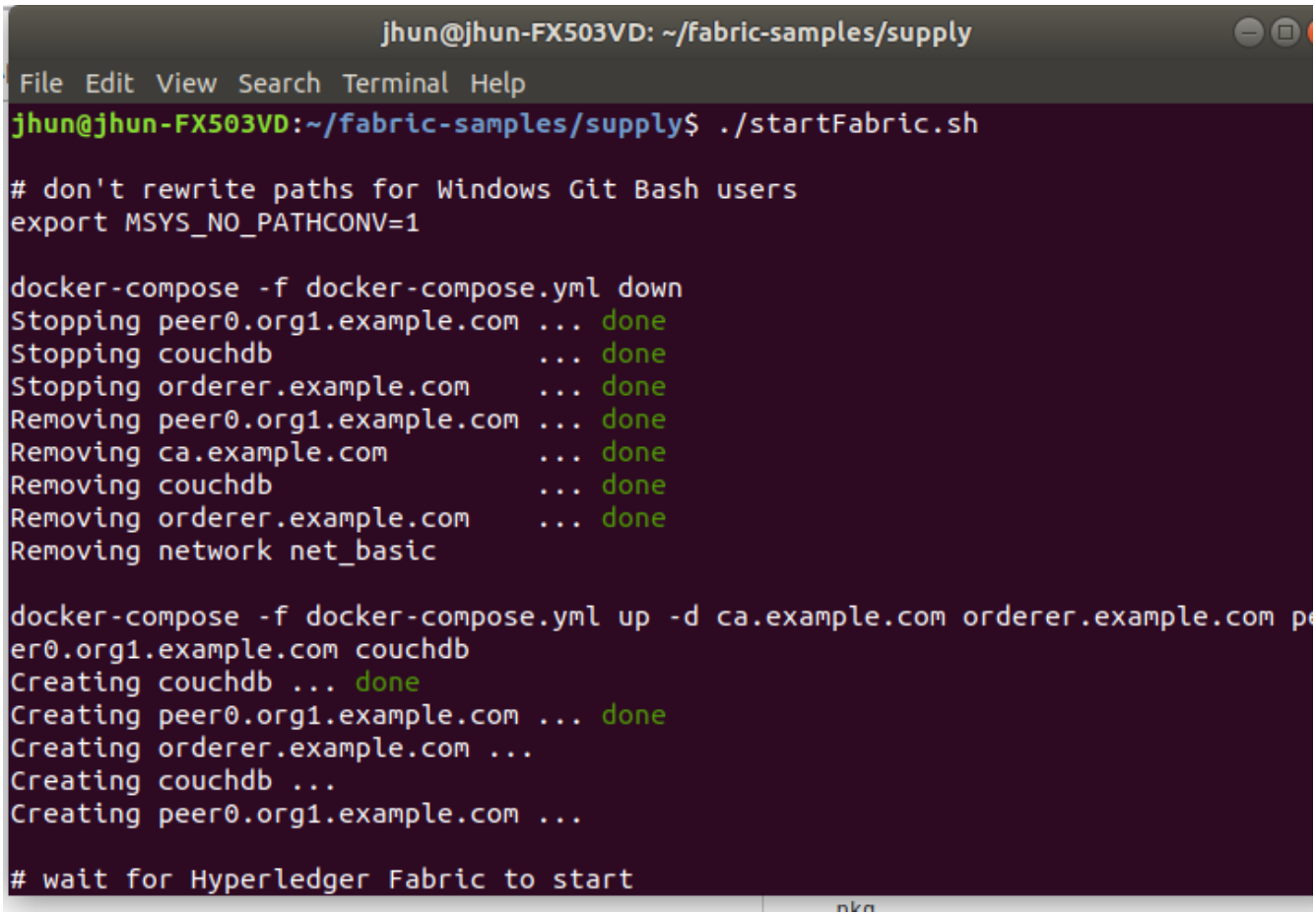
Step 3: Type this your terminal , This will download the required library for your chaincode

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

Copy **golang** , **hyperledger** and **pkg** to your **fabric-samples/chaincode** folder

Step 4: Type this command

cd fabric-samples/supply (Will move you to your supply folder inside fabric-samples)
./startFabric.sh (this will create peers for the network)



Step 5 npm install (install the list of dependencies)
Step 6 Update your chaincode
docker exec -it cli bash
peer chaincode install -n supply -v 1.1 -l "golang" -p"github.com/supply/go"
peer chaincode upgrade -n supply -v 1.1 -o orderer.example.com:7050 -C mychannel -l "golang" -p "github.com supply/go" -c '{"Args":[""]}' -P "OR ('Org1MSP.member','Org2MSP.member')"
Step 7 node enrollAdmin.js
Step 8 node registerSupplier.js
Step 9 node registerOEM.js
Step 10 node registerBank.js
Step 11 node app.js (to run the program)

Step 11 Open your postman (Testing if its working)
if you don't have postman (in the **Ubuntu software** search and download Postman)

Select and type in the url "localhost:3000/invoice
choose the method GET click Send

