

Pharma Net

Network setup and Execution as follows.

Introduction:

This project contains 4 phases

Phase 1: Project Setup

Step 1.1: Extract Zip file

Step 1.2: About zip file and its content

Step 1.3: Path Setup

Phase 2: Pre-Network setup

Step 2.1: Generate Certification and confix tx gen files.

Step 2.2: bring up network

Step 3.3: Join all the peers to the network.

Phase 3: Install & Instantiate Chain Code.

Step 3.1: install Chaincode into all the peers.

Step 3.2: Instantiate chiancode into any one peer. In this case peer0.manufacturer.

Phase 4: Node application and client testing.

Step 4.1: Bring up the node application.

Step 4.2: Execute test cases from Postman collections result details.

Lets go-over each and every phase in detail step by step.

Phase 1: Project Setup

Step 1.1: Download the zip file Capstone_Project_Sathish_Kumar.zip and extract it in any directory.

Step 1.2: You can see 4 folders and few files.

1. Application: This folder contains client and server application required files.
2. Chaincode: This folder contains fabric network business logic that is chaincode files.
3. Network: this folder contains all the fabric network setup files. Like crypto-config, channel-artificates, scripts, and .sh files.
4. Tests: This folder contains postman collection json files to test the application and chain code.
5. Instructions.docx and pdf files. These files explains how to setup, run and test this application.

Step 1.3: Set or Add path variable in .base_profile or terminal.

<basefolder> can be any user specific folder.

`<basefolder>/Capstone_Project_Sathish_Kumar/pharma-net/network/bin`

The bin folder contains files for Mac-OS. If user uses different os then need to copy the required files into the bin folder to verify this application.

Phase 2: Pre-Network setup

Step 2.1: Generate Certificate files using fabricnetwork.sh shell script file.

In new terminal window execute below commands.

Command 2.1: cd
<basefolder>/Capstone_Project_Sathish_Kumar/pharma-net/network

Command 2.2: create crypto-config & channel-artifacts folder if not available.

Command 2.3: execute fabricNetwork.sh generate command
. fabricNetwork.sh generate

This will generate the following network setup files for each organizations.

- File 1: genesis.block
- File 2: channel.tx
- File 3: distributorMSPanchors.tx
- File 4: manufacturerMSPanchors.tx
- File 5: transporterMSPanchors.tx
- File 6: consumerMSPanchors.tx
- File 7: retailerMSPanchors.tx

Command 2.4: execute below command to bring up the all the required docker container. If no container available, then this will download the required container.

. fabricNetwork.sh up

And also this command will create a new channel in orderer and also ask all other peers to join in this channel.

Phase 3: Install & Instantiate Chain Code

Command 3.1: execute below command to install the chain code on all the peer servers.

```
. fabricNetwork.sh install
```

Command 3.2: in a new terminal window execute below commands. This will login to chaincode container and install node app on chaincode container.

```
docker exec -it chaincode /bin/bash  
npm install  
npm run start-dev
```

Command 3.3: goto previous terminal windows and execute below command. This command will instantiate the chaincode in node application.

```
. fabricNetwork.sh instantiate
```

Phase 4: Node server application and client testing.

Command 4.1: In a new terminal window execute below commands one by one. This will install node application.

```
cd <basefolder>/Capstone_Project_Sathish_Kumar/pharma-  
net/application  
npm install  
node index.js
```

The node application is ready execute the transactions in fabric network blockchain application. You can see below the log statement “Distributed Pharma Network App listening on port 3000!”

