

Creating Invoice using Hyperledger Fabric | Mandatory

GO Version: go version go1.11.5 linux/amd64

OS: Ubuntu 18.04 LTS

Installing the prerequisites

Step 1

1. Download Go Language in this link <https://golang.org/doc/install> and follow the step by step installation.
2. Set your gopath by opening the terminal and run **echo \$GOPATH**

Step 2

1. Clone this repository by running
git clone https://github.com/ssvergara/invoice.git
2. Copy the fabric-samples and paste it to you home directory.
3. Open the Fabcar folder and copy all the files except the **node modules**
4. Create a folder named **Fabcar** under fabric-samples/ and paste the files we copied earlier.

Output

```
fabric-samples/  
  |- Fabcar  
  |- openFabric.sh  
  |-- enrollAdmin.js  
  |-- mainregisteruser.js  
  |-- 1stparticipantUser.js  
  |-- 2ndparticipantUser.js  
  |-- OEM.js  
  |-- bankuser.js  
  |-- appmain.js  
  |-- app1stparticipant.js  
  |-- appOEM.js  
  |-- appBank.js
```

4. Create a folder **fabcar** under fabric-samples/chaincode/
5. Make a new folder **go**.
6. Inside the cloned repository, under chaincode, Copy the **go** folder and paste it on to fabric-samples/chaincode/

Output

```
fabric-samples/  
|- chaincode  
|- fabcar  
|--go  
|---scm.go
```

Step 3.

1. Open terminal and go to this directory **fabric-samples/Fabcar/**
2. Start the fabric `./openFabric.sh`
3. Run `npm install` to create the modules.
4. Run `enrollAdmin.js`
5. Run `node mainregisteruser.js`
6. Run `node appmain.js`