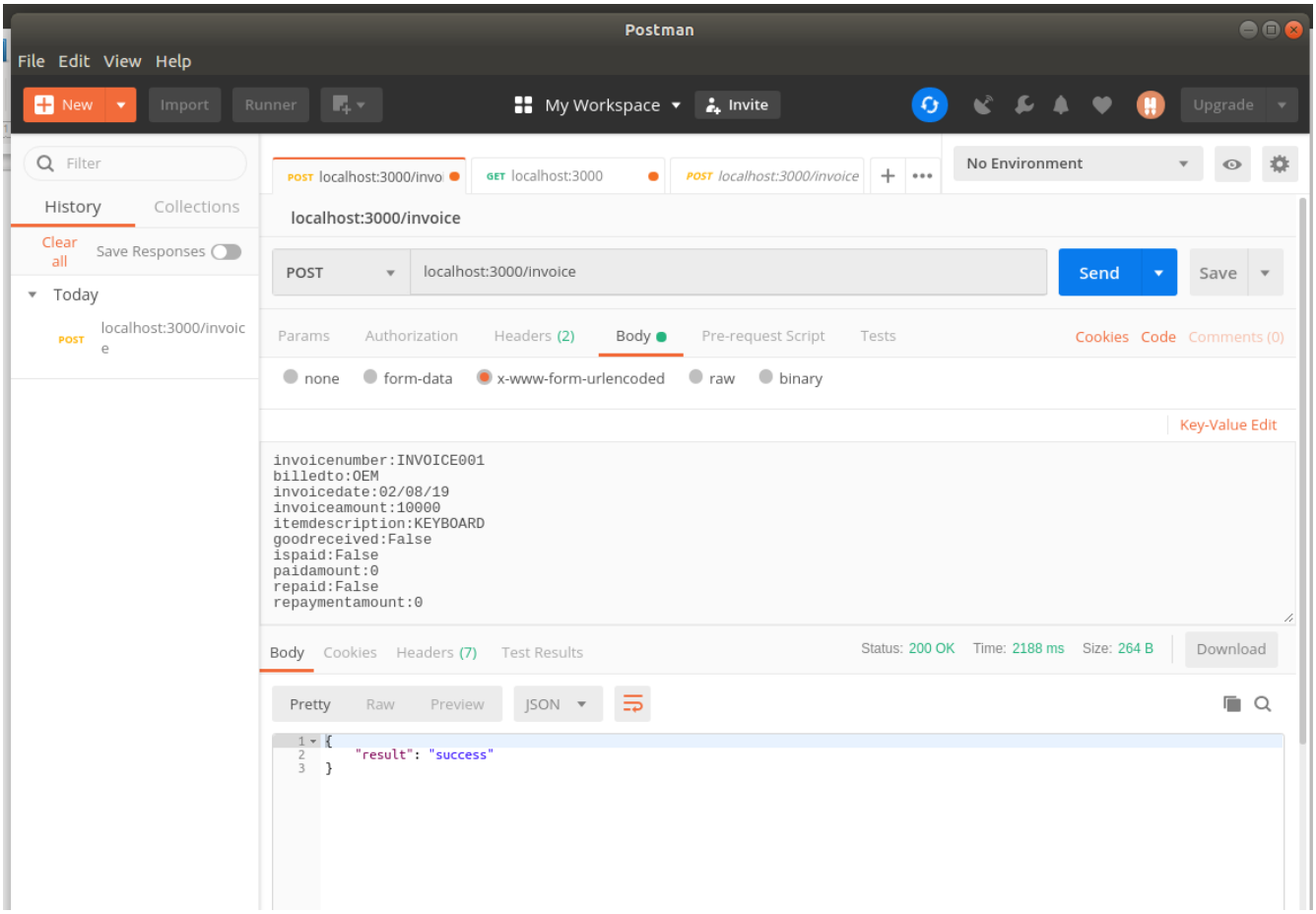
This will let you push data to your postman

Step 12 Raise an invoice

Change the method from GET to POST
in your url change your localhost:3000 into localhost:3000/invoice
Click on the Body Tab
Select the x-www-form-url-encoded
Click Bulk and Edit
Copy these data values