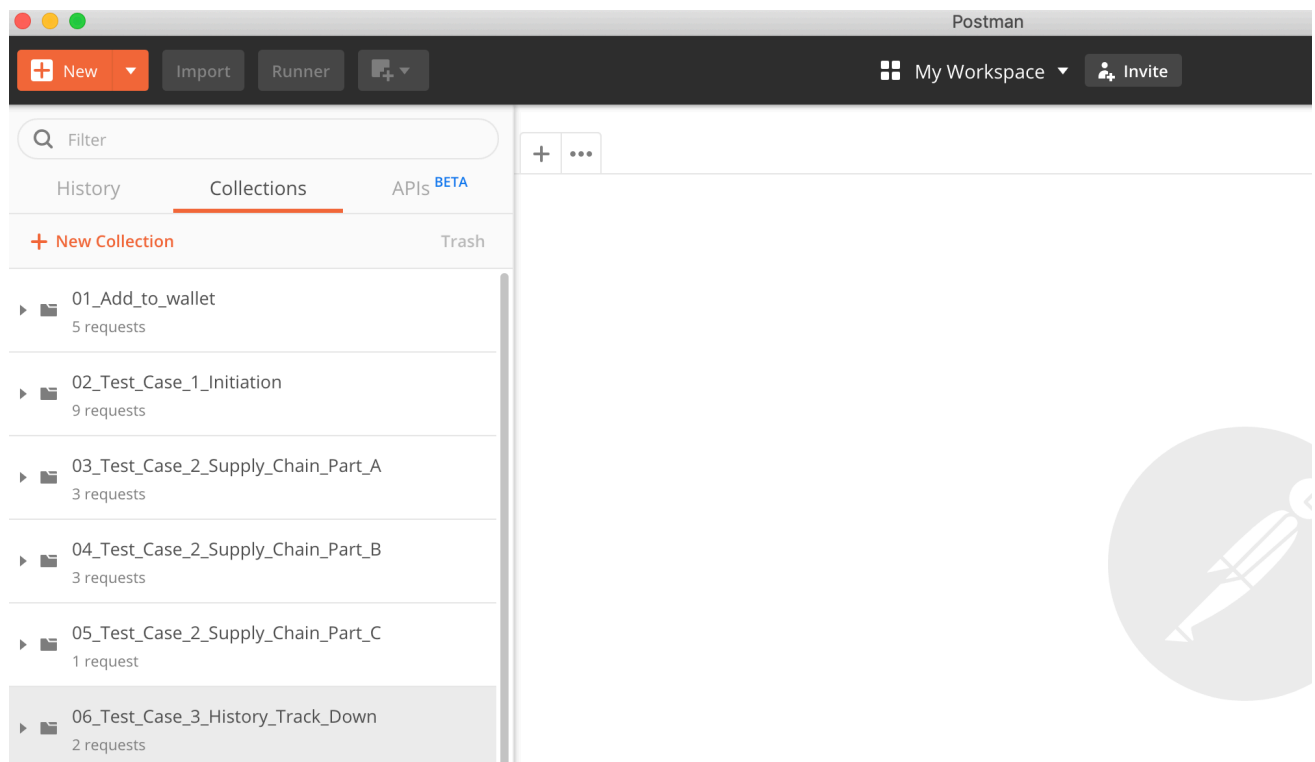
Command 4.2: Open postman application for testing.

Note : While executing postman if you get any error please try to execute again. It might take some more time to commit the data into the network. Hence postman some time may throw exception.

Command 4.3: Import all the postman collections available in<basefolder>/Capstone_Project_Sathish_Kumar/pharma-net/test folder in the same order.

1. 01_Add_to_wallet.postman_collection.json
2. 02_Test_Case_1_Initiation.postman_collection.json
3. 03_Test_Case_2_Supply_Chain_Part_A.postman_collection.js
n
4. 04_Test_Case_2_Supply_Chain_Part_B.postman_collection.js
n
5. 05_Test_Case_2_Supply_Chain_Part_C.postman_collection.js
n
6. 06_Test_Case_3_History_Track_Down.postman_collection.json

The postman collection snapshot looks below



Command 4.4: Edit and add the wallet private key files in add_to_wallet test case for all the organizations to execute the postman client application as below.

In a postman application goto -> 01_Add_to_wallet -> 1_AddToManufactureWallet -> In body change certificatePath and privateKeyPath attributes.