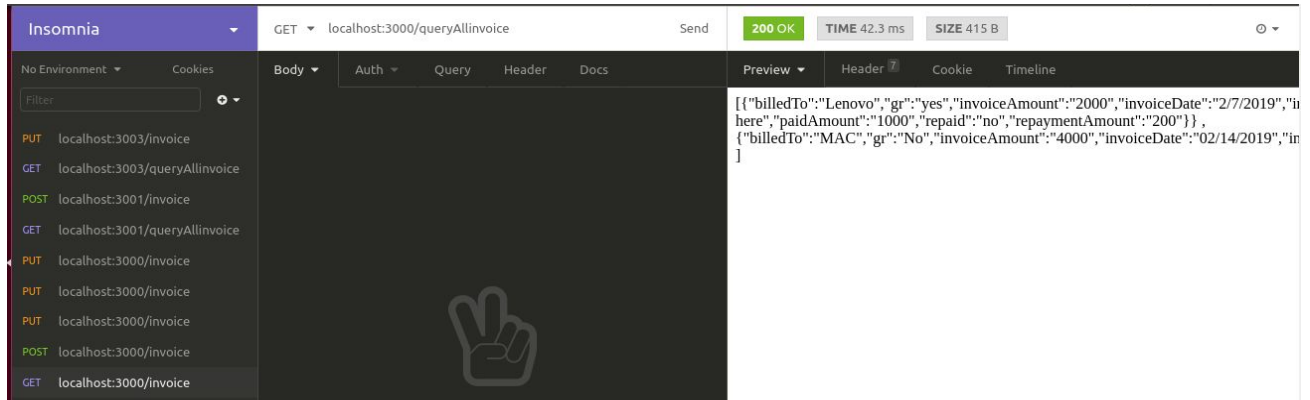
Output

```
academy@academy-VirtualBox:~/fabric-samples/fabcar1$ node enrollAdmin.js
Store path:/home/academy/fabric-samples/fabcar1/hfc-key-store
Successfully enrolled admin user "admin"
Assigned the admin user to the fabric client :{{"name":"admin","mspid":"Org1MSP","roles":null,"affiliation":"","enrollmentSecret":"","enrollme
nt":{"signInIdentity":{"3b9bc5e8fbf887f60ae2e0303ed4bca676561536c128620b14458Selfef212e"},"identity":{"certificate":"-----BEGIN CERTIFICATE---
-\\n\\nIICATCCAIAwIBAgIUfZvKwc+H4ZYLAJ3ETk+4QC/zmx4wcYgIKoZIZj0EAwiW/nCzELMAKAJAUEBHMCMVMMXEZARBgNVBAgtCKnbhGlm3JuaWEExFjaUBGNVBACDYNhbNbiBGcm
FuY2l2y28xcGTAXBgNVBAAOTG9yZzEuZXhhbXBsZS5ZSjBjb28xNDhAaBgNVBAMTInE2NHhLn9yZzEuZXhhbXBsZS5ZSjBjb28wIHhmNTkwMjEyMDAwNmAwNmhmNCNjAWMiJmYMDAxLnMjAJwIjAhMQ8wDOYD
VQQLZWZjBGlbnQXDAJBAGNBVABTBWFKbkblUWFkwEYWhkoZI/nzj0CAQYIKoZIZj0DAQCDQgEXEFyz/FaqUKO4HuIsRZN13xVPj248Jreniu59o0/nKhWConxkQ82+Aqjyz28hnKoLI
bLwsKPRHFAYpsaXCgU6NSHGowDgDYVR8PAQH/nBAQDAgeAMAawGA1UdEWEB/QCMAAHQYDVRODBVEFHjdVCCPnFWCGEG+oIp4GQT/nHVISMCSGA1UdIwQkMCAIEISgg3NdtruLoM
ZNAUYdFBHNMarRst3dsualc2XKL8/nMAOGCCGSMS49BAMCA0AMEQCIE+3jwttwS6NRqlNDMP/d/vmakqNTx5SNz/oVly/nfsykAIANSTCGWF5kSDVY1KofnXCAamqtBVXVOPEARZSLZ
oFgw==\\n-----END CERTIFICATE-----\\n\\n}}}}
academy@academy-VirtualBox:~/fabric-samples/fabcar1$ node mainRegisterUser.js
Store path:/home/academy/fabric-samples/fabcar1/hfc-key-store
Successfully loaded admin from persistence
Successfully registered user1 - secret:VR0HEEq1sHSQ
Successfully enrolled member user "user1"
User1 was successfully registered and enrolled and is ready to interact with the fabric network
academy@academy-VirtualBox:~/fabric-samples/fabcar1$ node appmain.js
Store path:/home/academy/fabric-samples/fabcar1/hfc-key-store
Example app listening on port 3000!
Successfully loaded user1 from persistence
Assigning transaction id: 72ecd401ab0e89e48144f57aeb3bc405687f37657196dda8d4f97c3ac22d551
{ chaincodeId: 'scm',
  chainId: 'mychannel',
  txid:
    TransactionID {
      _nonce: <Buffer bc 32 ba 60 a8 f8 ec 57 78 a1 ff a5 5f 47 3f 76 b2 d7 1c 9d e4 c6 32 a8>,
      _transaction_id: '72ecd401ab0e89e48144f57aeb3bc405687f37657196dda8d4f97c3ac22d551',
      _admin: false },
  args: [ undefined ] }
```

Step 4

Test the endpoints using Insomnia REST Client. [Download Insomnia REST Client here: <https://insomnia.rest/download/>]

1. Choose the function GET and enter the **localhost:3000/queryAllinvoice**. You must see a data like this:



2. Create a new tab and choose the function **POST** and enter the **localhost:3000/invoice**. Create the parameters and create your own invoices and click **Send**;