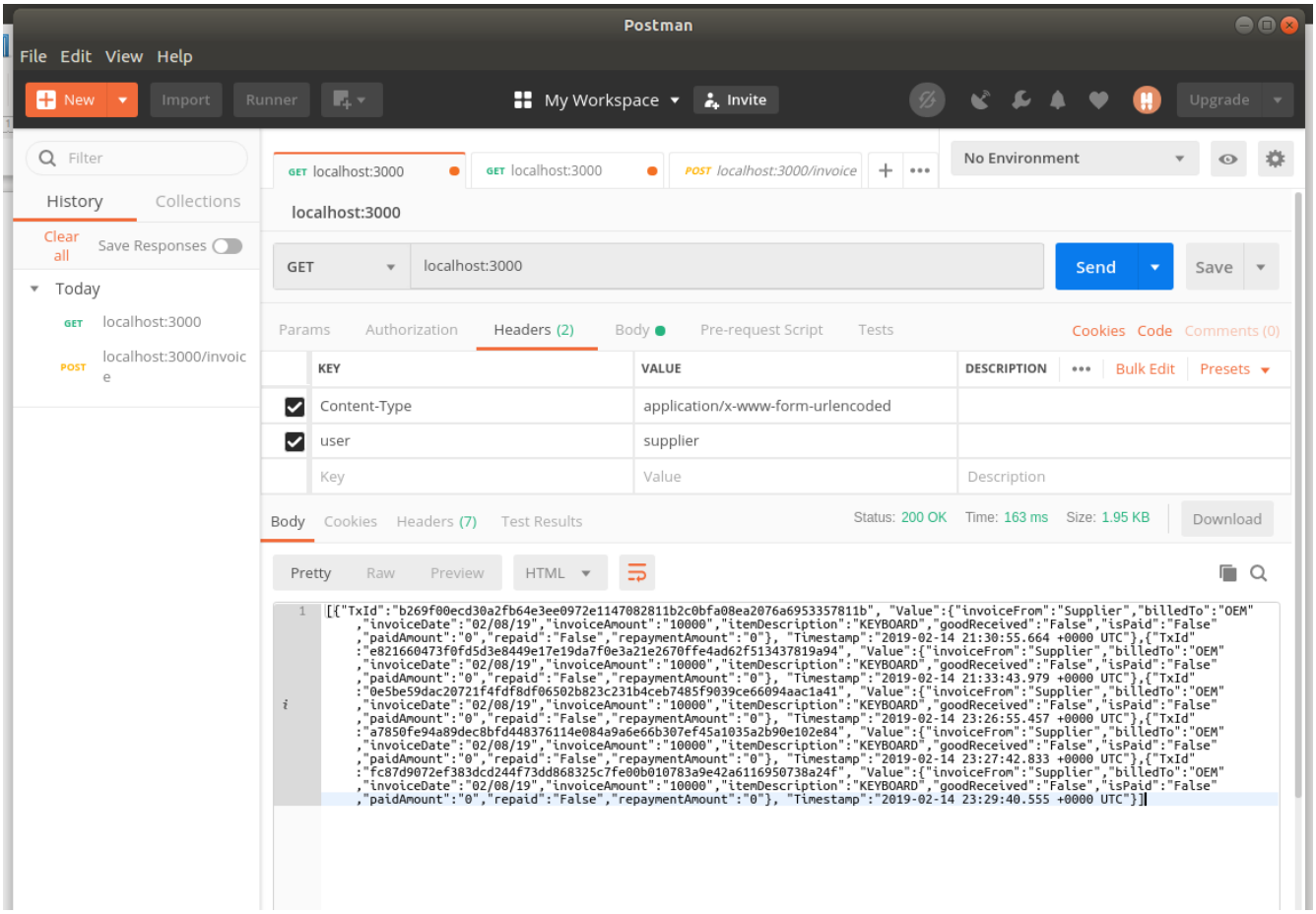
invoicenumber:INVOICE001
billedto:OEM
invoicedate:02/08/19
invoiceamount:10000
itemdescription:KEYBOARD
goodreceived:False
ispaid:False
paidamount:0
repaid:False
repaymentamount:0

Click Send
You will see the result: Success
Result: You have sucessfully raised an invoice



To view your invoice

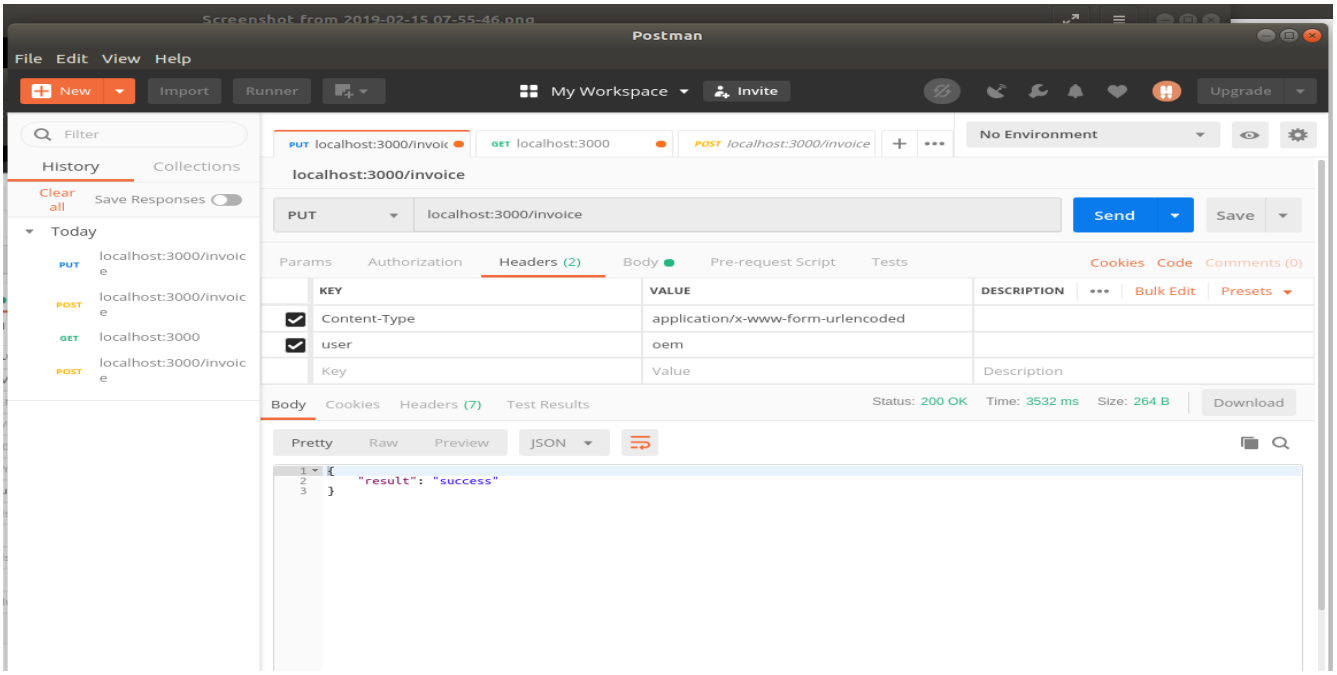
Select GET then type in your the url "localhost:3000/"
Select Headers Tab
Under the Content Type key, add the key "**user**"
Add the value "**supplier**"
Click the Send button.