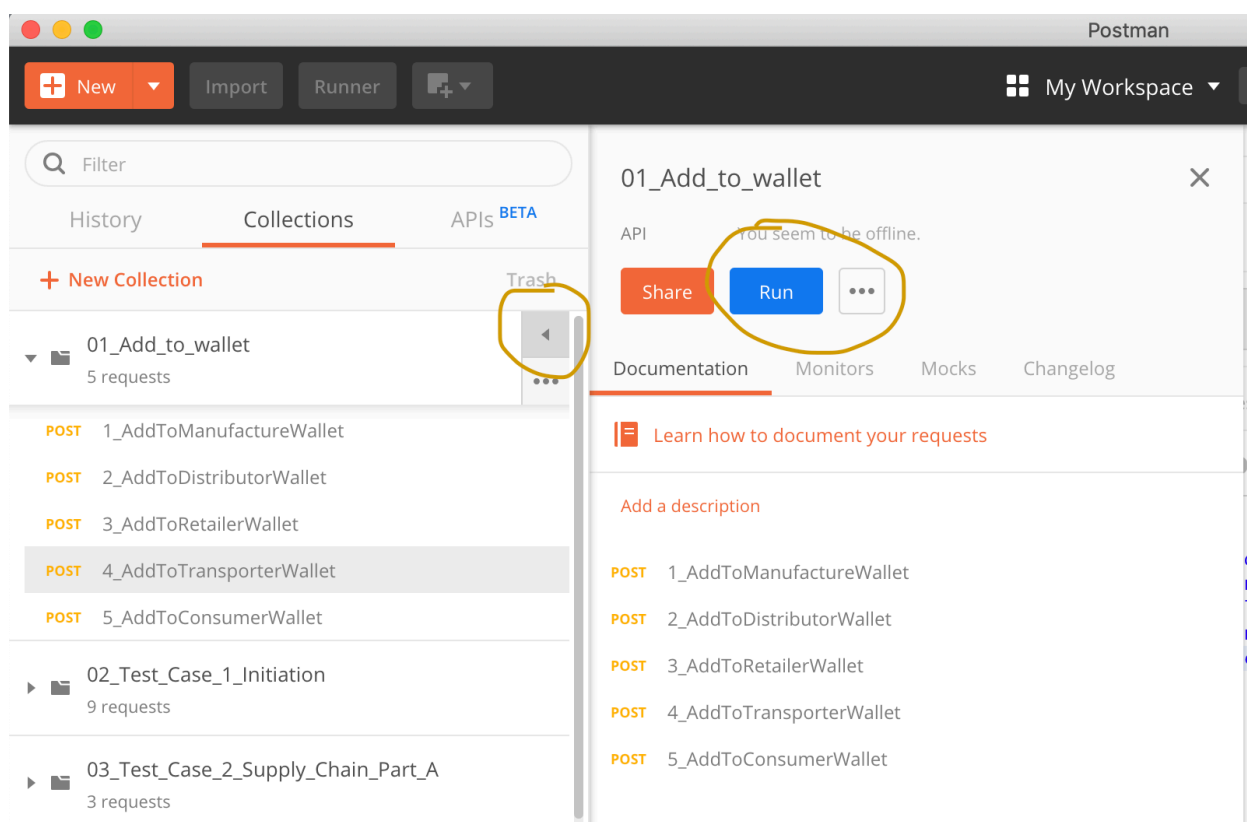
“certificatePath”:”<basefolder>/Capstone_Project_Sathish_Kumar/pharma-net/network/crypto-config/peerOrganizations/manufacturer.pharma-network.com/users/Admin@manufacturer.pharma-network.com/msp/signcerts/Admin@manufacturer.pharma-network.com-cert.pem”

“privateKeyPath”:”<basefolder>/Capstone_Project_Sathish_Kumar/pharma-net/network/crypto-config/peerOrganizations/manufacturer.pharma-network.com/users/Admin@manufacturer.pharma-

network.com/msp/keystore/d5a4efd1ad5cd7cab757afa3f25a86afe79bc1725d4349cb3ebfbd31b51b48f3_sk”

Note : The keystore file will be generated at runtime hence you cannot use the same file mentioned in examples.

Command 4.5: Once you added certificatePath & privateKeyPath for all the 5 organizations. You can use postman collection runner util to run 01_Add_to_Wallet test cases at once as below.



Hit the run button it will prompt to select the test cases. Beased on the need you can chose multiple test cases. Or All, Or None. Please select all the test case and run.

And output would be.

Add To Wallet testcases output.

The screenshot displays the 'Collection Runner' application window. The top navigation bar includes 'Collection Runner', 'Run Results', and 'My Workspace'. Below this, the test suite '01_Add_to_wallet' is shown with 'No Environment' and a 'just now' timestamp. It features a 'Run Summary' button and an 'Export Results' button. The test results are organized into 'Iteration 1', listing five POST requests. Each request row shows the method, name, URL, and response details (status, time, size). A yellow box highlights the response details for the first four requests, which all show '200 OK'. A red arrow points to the fifth request, '5_AddToConsumerWallet', with the text 'On Clicking Testcase name you can see the response body'. A dropdown menu is open for this request, showing 'Request URL', 'Request Headers (9)', 'Request Body', 'Response Headers (7)', and 'Response Body'. The 'Response Body' is expanded, showing a JSON object:

```
{  "status": "success",  "message": "Identity for organization : consumer, added to wallet successfully."}
```

. A red arrow points to the '200 OK' status of the fifth request with the text 'Overall response status code of the test cases.'