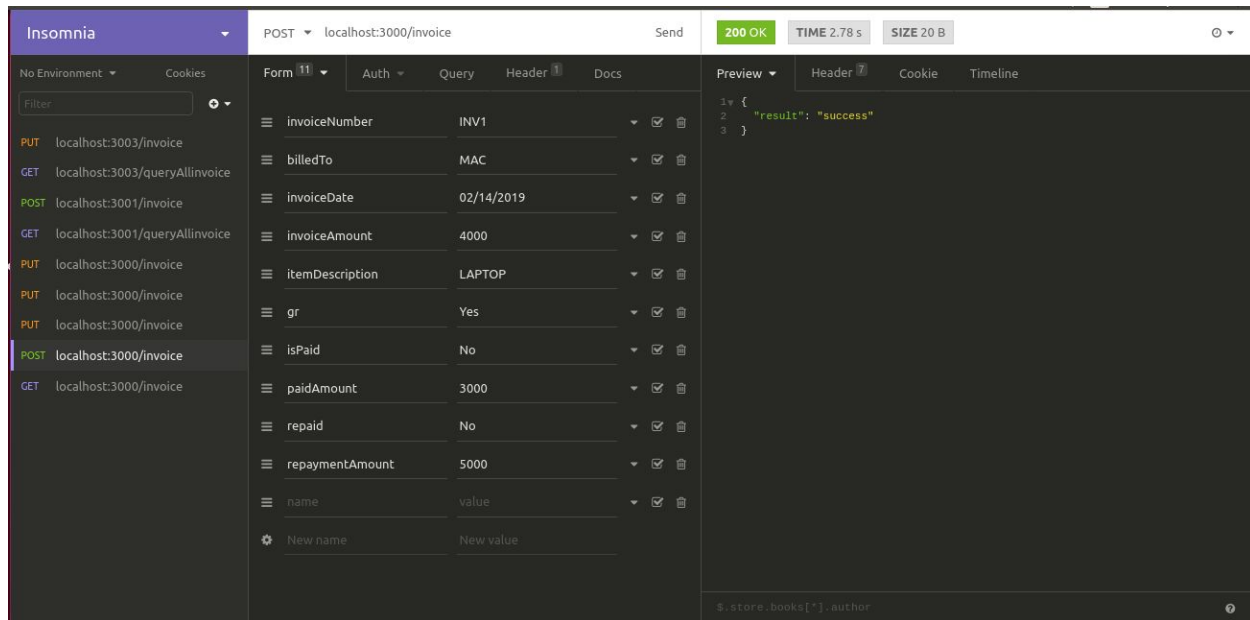
Raise Invoice

Parameters

- **invoiceNumber**
- **billedTo**
- **invoiceDate**
- **invoiceAmount**
- **itemDescription**
- **gr**
- **isPaid**
- **paidAmount**
- **repaid**
- **repaymentAmount**

Note: gr, isPaid, paidAmount, repaid, repaymentAmount, default values are as follows:

gr = N
isPaid = N
paidAmount = 0
repaid = N
repaymentAmount = 0

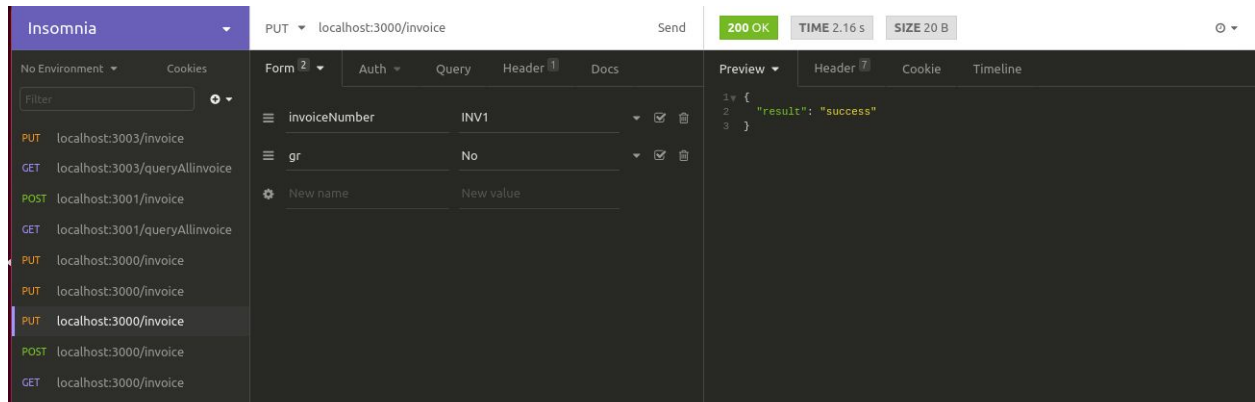


3. Create a new tab and choose the function **PUT** and enter the localhost:3000/invoice. Create the parameters and create your own invoices and click **Send**;