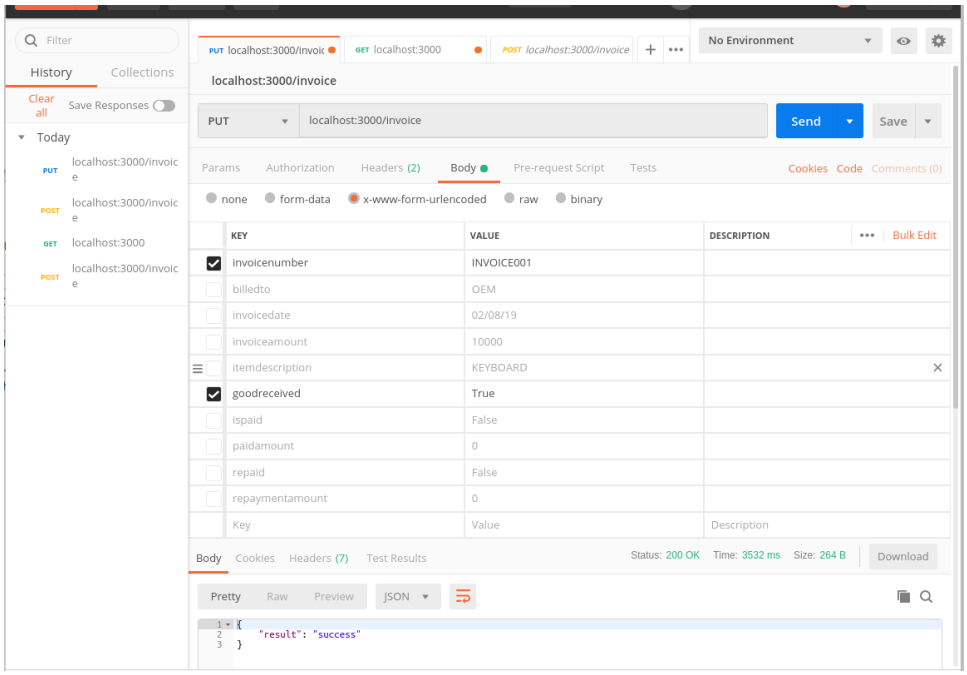


Step 13: **Declare goodreceived**

Select PUT then type the url "localhost:3000/invoice"
Select Headers Tab.
Change the value of user into **oem**
Select the Body Tab
Select the x-www-form-url-encoded
Uncheck everything except **invoicenumber** and **goodreceived**

make the value of **goodreceived** into **True**
then click send You will the result "**Success**"