Iteration	Method	Testcase Name	URL	Status	Time	Size
1	POST	1_AddToManufacturerWallet	http://localhost:3000/ad... / 1_AddToManufacturerWallet	200 OK	9 ms	104 B
		This request does not have any tests.				
	POST	2_AddToDistributorWallet	http://localhost:3000/ad... / 2_AddToDistributorWallet	200 OK	6 ms	103 B
		This request does not have any tests.				
	POST	3_AddToRetailerWallet	http://localhost:3000/ad... / 3_AddToRetailerWallet	200 OK	8 ms	100 B
		This request does not have any tests.				
	POST	4_AddToTransporterWallet	http://localhost:3000/ad... / 4_AddToTransporterWallet	200 OK	6 ms	103 B
		This request does not have any tests.				
	POST	5_AddToConsumerWallet	http://localhost:3000/ad... / 5_AddToConsumerWallet	200 OK	7 ms	100 B

On Clicking Testcase name you can see the response body

Overall response status code of the test cases.

Command 4.6: Like this you can execute one by one collections.

Executing 02_Test_Case_1_Initiation collection and result would be.

The screenshot displays the 'Collection Runner' window with the 'Run Results' tab selected. The collection '02_Test_Case_1_Initiation' is shown with a '0 FAILED' status and a '2 mins ago' timestamp. A red arrow points to the '02_Test_Case_1_Initiation' collection name, with a text overlay stating: 'All Test cases result can be seen by clicking Testcase name and Response Body'. The test results are listed in a table with columns for status, request details, and response metrics. A yellow box highlights the response metrics for the first five test cases.

Iteration	Method	Test Case Name	Request URL	Status	Response Time (ms)	Response Size (B)
1	POST	001_Registrator_Manufacture_1	http://localhost:3000/reg...	200 OK	61313 ms	274 B
		This request does not have any tests.				
	POST	002_Registrator_Transporter_1	http://localhost:3000/reg...	200 OK	2144 ms	266 B
		This request does not have any tests.				
	POST	003_Registrator_Transporter_2	http://localhost:3000/reg...	200 OK	2138 ms	274 B
		This request does not have any tests.				
	POST	004_Registrator_Distributor_1	http://localhost:3000/reg...	200 OK	2256 ms	271 B
		This request does not have any tests.				
	POST	005_Registrator_Retailer_1	http://localhost:3000/reg...	200 OK	2113 ms	265 B
		This request does not have any tests.				
	POST			200 OK	2171 ms	325 B
		This request does not have any tests.				
	POST			200 OK	2122 ms	325 B
		This request does not have any tests.				
	POST			200 OK	2143 ms	325 B
		This request does not have any tests.				
	POST			200 OK	2143 ms	325 B
		This request does not have any tests.				

Request URL
Request Headers (9)
Request Body
Response Headers (7)
Response Body

```
{
  "status": "success",
  "message": "The New Company has been Registered Successfully.",
  "drug": {
    "companyCrn": "RET001",
    "companyName": "Upgrad",
    "location": "Mumbai",
    "organisationRole": "retailer",
    "createdAt": "2019-12-30T11:46:54.220Z",
    "modifiedAt": "2019-12-30T11:46:54.220Z"
  }
}
```

Command 4.7 : Executing 03_Test_Case_2_Supply_Chain_Part_A and its result would be

Collection Runner

Run Results

My Workspace

0 PASSED

0 FAILED

03_Test_Case_2_Supply_Chain_Part_A

No Environment

just now

Run Summary

Iteration 1

POST

001_CreatrePO_DIST_MANU

http://localhost:3000/cre...

...rt_A / 001_CreatrePO_DIST_MANU

200 OK

2154 ms

215 B

This request does not have any tests.

POST

012_CreatreShipment_MANUFA_DISTRI

http://localhost:3000/cre...

...reatreShipment_MANUFA_DISTRI

200 OK

2154 ms

271 B

This request does not have any tests.

POST

013_Update_Shipment_TRANS_DISTRI

http://localhost:3000/up...

..._Update_Shipment_TRANS_DISTRI

200 OK

2137 ms

276 B

This request does not have any tests.