Goods Received

Parameters

- invoiceNumber
- gr

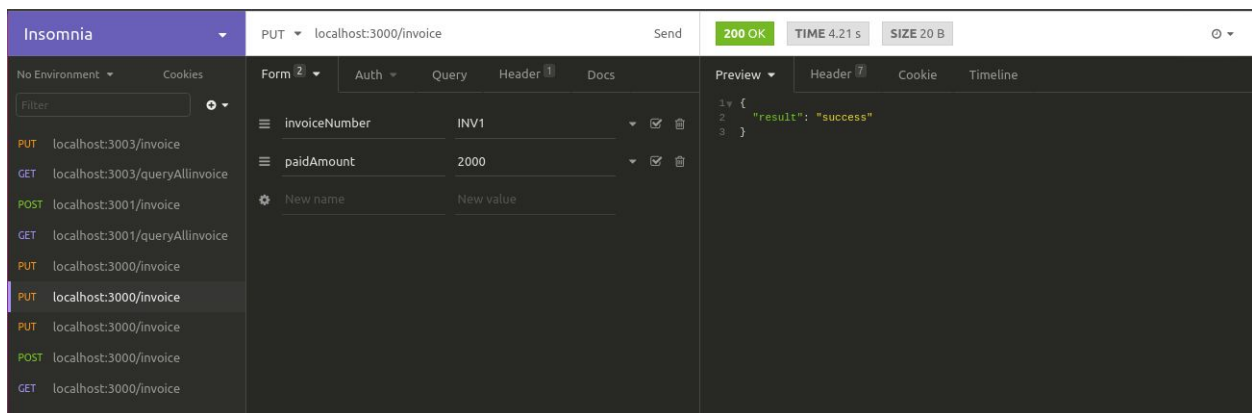


4. Create a new tab and choose the function **PUT** and enter the localhost:3000/invoice. Create the parameters and create your own invoices and click **Send**;

Bank Payment to Supplier Parameters

- invoiceNumber
- paidAmount

Note: Paid Amount must be less than Invoice Amount



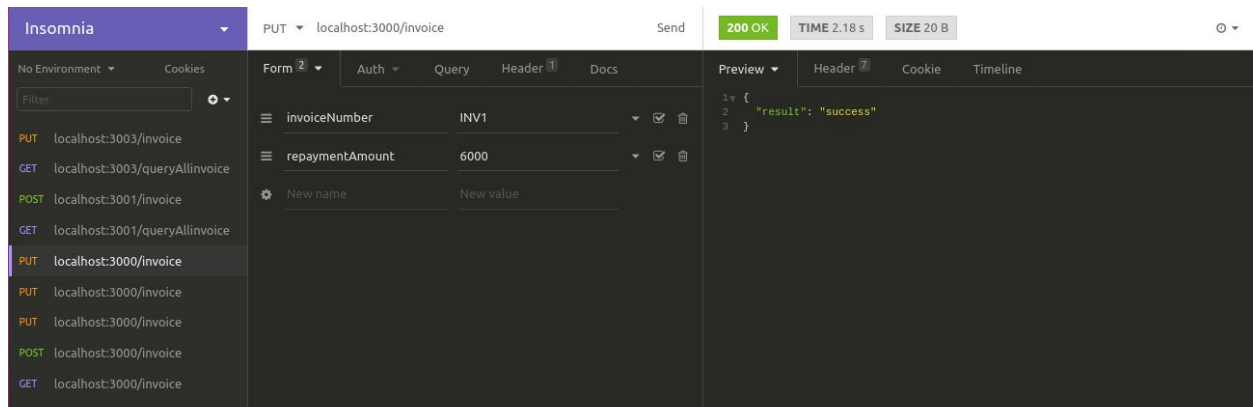
5. Create a new tab and choose the function **PUT** and enter the localhost:3000/invoice. Create the parameters and create your own invoices and click **Send**;

OEM repays to Bank

Parameters

- **invoiceNumber**
- **repaymentAmount**

Note: Repayment Amount must be greater than the value of Paid Amount.