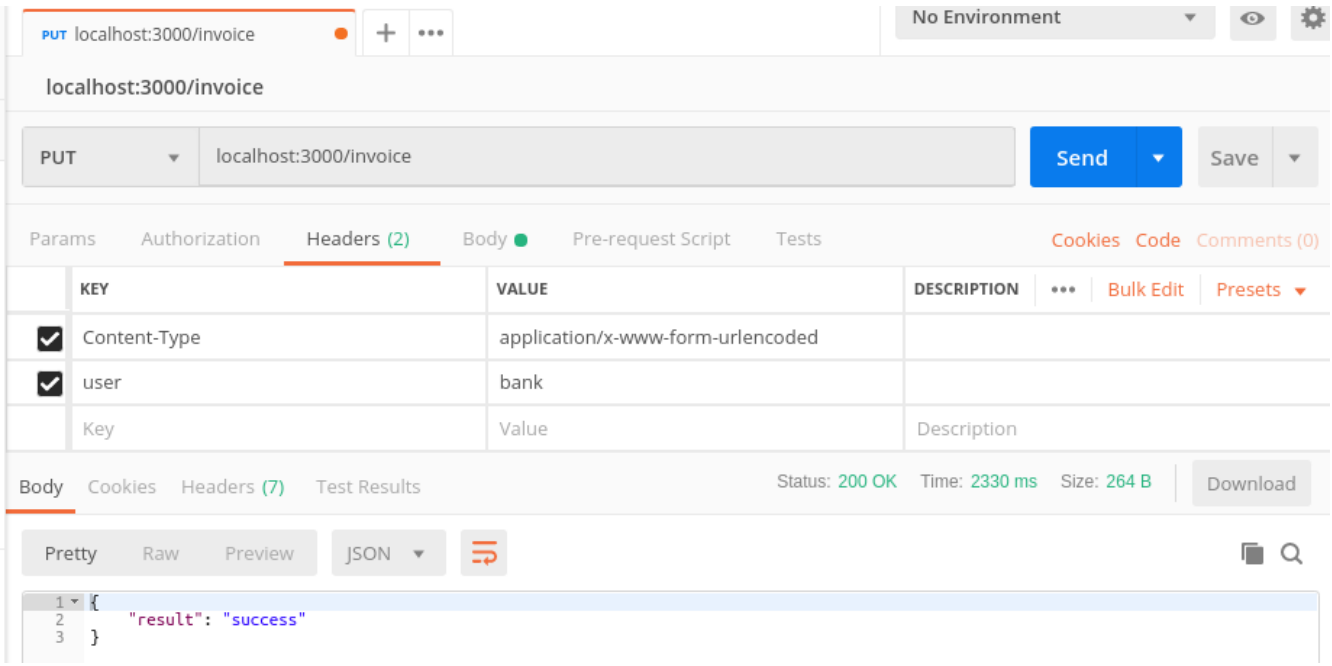


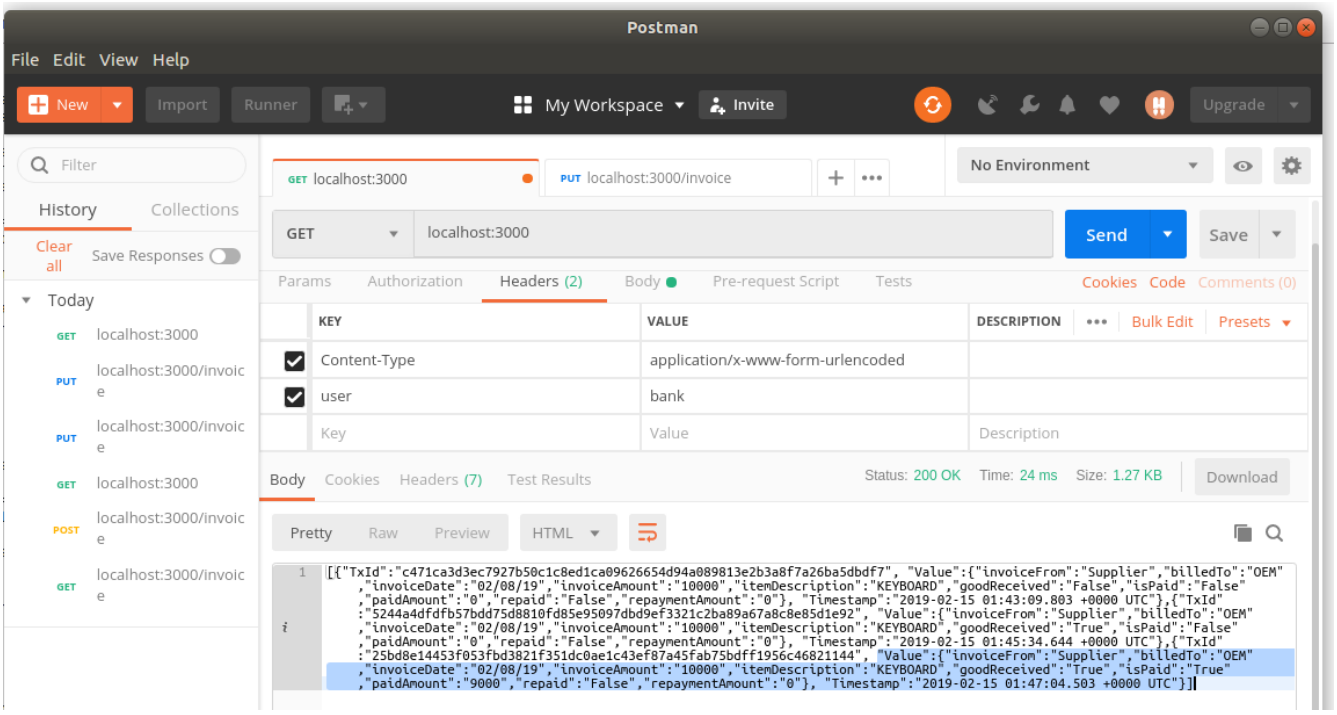


Step 14: **Bank will pay the supplier**

Select PUT then type the url "localhost:3000/invoice"
Select Headers Tab.
Change the value of **user** into **bank**
Select the Body Tab
Select the x-www-form-url-encoded
Uncheck everything except invoicenumber and paidamount

make the value of **paidamount** into **9000**
then click send You will the result "**Success**"