Request Headers (9)

Request Body

Response Headers (7)

Response Body

```
{
  "status": "success",
  "message": "The Drug delivered successfully. The
shipment updates Details are : ",
  "shipment": {
    "creator": "MAN001",
    "assets": [
      ["Paracetamol", "0001"],
      ["Paracetamol", "0002"],
      ["Paracetamol", "0003"]
    ],
    "transporter": "TRA001",
    "status": "delivered",
    "shipTo": "DIST001"
  }
}
```

The Shipment status updated to delivered and the Drug is now with DIST001

Command 4.8: Executing 04_Test_Case_2_Supply_Chain_Part_B and its result would be

Collection Runner

Run Results

My Workspace

0 PASSED

0 FAILED

04_Test_Case_2_Supply_Chain_Part_B

No Environment

just now

Run Summary

Export Results

Iteration 1

POST

014_CreateShipment_DISTRI_RETAILER

http://localhost:3000/cre...

...reateShipment_DISTRI_RETAILER

200 OK

2137 ms

248 B

This request does not have any tests.

POST

015_Update_Shipment_TRANS_RETAIL

http://localhost:3000/up...

...Update_Shipment_TRANS_RETAIL

200 OK

2144 ms

253 B

This request has the following tests:

Request Headers (9)

Request Body

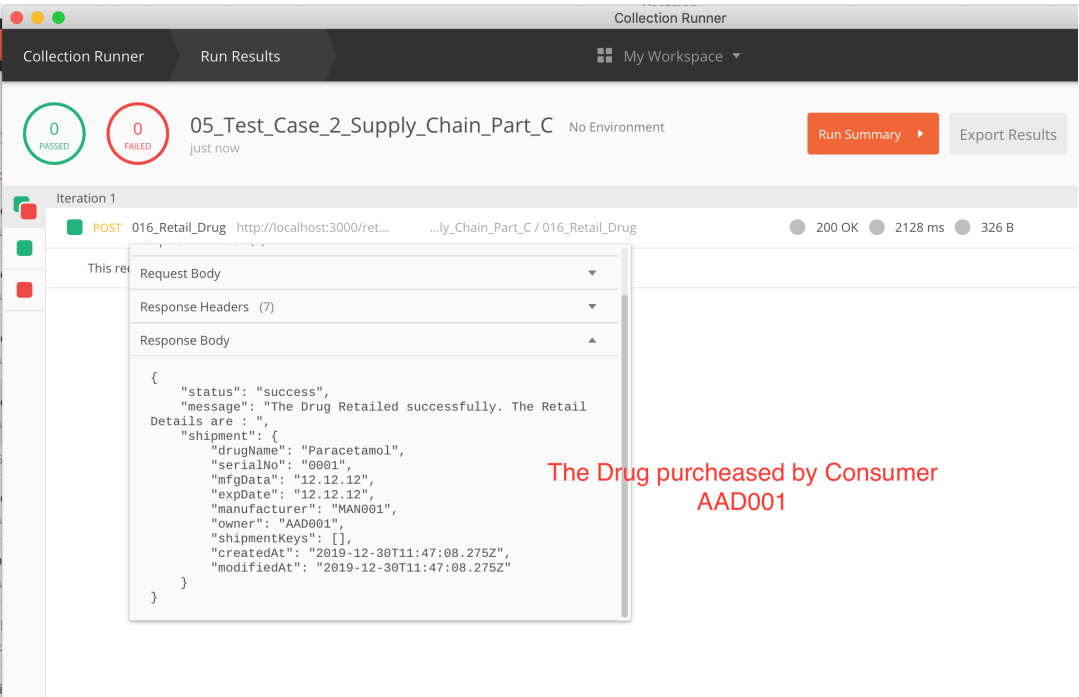
Response Headers (7)

Response Body

```
{
  "status": "success",
  "message": "The Drug delivered successfully. The shipment updates Details are : ",
  "shipment": {
    "creator": "DIST001",
    "assets": [
      ["Paracetamol", "0001"],
      ["Paracetamol", "0002"]
    ],
    "transporter": "TRA001",
    "status": "delivered",
    "shipTo": "RET001"
  }
}
```

Finally Drug Moved to retailer

Command 4.8: Executing 05_Test_Case_2_Supply_Chain_Part_C and its result would be



Command 4.9: Executing 06_Test_Case_3_History_Track_Down and its result would be