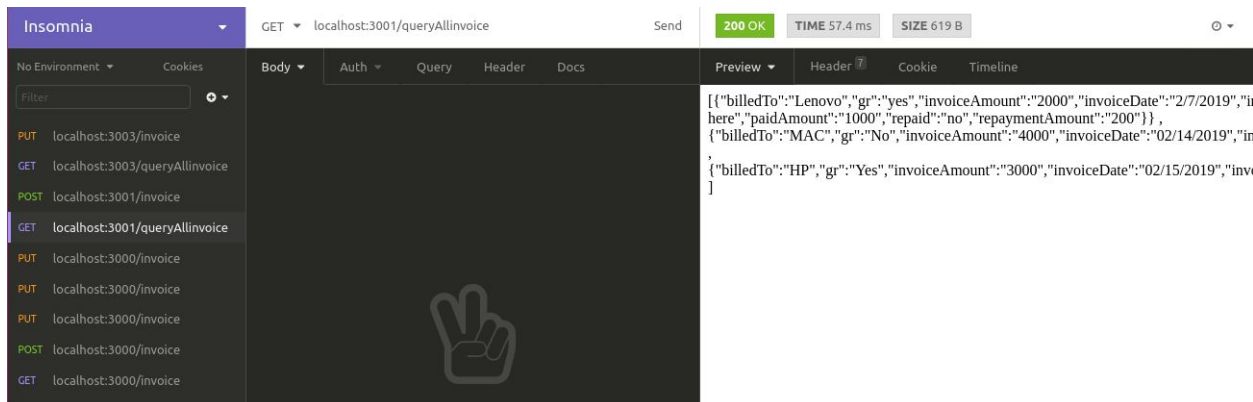
Creating Invoice using Hyperledger Fabric | Optional

Step 1 | Raise Invoice (only supplier can create Invoice)

1. Open a new terminal and open the folder **fabric-samples/Fabcar/**
2. Enroll a new user **node enrollAdmin.js**
3. Run **node 1stparticipantUser.js**
4. Run **node app1stparticipant.js**

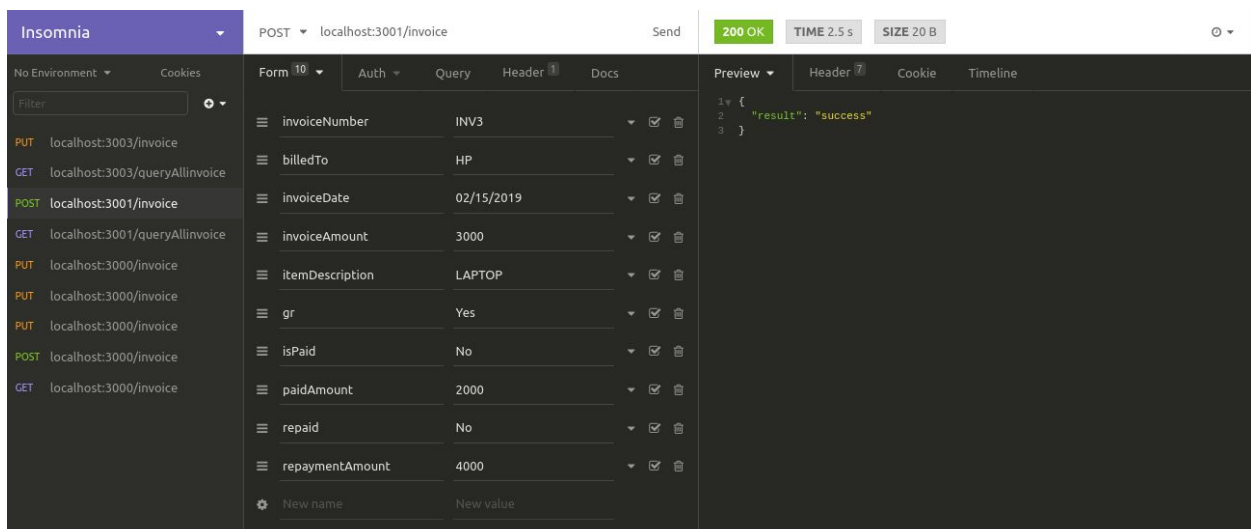
Open the Insomnia REST Client

1. Choose the function GET and enter the **localhost:3001/invoice**



2. Choose the POST function and create the parameters:

- **invoiceNumber**
- **billedTo**
- **invoiceDate**
- **invoiceAmount**
- **itemDescription**
- **gr**
- **isPaid**
- **paidAmount**
- **repaid**
- **repaymentAmount**