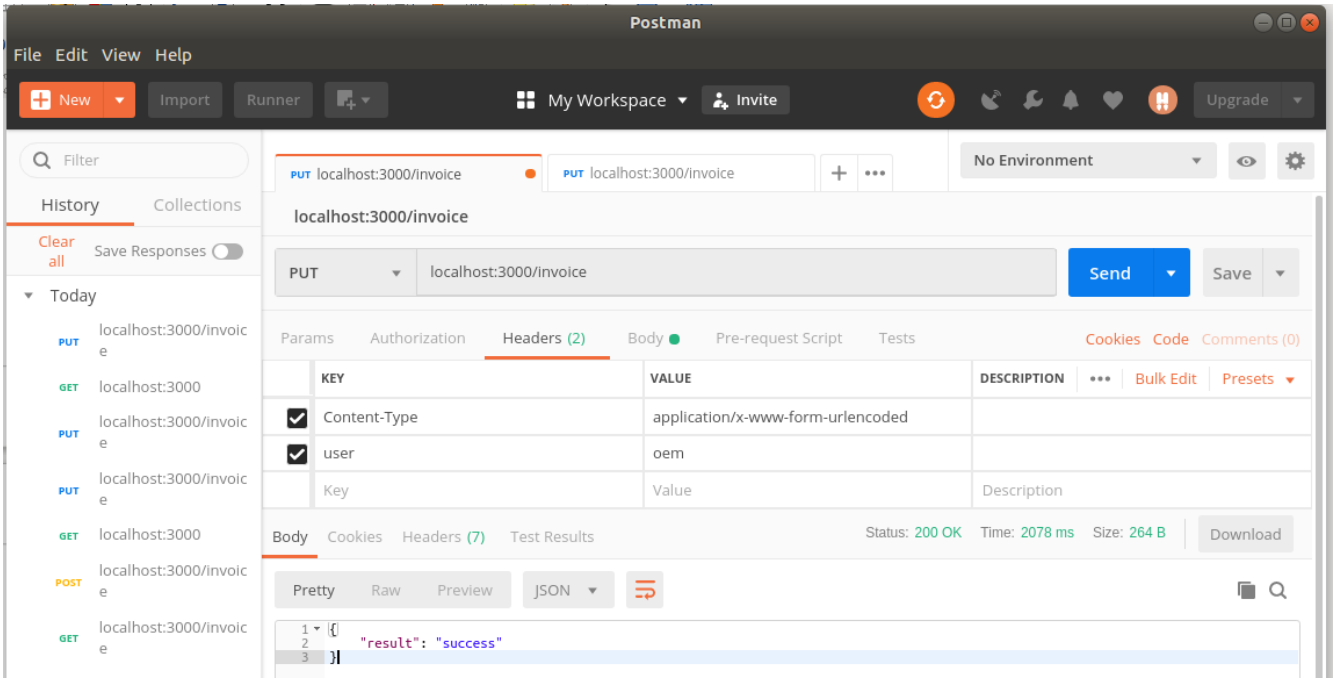


Check the data if changed
Use GET method then make the url into localhost:3000
Then click Send
The invoice will indicate **isPaid**: **True**
and the paid amount will be **9000**