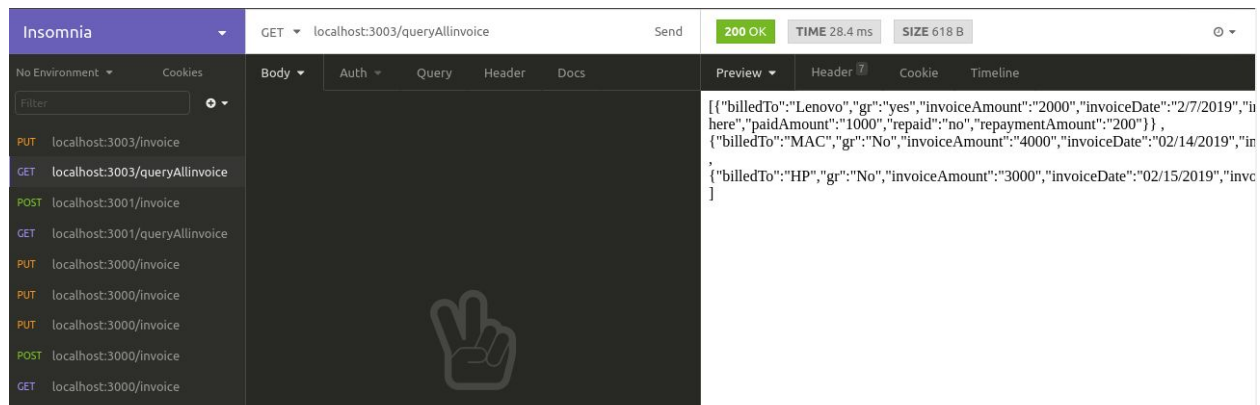


Step 2 | Goods received (only OEM can do GR)

1. Open a new terminal and open the folder **fabric-samples/Fabcar/**
2. Enroll a new user **node enrollAdmin.js**
3. Run **node OEM.js**
4. Run **node appOEM.js**

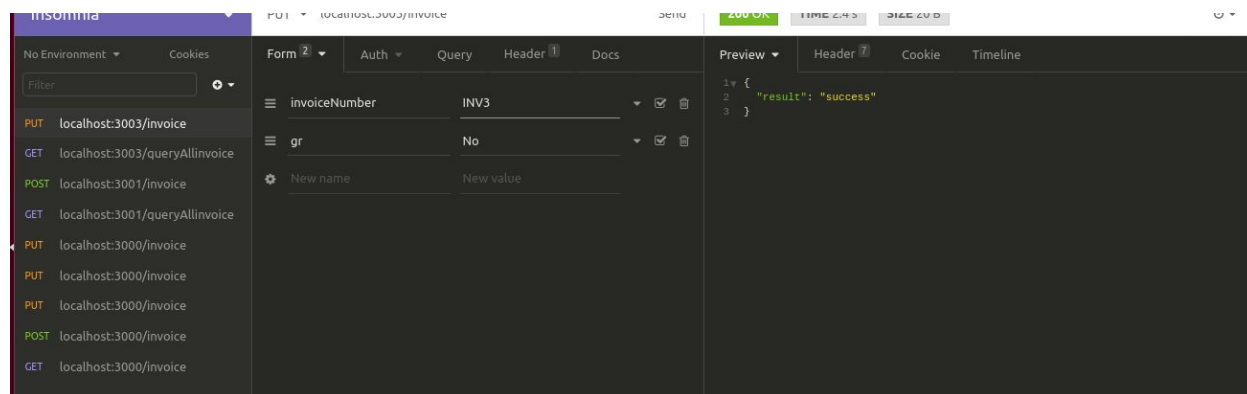
Open the Insomnia REST Client

1. Choose the function GET and enter the **localhost:3003/invoice**



2. Choose the PUT function and create the parameters:

- invoiceNumber
- gr

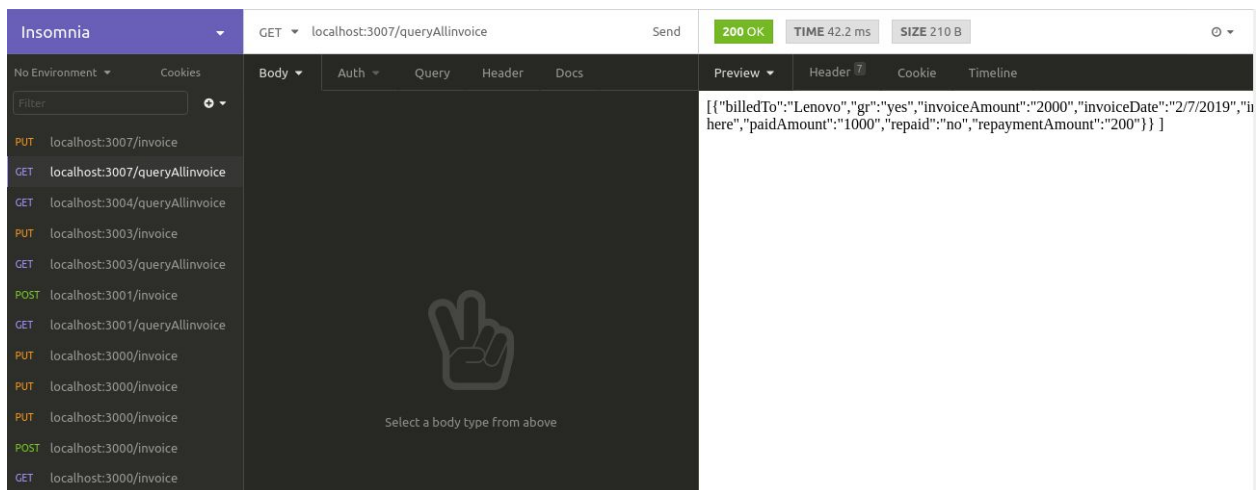


Step 3 | Goods received (only bank can pay to supplier)

1. Open a new terminal and open the folder **fabric-samples/Fabcar/**
2. Enroll a new user **node enrollAdmin.js**
3. Run **node bankuser.js**
4. Run **node appBank.js**