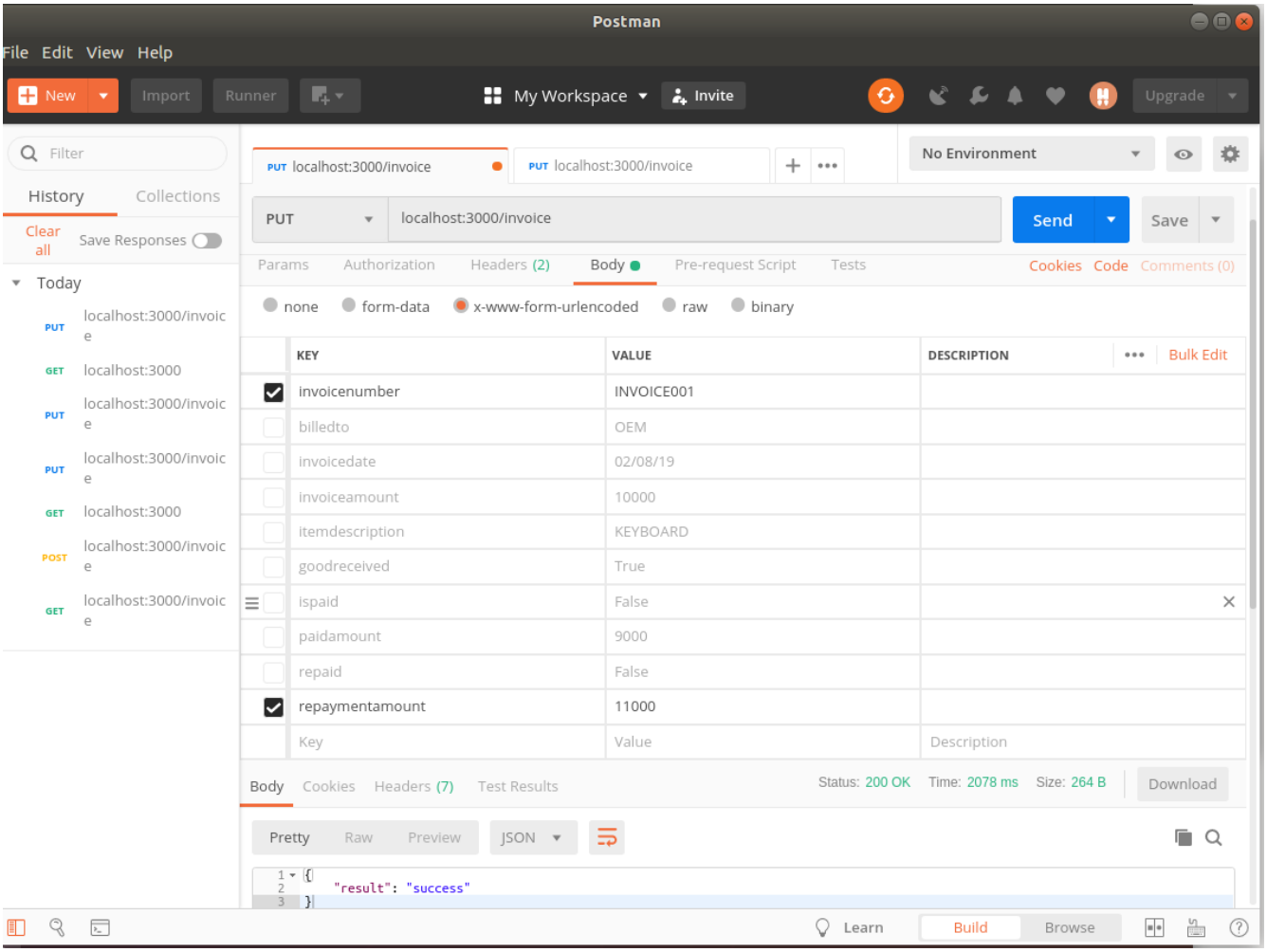
Step 15 **OEM will pay the bank**

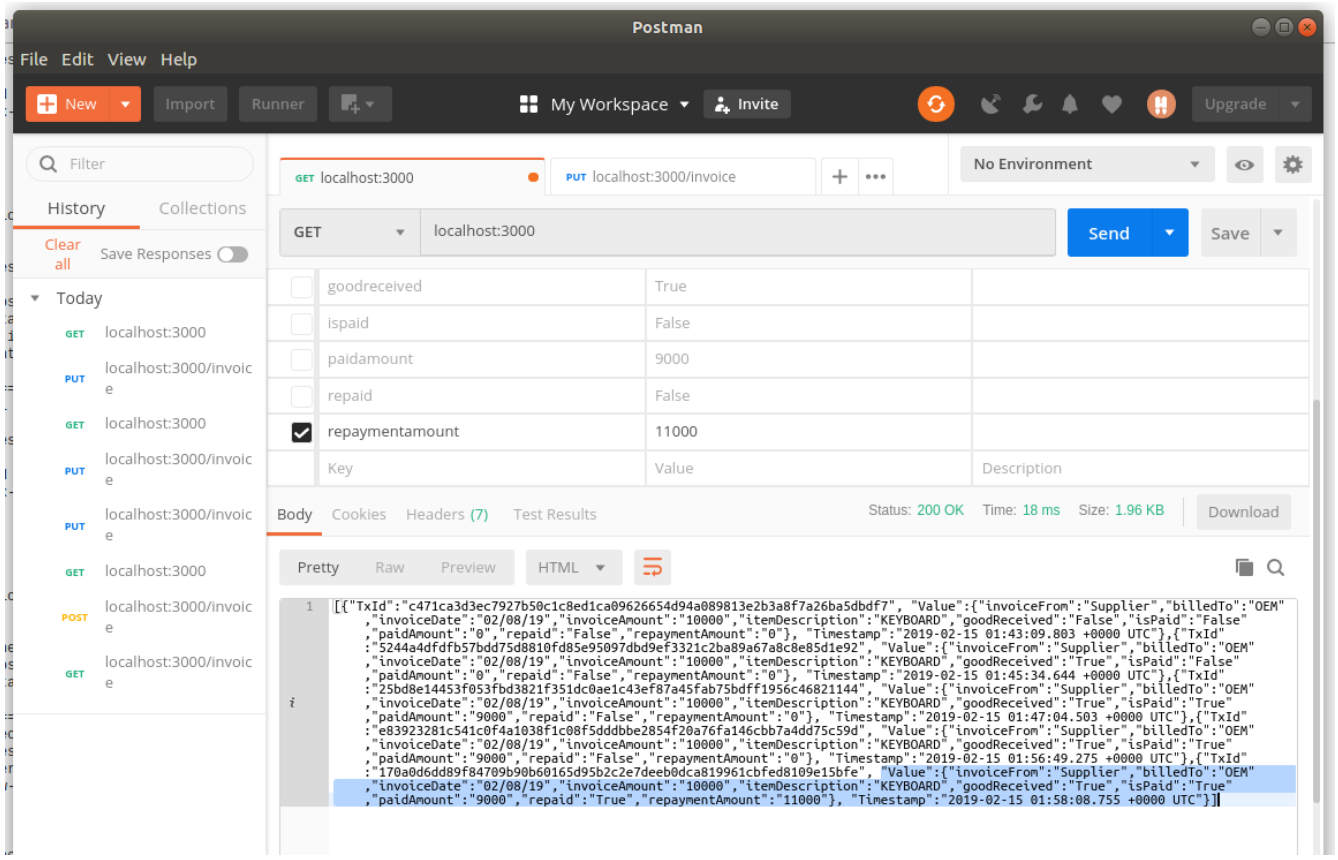
Select PUT then type the url "**localhost:3000/invoice**"
Select Headers Tab.
Change the value of **user** into **oem**
Select the Body Tab
Select the x-www-form-urlencoded
Uncheck everything except **invoicenumber** and **repaymentamount**

make the value of **repaymentamount** into **11000**



Check the data if changed
Use GET method then make the url into localhost:3000
Then click Send
The invoice will indicate **repaid**: **True**
and the repaymentamount will be 11000


Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
Clear all
Save Responses
Today
PUT localhost:3000/invoice
GET localhost:3000
PUT localhost:3000/invoice
PUT localhost:3000/invoice
GET localhost:3000
POST localhost:3000/invoice
GET localhost:3000/invoice
PUT localhost:3000/invoice
PUT localhost:3000/invoice
No Environment
PUT
localhost:3000/invoice
Send
Save
Params
Authorization
Headers (2)
Body
Pre-request Script
Tests
Cookies
Code
Comments (0)
none
form-data
x-www-form-urlencoded
raw
binary
KEY
VALUE
DESCRIPTION
Bulk Edit
invoicenumber
INVOICE001
billedto
OEM
invoicedate
02/08/19
invoiceamount
10000
itemdescription
KEYBOARD
goodreceived
True
ispaid
False
paidamount
9000
repaid
False
repaymentamount
11000
Key
Value
Description
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 2078 ms
Size: 264 B
Download
Pretty
Raw
Preview
JSON
{
"result": "success"
}
Learn
Build
Browse


Postman
File Edit View Help
New
Import
Runner
My Workspace
Invite
Upgrade
Filter
History
Collections
Clear all
Save Responses
Today
GET localhost:3000
PUT localhost:3000/invoice
GET localhost:3000
PUT localhost:3000/invoice
PUT localhost:3000/invoice
GET localhost:3000
POST localhost:3000/invoice
GET localhost:3000/invoice
No Environment
GET
localhost:3000
Send
Save
goodreceived
True
ispaid
False
paidamount
9000
repaid
False
repaymentamount
11000
Key
Value
Description
Body
Cookies
Headers (7)
Test Results
Status: 200 OK
Time: 18 ms
Size: 1.96 KB
Download
Pretty
Raw
Preview
HTML
[{"TxId":"c471ca3d3ec7927b50c1c8ed1ca09626654d94a089813e2b3a8f7a26ba5dbdf7", "Value":{"invoiceFrom":"Supplier","billedTo":"OEM"
,"invoiceDate":"02/08/19","invoiceAmount":"10000","itemDescription":"KEYBOARD","goodReceived":"False","isPaid":"False"
,"paidAmount":"0","repaid":"False","repaymentAmount":"0"}, "Timestamp":"2019-02-15 01:43:09.803 +0000 UTC"},{"TxId"
:"5244a4dfdfb57bdd75d8810fd85e95097dbd9ef3321c2ba89a67a8c8e85d1e92", "Value":{"invoiceFrom":"Supplier","billedTo":"OEM"
,"invoiceDate":"02/08/19","invoiceAmount":"10000","itemDescription":"KEYBOARD","goodReceived":"True","isPaid":"False"
,"paidAmount":"0","repaid":"False","repaymentAmount":"0"}, "Timestamp":"2019-02-15 01:45:34.644 +0000 UTC"},{"TxId"
:"25bd8e14453f053fbd3821f351dc0ae1c43ef87a45fab75bdff1956c46821144", "Value":{"invoiceFrom":"Supplier","billedTo":"OEM"
,"invoiceDate":"02/08/19","invoiceAmount":"10000","itemDescription":"KEYBOARD","goodReceived":"True","isPaid":"True"
,"paidAmount":"9000","repaid":"False","repaymentAmount":"0"}, "Timestamp":"2019-02-15 01:47:04.503 +0000 UTC"},{"TxId"
:"e83923281c541c0f4a1038f1c08f5ddddbbe2854f20a76fa146cbb7a4dd75c59d", "Value":{"invoiceFrom":"Supplier","billedTo":"OEM"
,"invoiceDate":"02/08/19","invoiceAmount":"10000","itemDescription":"KEYBOARD","goodReceived":"True","isPaid":"True"
,"paidAmount":"9000","repaid":"False","repaymentAmount":"0"}, "Timestamp":"2019-02-15 01:56:49.275 +0000 UTC"},{"TxId"
:"170a0d6dd89f84709b90b60165d95b2c2e7deeb0dca819961cbfed8109e15bfe", "Value":{"invoiceFrom":"Supplier","billedTo":"OEM"
,"invoiceDate":"02/08/19","invoiceAmount":"10000","itemDescription":"KEYBOARD","goodReceived":"True","isPaid":"True"
,"paidAmount":"9000","repaid":"True","repaymentAmount":"11000"}, "Timestamp":"2019-02-15 01:58:08.755 +0000 UTC"}]

Step 16 Check Invoice Audit Trail

Select PUT then type the url "**localhost:3000/invoice**"
Select Headers Tab.
Change the value of **user** into **supplier**
Select the Body Tab
Select the x-www-form-url-encoded
Uncheck everything except **invoicenumber**

Then Click Send

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