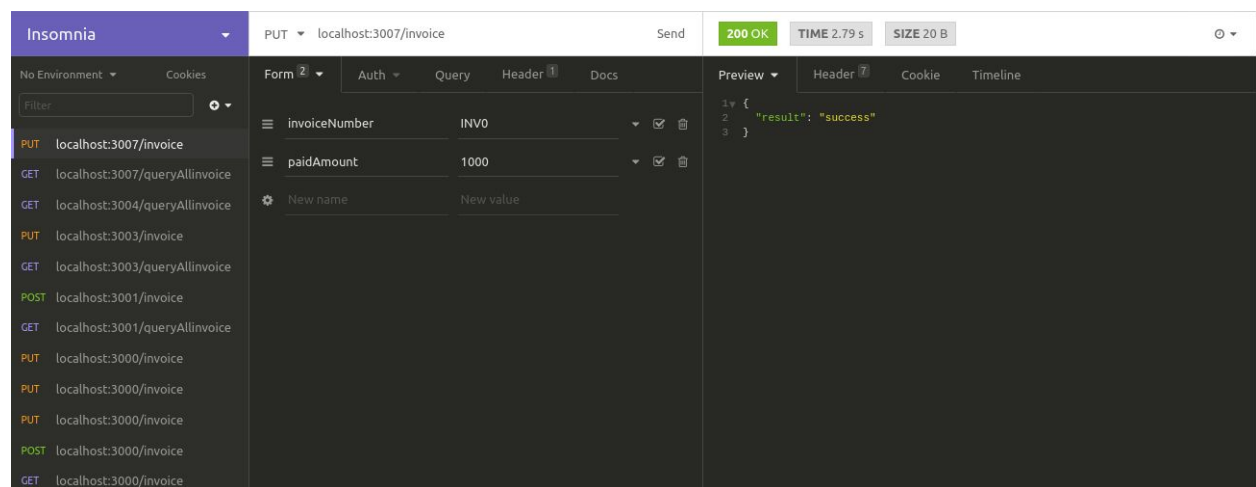
Open the Insomnia REST Client

1. Choose the function GET and enter the **localhost:3007/invoice**



2. Choose the PUT function and create the parameters:

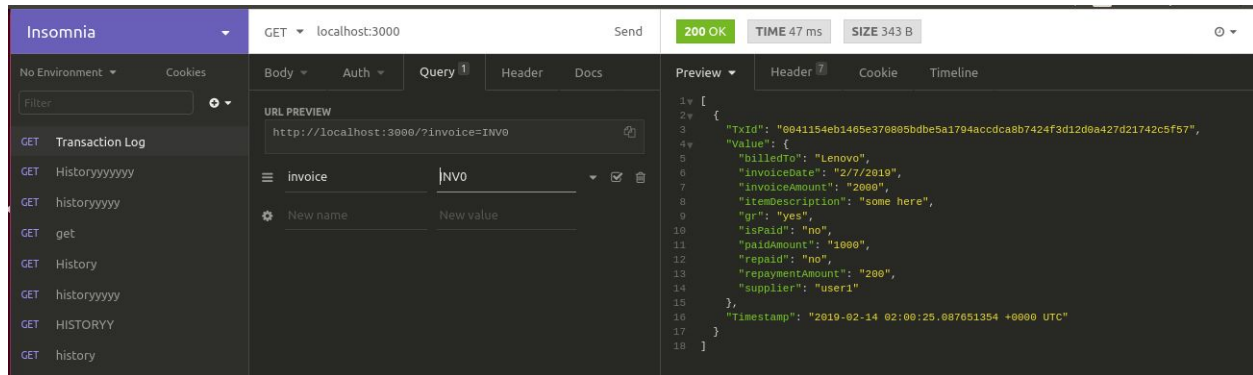
- invoiceNumber
- paidAmount



Step 4 | Transaction Log

Open the Insomnia Rest Client. Choose the function **GET** and enter the localhost:3000

- Click the query tab and type **invoice** as the name and **INV0** as the value to get the transaction Id and timestamp.

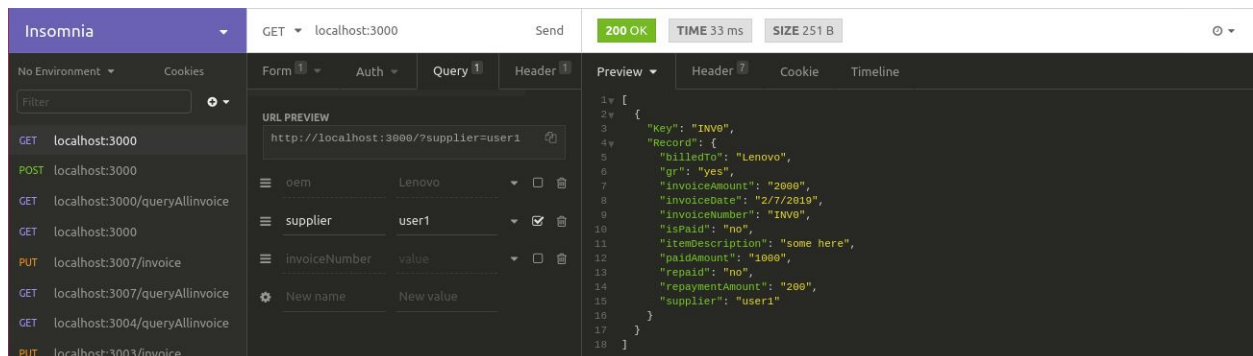


Step 5 | Display Invoices

Open the Insomnia Rest Client, choose the function **GET** and enter the localhost:3000

Display invoices per Supplier

- Type the **user1** in **supplier** to get the invoices per supplier



Display invoices per OEM

- Type the **Lenovo** in **oem** to get the invoices per oem

Insomnia

No Environment Cookies

Filter

GETlocalhost:3000

POSTlocalhost:3000

GETlocalhost:3000/queryAllinvoice

GETlocalhost:3000

PUTlocalhost:3007/invoice

GETlocalhost:3007/queryAllinvoice

GETlocalhost:3004/queryAllinvoice

PUTlocalhost:3003/invoice

GETlocalhost:3000Send

200 OKTIME 753 msSIZE 251 B

FormAuthQueryHeader

URL PREVIEW

oemLenovo

suppliervalue

invoiceNumbervalue

New nameNew value

PreviewHeaderCookieTimeline

```
1v [
2  {
3    "Key": "INVO",
4    "Record": {
5      "billedTo": "Lenovo",
6      "gr": "yes",
7      "invoiceAmount": "2000",
8      "invoiceDate": "2/7/2019",
9      "invoiceNumber": "INVO",
10     "isPaid": "no",
11     "itemDescription": "some here",
12     "paidAmount": "1000",
13     "repaid": "no",
14     "repaymentAmount": "200",
15     "supplier": "user1"
16   }
17 }
18 ]
```